AWS Snowball

Developer Guide

AWS Snowball: Developer Guide

Table of Contents

This guide is for the Snowball Edge (100 TB of storage space). If you are looking for documentation for the Snowball, see the AWS Snowball User Guide.

What Is an AWS Snowball Edge?

AWS Snowball is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the internet. AWS Snowball Edge is a 100 TB data transfer device with on-board storage and compute power for select AWS capabilities. In addition to transferring data to AWS, Snowball Edge can undertake local processing and edge-computing workloads.

Features include:

- An Amazon S3-compatible endpoint on the device.
- A file interface with Network File System (NFS) support.
- A cluster mode where multiple Snowball Edge devices can act as a single, scalable storage pool with increased durability.
- The ability to run AWS Lambda powered by AWS Greengrass functions as data is copied to the device.

Each AWS Snowball appliance type can transport data at speeds faster than the internet. 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. The AWS Snowball Edge appliance differs from the standard Snowball because it can bring the power of the AWS Cloud to your local environment, with local storage and compute functionality.

The storage and compute functionality is built in. With an AWS Snowball Edge appliance, there's no need for a powerful workstation to handle encryption, because all encryption happens on the appliance itself. There are two primary use cases supported for the AWS Snowball Edge appliance:

- **Local storage and compute** – You can use one AWS Snowball Edge appliance, or a cluster of appliances, to harness AWS storage and compute services—like AWS Lambda powered by AWS Greengrass and an Amazon S3 compatible interface—in an environment that might or might not have an internet connection.
- **Transfer large amounts of data into and out of Amazon S3** – As with the original AWS Snowball appliance, the Snowball, the AWS Snowball Edge appliance can be used to transfer many terabytes or petabytes of data between Amazon S3 and your local environment, bypassing the internet.

> **Note**
> There are many options for transferring your data into AWS. The AWS Snowball service is intended for transferring large amounts of data. If you want to transfer less than 30 terabytes of data, or if you don't have a local computing or clustering need, using an AWS Snowball Edge appliance might not be your most economical choice.

AWS Snowball Edge Features

AWS Snowball Edge appliances have the following features:

- Each AWS Snowball Edge appliance has 100 TB of storage capacity, giving it twice the storage space of the original Snowball appliance.
- Any one of three network adapters on the AWS Snowball Edge appliance supports network speeds of over 10 GB/second.
- Encryption is enforced, protecting your data at rest and in physical transit.

- You can import or export data between your local environments and Amazon S3, physically transporting the data with one or more AWS Snowball Edge appliances, completely bypassing the internet.
- AWS Snowball Edge appliances are their own rugged shipping containers, and the built-in E Ink display changes to show your shipping label when the appliance is ready to ship.
- AWS Snowball Edge appliances come with an on-board LCD display that can be used to manage network connections, get status, and watch helpful how-to videos to help you get the most out of your job.
- AWS Snowball Edge appliances can be clustered for local storage and compute jobs to achieve 99.999% data durability across 5 to 10 devices, and to locally grow and shrink storage on demand.
- You can use the file interface to read and write data to an AWS Snowball Edge appliance through a file share or Network File System (NFS) mount point.
- You can write Python-language Lambda functions and associate them with buckets when you create an AWS Snowball Edge appliance job. Each function triggers whenever there's a local Amazon S3 PUT object action executed on the associated bucket on the appliance.

Prerequisites for Using Snowball Edge

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

- Create an AWS account and an administrator user in AWS Identity and Access Management (IAM). For more information, see Sign Up for AWS (p. 20).
- 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. 31).
- 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. 14).
  - 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.

Services Related to the AWS Snowball Edge

You can use AWS Snowball with an AWS Snowball Edge appliance with the following related AWS services:

- **Amazon S3** – You can use the Amazon S3 Adapter for Snowball, which supports a subset of the Amazon S3 API actions, to transfer data onto an AWS Snowball Edge appliance. You can do this in a single AWS Snowball Edge appliance or in a cluster of appliances for increased data durability. In addition, you can import data hosted on an AWS Snowball Edge appliance into Amazon S3 and your local environment through a shipped AWS Snowball Edge appliance. For more information on using Amazon S3, see the Amazon Simple Storage Service Getting Started Guide.
- **AWS Lambda powered by AWS Greengrass** – You can trigger Lambda functions based on Amazon S3 storage actions made on an AWS Snowball Edge appliance. These Lambda functions are associated with an AWS Snowball Edge appliance during job creation. For more information on using Lambda, see the AWS Lambda Developer Guide.

Accessing the AWS Snowball Service

There are two ways to access both the AWS Snowball service and the appliances. You can either use the AWS Snowball Management Console or the job management API to create and manage jobs for AWS Snowball. For more information on the job management API, see Job Management API Reference for AWS Snowball.

Accessing the AWS Snowball Edge Appliance

Once your AWS Snowball Edge appliance or appliances are onsite, you can access them through either the LCD display built into each appliance, the Amazon S3 Adapter for Snowball, or through the available file interface. You can also use the adapter or the file interface to transfer data. For more information, see Using an AWS Snowball Edge (p. 34).

Pricing for the AWS Snowball Edge

For information about the pricing and fees associated with the service and its appliances, see AWS Snowball Edge Pricing.

Are You a First-Time AWS Snowball User?

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

1. To learn more about the different types of jobs, see Jobs for AWS Snowball Edge Appliances (p. 9).
2. For an end-to-end overview of how to use an AWS Snowball Edge appliance, see How AWS Snowball Works with the Snowball Edge (p. 7).
3. When you're ready to get started, see Getting Started with an AWS Snowball Edge Appliance (p. 20).

AWS Snowball Device Differences

The Snowball and the Snowball Edge are two different devices. This guide is for the Snowball Edge. If you are looking for documentation for the Snowball, see the *AWS Snowball User 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	✓	

Use Case	Snowball	Snowball Edge
Export from Amazon S3	✓	✓
Durable local storage		✓
Local compute with AWS Lambda		✓
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. For information on specifications for the Snowball Edge, see AWS Snowball Edge Specifications (p. 91).

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	✓	

Storage capacity (usable capacity)	Snowball	Snowball Edge
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 IP addresses.	✓	✓
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.
- 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.
- 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 (p. 36).

- 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 (p. 46).
- 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 (p. 52).
- 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 in the *AWS Snowball User Guide.*

- Supported Network Hardware (p. 92) in this 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 Snowball Edge

With AWS Snowball, you can use one of two appliances. From this guide, you can learn how to use AWS Snowball with an AWS Snowball Edge appliance. The appliances are owned by AWS, and they reside at your on-premises location while they're in use.

There are three job types you can use with an AWS Snowball Edge appliance. Although the job types differ in their use cases, every job type has the same workflow for how you order, receive, and return appliances.

The shared workflow

1. **Create the job** – Each job is created in the AWS Snowball Management Console or programmatically through the job management API, and the status for a job can be tracked in the console or through the API.
2. **An appliance is prepared for your job** – We prepare an AWS Snowball Edge appliance for your job, and the status of your job is now **Preparing Snowball**.
3. **An appliance 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 console or with the job management API. For information on who your region's carrier is, see Shipping Considerations for AWS Snowball (p. 68).
4. **Receive the appliance** – A few days later, your region's carrier delivers the AWS Snowball Edge appliance to the address that you provided when you created the job, and the status of your job changes to **Delivered to you**. When it arrives, you'll notice that it didn't arrive in a box, because the appliance 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 appliance 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 appliance, and it is encrypted so that only the unlock code can decrypt it. You can get the manifest from the console or with the job management API when the appliance is on-premises at your location.
   - The unlock code is a 29-character code used to decrypt the manifest. You can get the unlock code from the console or with the job management API. We recommend that you keep the unlock code saved somewhere separate from the manifest to prevent unauthorized access to the appliance while it's at your facility.
6. **Position the hardware** – Move the appliance into your data center and open it following the instructions on the case. Connect the appliance to power and your local network.
7. **Power on the appliance** – Next, power on the appliance by pressing the power button above the LCD display. Wait a few minutes, and the **Ready** screen appears.

8. **Get the IP address for the appliance** – The LCD display has a **CONNECTION** tab on it. Tap this tab and get the IP address for the AWS Snowball Edge appliance.
9. **Use the Snowball client to unlock the appliance** – When you use the Snowball client to unlock the AWS Snowball Edge appliance, type the IP address of the appliance, the path to your manifest, and the unlock code. The Snowball client decrypts the manifest and uses it to authenticate your access to the appliance.

   Note
   For a cluster job, there are additional steps; see How Clustered Local Compute and Storage Works (p. 9).

10. **Use the appliance** – The appliance is up and running. You can use it to transfer data or for local compute and storage. You can read and write data with the Amazon S3 Adapter for Snowball or the Network File System (NFS) mount point.
11. **Prepare the appliance for its return trip** – After you're done with the appliance in your on-premises location and the file interface status is **Complete**, press the power button above the LCD display. It takes about 20 seconds or so for the appliance to power off. Unplug the appliance and its power cables into the cable nook on top of the appliance, and shut all three of the appliance's doors. The appliance is now ready to be returned.
12. **Your region's carrier returns the appliance to AWS** – When the carrier has the AWS Snowball Edge appliance, the status for the job becomes **In transit to AWS**.

    Note
    For export and cluster jobs, there are additional steps; see How Export Works (p. 8) and How Clustered Local Compute and Storage Works (p. 9).

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. 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 AWS Snowball Edge 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 has exactly one appliance associated with it. After your job parts are created, your first job part enters the **Preparing Snowball** status.

Note
The listing operation to split your job into parts is a function of Amazon S3, and you are billed for it as you are for any Amazon S3 operation.

Soon after that, we start exporting your data onto an appliance. Typically, exporting data takes one business day; however, this process can take longer. Once the export is done, AWS gets the appliance ready for pickup by your region's carrier.

When it arrives in a few days, you'll connect the AWS Snowball Edge appliance to your network and transfer the data that you want imported into Amazon S3 onto the appliance. When you're done transferring data, ship the appliance back to AWS. Once we receive a returned appliance for your export job part, we erase it completely. 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.

How Local Compute and Storage Works

You can use the local compute and storage functionality of an AWS Snowball Edge appliance with all job types in regions that support Lambda. The compute functionality is AWS Lambda powered by AWS Greengrass, where Python-language AWS Lambda functions can be triggered by Amazon S3 PUT object actions on buckets specified when you created the job. For more information, see Local Compute and Storage Only Jobs (p. 16).

How Clustered Local Compute and Storage Works

A special kind of job for local storage and compute only, the cluster job is for those workloads that require increased data durability and storage capacity. For more information, see Local Cluster Option (p. 17).

> **Note**
> As with standalone local storage and compute jobs, the data stored in a cluster can't be imported into Amazon S3 without ordering additional appliances as a part of separate import jobs. Then you could transfer the data from the cluster to those appliances and import the data when you return the appliances for the import jobs.

Clusters have anywhere from 5 to 10 AWS Snowball Edge appliances, called nodes. When you receive the nodes from your regional carrier, connect all the nodes to power and network to obtain their IP addresses. With these IP addresses, you unlock all the nodes of the cluster at once with a single unlock command, with the IP address of one of the nodes. For more information, see Using the Snowball Client (p. 36).

You can write data to an unlocked cluster by using the Amazon S3 Adapter for Snowball or the NFS mount point through the leader node, and it distributes the data among the other nodes.

When you're done with your cluster, ship all the nodes back to AWS. Once we receive a returned cluster node, we perform a complete erasure of the Snowball. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

Jobs for AWS Snowball Edge Appliances

A job in AWS Snowball is a discrete unit of work, defined when you create it in the console or the job management API. With the AWS Snowball Edge appliance, there are three different job types, all of which are capable of local storage and compute functionality. This functionality uses the file interface or Amazon S3 Adapter for Snowball to read and write data. It triggers Lambda functions based on Amazon S3 PUT object API actions running locally on the AWS Snowball Edge appliance.

> **Important**
> With an AWS Snowball Edge appliance, all jobs can use the compute functionality in regions where AWS Lambda is supported. How the compute functionality is implemented in AWS Snowball jobs is specific to Snowball—the functionality can differ significantly from how Lambda works in the cloud. Before creating your first compute job, we recommend that you familiarize yourself with how AWS Lambda powered by AWS Greengrass works. For more information, see Using AWS Lambda with an AWS Snowball Edge (p. 58).

- Import Jobs into Amazon S3 (p. 13) – The transfer of 100 TB or less of your local data copied onto a single appliance, and then moved into Amazon S3. For import jobs, AWS Snowball Edge appliances and jobs have a one-to-one relationship. Each job has exactly one appliance associated with it. If you need to import more data, you can create new import jobs or clone existing ones. When you return an appliance of this job type, that data on it is imported into Amazon S3.
- Export Jobs from Amazon S3 (p. 14) – The transfer of any amount of data (located in Amazon S3), copied onto any number of AWS Snowball Edge appliances, and then moved one AWS Snowball Edge

appliance 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 100 TB in size, and each job part has exactly one AWS Snowball Edge appliance associated with it. When you return an appliance of this job type, it's erased.

- Local Compute and Storage Only Jobs (p. 16) – These jobs involve one AWS Snowball Edge appliance, or multiple appliances used in a cluster. These jobs don't start with data in their buckets like an export job, and can't have data imported into Amazon S3 at the end like an import job. When you return an appliance of this job type, it's erased. With this job type, you also have the option of creating a cluster of appliances. A cluster improves local storage durability and you can scale up or down with local storage capacity.

  In regions where Lambda is not available, this job type will be called *Local storage only*.

Job Details

Each job is defined by the details that you specify when it's created. The following table describes all the details of a job.

Console Identifier	API Identifier	Detail Description
Job name	Description	A name for the job, containing alphanumeric characters, spaces, and any Unicode special characters.
Job type	JobType	The type of job, either import, export, or local compute and storage.
Job ID	JobId	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.
Address	AddressId	The address that the appliance will be shipped to. In the case of the API, this is the ID for the address data type.
Created date	N/A	The date that you created this job.
Shipping speed	ShippingOption	Speed options are based on region. For more information, see Shipping Speeds (p. 71).
IAM role ARN	RoleARN	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 AWS Snowball to assume is only

Console Identifier	API Identifier	Detail Description
		used to copy your data between your Amazon S3 buckets and the Snowball. For more information, see Permissions Required to Use the AWS Snowball Console (p. 81).
AWS KMS key	KmsKeyARN	In AWS 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 AWS Snowball (p. 73).
Snowball capacity	SnowballCapacityPreference	AWS Snowball appliances come in three sizes: the 50 TB and 80 TB Snowballs and the 100 TB AWS Snowball Edge appliance. Which are available depends on your AWS Region.
Storage service	N/A	The AWS storage service associated with this job, in this case Amazon S3.
Resources	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 type	JobType	The type of job, either import, export, or local compute and storage.
Snowball type	SnowballType	The type of appliance used, either a Snowball or an AWS Snowball Edge appliance.
Cluster ID	ClusterId	A unique 39-character label that identifies your cluster.

Job Statuses

Each job has a *status*, which changes to denote the current state of the job. This job status information doesn't reflect the health, the current processing state, or the storage used for the associated appliances.

Console Identifier	API Identifier	Status Description
Job created	New	Your job has just been created. This status is

Console Identifier	API Identifier	Status Description
		the only one during which you can cancel a job or its job parts, if the job is an export job.
Preparing Appliance	PreparingAppliance	AWS is preparing an appliance for your job.
Exporting	InProgress	AWS is exporting your data from Amazon S3 onto an appliance.
Preparing shipment	PreparingShipment	AWS is preparing to ship an appliance to you.
In transit to you	InTransitToCustomer	The appliance has been shipped to the address you provided during job creation.
Delivered to you	WithCustomer	The appliance has arrived at the address you provided during job creation.
In transit to AWS	InTransitToAWS	You have shipped the appliance back to AWS.
At AWS	WithAWS	Your shipment has arrived at AWS. If you're importing data, your import typically begins within a day of its arrival.
Importing	InProgress	AWS is importing your data into Amazon Simple Storage Service (Amazon S3).
Completed	Complete	Your job or a part of your job has completed successfully.
Canceled	Cancelled	Your job has been canceled.

Cluster Statuses

Each cluster has a *status*, which changes to denote the current general progress state of the cluster. Each individual node of the cluster has its own job status.

This cluster status information doesn't reflect the health, the current processing state, or the storage used for the cluster or its nodes.

Console Identifier	API Identifier	Status Description
Awaiting Quorum	AwaitingQuorum	The cluster hasn't created yet, because there aren't enough nodes to begin processing the cluster request. For a cluster to be created, it needs to have at least five nodes.
Pending	Pending	Your cluster has been created, and we're getting its nodes ready to ship out. You can track the status of each node with that node's job status.
Delivered to you	InUse	At least one node of the cluster is at the address you provided during job creation.
Completed	Complete	All the nodes of the cluster have been returned to AWS.
Canceled	Cancelled	The request to make a cluster was canceled. Cluster requests can only be canceled before they enter the Pending state.

Import Jobs into Amazon S3

With an import job, your data is copied to the AWS Snowball Edge appliance with the built-in Amazon S3 Adapter for Snowball or NFS mount point. 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 100 TB AWS Snowball Edge appliance.

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 are overwritten. The import job type is also capable of local storage and compute functionality. This functionality uses the file interface or Amazon S3 Adapter for Snowball to read and write data, and triggers Lambda functions based off of Amazon S3 PUT object API actions running locally on the AWS Snowball Edge appliance.

When all of your data has been imported into the specified Amazon S3 buckets in the AWS Cloud, AWS performs a complete erasure of the appliance. This erasure follows the 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 Jobs from Amazon S3

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 an AWS Snowball Edge appliance, you can download a job report. This report alerts you to any objects that failed the transfer to the appliance. 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 appliances as it takes to complete the transfer. AWS Snowball Edge appliances for an export job's job parts are delivered one after another, with subsequent appliances shipping to you after the previous job part enters the **In transit to AWS** status.

When you copy objects into your on-premises data destination from an appliance using the Amazon S3 Adapter for Snowball or the NFS mount point, those objects are saved as files. If you copy objects into a location that already holds files, any existing files with the same names are overwritten. The export job type is also capable of local storage and compute functionality. This functionality uses the file interface or Amazon S3 Adapter for Snowball to read and write data, and triggers Lambda functions based off of Amazon S3 PUT object API actions running locally on the AWS Snowball Edge appliance.

When AWS receives a returned appliance, we completely erase it, following 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 export an entire Amazon S3 bucket or a specific range of objects keys.

Using Export Ranges

When you create an export job in the AWS Snowball Management Console or with the job management API, you can export an entire Amazon S3 bucket or a specific range of objects keys. Object key names uniquely identify objects in a bucket. If you 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 Banana Car
Aardwolf	(none)	Aardwolf Aasvogel/apple Aasvogel/banana Aasvogel/cherry Banana Car
Aar	(none)	Aardvark Aardwolf

Specified Range Beginning	Specified Range Ending	Objects in the Range That Will Be Exported
		Aasvogel/apple Aasvogel/banana Aasvogel/cherry Banana Car
car	(none)	No objects are exported, and you 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

Local Compute and Storage Only Jobs

Local compute and storage jobs enable you to use Amazon S3 and AWS Lambda powered by AWS Greengrass locally, without an internet connection. While local storage and compute functionality also exists for the import and export job types, this job type is only for local use. You can't export data from Amazon S3 onto the appliance or import data into Amazon S3 when the appliance is returned.

Local Compute Jobs

The local compute functionality is AWS Lambda powered by AWS Greengrass, and can automatically run Python-language code in response to Amazon S3 PUT object action API calls to the AWS Snowball Edge appliance. You write the Python code as a Lambda function in the Lambda console.

Buckets and Lambda functions have a one-to-one relationship, meaning that one Lambda function can be associated with one bucket when the job is created. For more information, see Using AWS Lambda with an AWS Snowball Edge (p. 58).

Local Storage Jobs

You can read and write objects to an AWS Snowball Edge appliance using the Amazon S3 Adapter for Snowball or the file interface. The adapter comes built-into the appliance, and it supports Amazon S3 REST API actions. This Amazon S3 REST API support is limited to a subset of S3 REST API actions. For more information, see Using the Amazon S3 Adapter (p. 46).

When you've finished using the appliance, return it to AWS and the appliance will be erased. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

Local Cluster Option

A cluster is a logical grouping of Snowball Edge devices, in groups of 5 to 10 devices. A cluster is created as a single job, which offers increased durability and storage size when compared to other AWS Snowball job offerings. For more information on cluster jobs, see Using an AWS Snowball Edge Cluster (p. 63).

Cloning a Job in the Console

When you first create an import job or a local compute and storage job, you might discover that you need more than one AWS Snowball Edge appliance. Because import jobs and local compute and storage jobs are associated with a single appliance, requiring more than one appliance means that you need to create more than one job. When creating additional jobs, you can go through the job creation wizard again in the console, or you can clone an existing job.

> **Note**
> Cloning a job is a shortcut available in the console to make creating additional jobs easier. If you're creating jobs with the job management API, you can simply run the job creation command again.

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

To clone a job in the console

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.

Canceling Jobs in the Console

If you need to cancel a job request or a cluster creation request for any reason, you have at least an hour after you created the request to do so. You can only cancel jobs can when they have **Job created** status. Once a job begins processing, you can no longer cancel it. Likewise, to cancel a cluster creation request you have about an hour.

To cancel a job in the console

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.

Setting Up Your AWS Access for AWS Snowball Edge

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

1. Sign Up for AWS (p. 18).
2. Create an IAM User (p. 18).

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 Snowball. You are charged only for the services that you use. For more information about pricing and fees for AWS Snowball, see AWS Snowball Edge Pricing. AWS 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 Snowball, 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 AWS 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 Permissions Required to Use the AWS Snowball Console (p. 81).

Next Step

Getting Started with an AWS Snowball Edge Appliance (p. 20)

Getting Started with an AWS Snowball Edge Appliance

With AWS Snowball using an AWS Snowball Edge appliance, you can access the storage and compute power of the AWS Cloud locally and cost effectively in places where connecting to the internet might not be an option. You can also 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 AWS Snowball Edge appliance job in the AWS Snowball Management Console. The console presents the most common workflows, separated into job types. You can find more information on specific components of the AWS Snowball Edge appliance later in this documentation. For an overview of the service as a whole, see How AWS Snowball Works with the Snowball Edge (p. 7). The getting started exercises cover the following job types:

Each of the getting started exercises assume that you use the AWS Snowball Management Console to create your job, the Snowball client to unlock the AWS Snowball Edge appliance, and the Amazon S3 Adapter for Snowball to read and write data. If you'd rather create your job programmatically with more options for the jobs you're creating, you can use the job management API. For more information, see AWS Snowball API Reference.

Before you can get started, you need to create an AWS account and an administrator user in AWS Identity and Access Management (IAM). If you already have these, then you can skip these first two steps.

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. 20). Otherwise, see Sign Up for AWS (p. 18).

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. 18).

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

Next:

- Getting Started with AWS Snowball Edge: Your First Job (p. 20)

Getting Started with AWS Snowball Edge: Your First Job

Regardless of the type of job you're creating for an AWS Snowball Edge appliance, there's a set of common procedures in the AWS Snowball Management Console. These are procedures for creating a job,

monitoring the status of your job, setting up the appliance, unlocking the appliance, transferring data, and returning the appliance. Following, you can walk through these procedures. Whenever there is a job-type specific consideration, we call out that consideration in a note.

Topics

Create Your First Job

The process for creating an AWS Snowball Edge appliance job in the AWS Snowball Management Console has the following steps.

To create an AWS Snowball Edge appliance job in the console

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console.
2. Choose **Create Job**.
3. On the **Plan your job** page of the job creation wizard, choose your job type.

   Job-Type Specific Consideration

   - If you are creating a cluster, select the **Make this a cluster** check box.
4. Choose **Next**.
5. On the **Give shipping details** page, provide the shipping address that you want the AWS Snowball Edge appliance for this job delivered to. In some regions, you choose your shipping speed at this point. For more information, see Shipping Speeds (p. 71).
6. Choose **Next**.
7. On the **Give job details** page, provide the details for your job, including a name, region, and at least one bucket.

   Important
   With an AWS Snowball Edge appliance, all jobs can use the compute functionality in regions where Lambda is supported. How the compute functionality is implemented in AWS Snowball jobs is specific to Snowball—the functionality can differ significantly from how Lambda works in the cloud. Before creating your first compute job, we recommend that you familiarize yourself with how AWS Lambda powered by AWS Greengrass works. For more information, see Using AWS Lambda with an AWS Snowball Edge (p. 58).
8. Choose **Next**.
9. On the **Set security** page, specify the following:

- The Amazon Resource Name (ARN) for the AWS Identity and Access Management (IAM) role that AWS Snowball assumes to import your data to your destination S3 bucket when you return the AWS Snowball Edge appliance.
- The ARN for the AWS Key Management Service (AWS KMS) master key to be used to protect your data within the AWS Snowball Edge appliance. For more information, see Security for AWS Snowball Edge (p. 72).

10. Choose **Next**.
11. On the **Set notifications** 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 Notifications for the AWS Snowball Edge (p. 90).
12. Choose **Next**.
13. On the **Review** page, review the information you've provided. To make changes, choose **Edit** 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 Job Statuses (p. 11). 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 job, AWS processes the information you provided and prepares an AWS Snowball Edge appliance specifically for your job. During the processing stage, if there's an issue with your job, we contact you by email. Otherwise, we ship the appliance 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 appliance we prepared for your job. In your job's details, you should see a link to the tracking webpage with your tracking number provided.

Job-Type Specific Consideration

For an export job, once the Snowball is prepared, the status for your first job part becomes **Exporting**. Exporting typically takes one business day; however, it can take longer on occasion. 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 AWS Snowball Edge appliance, and also success and failure logs. For more information, see Get Your Job Completion Report and Logs in the Console (p. 27).

Next: Receive the Snowball Edge (p. 22)

Receive the Snowball Edge

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

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

The appliance looks like the following image.

If you're ready to connect the appliance to your internal network, see the next section.

Next: Connect to Your Local Network (p. 24)

Connect to Your Local Network

In this procedure, you connect the AWS Snowball Edge appliance to your local network. The appliance doesn't need to be connected to the internet. The appliance has three doors, a front, a back, and a top.

To connect the appliance to your network

1. Open the front and back doors, sliding them inside the appliance door slots. Doing this gives you access to the touch screen on the LCD display embedded in the front of the appliance, and the power and network ports in the back.
2. Open the top door and remove the provided power cable from the cable nook, and plug the appliance into power.
3. Choose one of your RJ45, SFP+, or QSFP+ network cables, and plug the appliance into your network. The network ports are on the back of the appliance.
4. Power on the AWS Snowball Edge appliance by pressing the power button above the LCD display.
5. When the appliance is ready, the LCD display shows a short video while the appliance is getting ready to start. After about ten minutes, the appliance is ready to be unlocked.
6. (Optional) Change the default network settings through the LCD display by choosing **CONNECTION**. To learn more about specifying network settings for the appliance, see Changing Your IP Address (p. 36).

The appliance is now connected to your network.

> **Important**
> To prevent corrupting your data, don't disconnect the AWS Snowball Edge appliance or change its connection settings while it's in use.

Next: Get Your Credentials and Tools (p. 24)

Get Your Credentials and Tools

Each 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.

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

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 choose **View job details**.
4. In the details pane that appears, expand **Credentials** and then do the following:
   - Make a note of the unlock code (including the hyphens), because you 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
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 the AWS Snowball Edge Appliance (p. 29).

Now that you have your credentials, the next step is to download the Snowball client, which is used to unlock the AWS Snowball Edge appliance.

Next: Download and Install the Snowball Client (p. 25)

Download and Install the Snowball Client

The Snowball client is the tool that you use to unlock the AWS Snowball Edge appliance. You can download the Snowball client for your operating system from the AWS Snowball Tools Download page.

Next: Unlock the Snowball Edge (p. 25)

Unlock the Snowball Edge

To unlock the AWS Snowball Edge appliance, run the `snowballEdge unlock-device` command. To run this command, the AWS Snowball Edge appliance that you use for your job must be onsite, plugged into power and network, and turned on. In addition, the LCD display on the AWS Snowball Edge appliance's front must indicate that the appliance is ready for use.

To unlock the appliance with the Snowball client

1. Get your manifest and unlock code.
   a. Download a copy of the manifest from the AWS Snowball Management Console. Your job's manifest is encrypted so that only the job's unlock code can decrypt it. Make a note of the path to the manifest file on your local server.
   b. Get the unlock code, a 29-character code that also appears when you download your manifest. We recommend that you write down the unlock code and keep it in a separate location from the manifest that you downloaded, to prevent unauthorized access to the AWS Snowball Edge appliance while it's at your facility.
2. Find the IP address for the AWS Snowball Edge appliance on the AWS Snowball Edge appliance's LCD display, under the **Connections** tab. Make a note of that IP address.
3. Run the `snowballEdge unlock-device` command to authenticate your access to the AWS Snowball Edge appliance with the AWS Snowball Edge appliance's endpoint and your credentials, as follows:

```
snowballEdge unlock-device --endpoint https://ip address --manifest-file Path/to/
manifest/file --unlock-code
 29 character unlock code
```

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

```
snowballEdge unlock-device --endpoint https://192.0.2.0 --manifest-file /Downloads/
JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin  --unlock-code 12345-abcde-12345-
ABCDE-12345
```

In this example, the IP address for the appliance 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 appliance for this job.

Job-Type Specific Consideration

If you're unlocking a cluster, you need to specify only the IP address for one of the cluster nodes. For more information, see Using the Snowball Client (p. 36).

Now you can begin using the AWS Snowball Edge appliance.

Next: Use the Snowball Edge (p. 26)

Use the Snowball Edge

Now you can use the AWS Snowball Edge appliance. Regardless of your job type, keep the following information in mind while using the appliance:

- You can use the built-in Amazon S3 Adapter for Snowball to read or write data to an appliance or cluster of appliances. The adapter can work through the AWS Command Line Interface (AWS CLI), one of the AWS SDKs, or your own RESTful application. For more information on the adapter, see Using the Amazon S3 Adapter (p. 46).
- You can use the file interface to read or write data to an appliance or a cluster of appliances. For more information, see Using the File Interface for the AWS Snowball Edge (p. 52).
- There is at least one directory at the root level of the appliance. This directory and any others at the root level have the same names as the 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 these bucket directories or into their subdirectories.

When you're done using the AWS Snowball Edge appliance, it's time to prepare the appliance for its return trip to AWS.

Next: Stop the Snowball Client, and Power Off the Snowball Edge (p. 26)

Stop the Snowball Client, and Power Off the Snowball Edge

When you've finished transferring data on to the AWS Snowball Edge appliance, prepare it for its return trip to AWS. Before you continue, make sure that all data transfer to the appliance has stopped. If you were using the file interface to transfer data, you need to disable it before you power off the appliance. For more information, see Disabling the File Interface (p. 57).

When all communication with the appliance has ended, simply turn it off by pressing the power button above the LCD display. It takes about 20 seconds for the appliance to shut down.

Next: Disconnect the Snowball Edge (p. 26)

Disconnect the Snowball Edge

Disconnect the AWS Snowball Edge appliance cables. Secure the appliance's power cable into the cable nook beneath the top door on the appliance. Pull out and close the front and back doors. When they close completely, you hear an audible click. When the return shipping label appears on the E Ink display on top of the appliance, it's ready to be returned. To see who your region's carrier is, see Shipping Carriers (p. 69).

Job-Type Specific Consideration

If you are importing data, 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 Snowball Edge (p. 27)

Return the Snowball Edge

The prepaid shipping label on the E Ink display contains the correct address to return the AWS Snowball Edge appliance. For information on how to return the appliance, see Shipping Carriers (p. 69). The appliance is delivered to an AWS sorting facility and forwarded to the AWS data center. The carrier automatically reports 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 appliance. Always use the shipping label that is displayed on the 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 AWS Snowball Edge appliance has been received by AWS, when data import begins, and when the job is completed.

Job-Type Specific Consideration

If you are exporting data or if you are using an appliance for a local storage and compute only job, then your job is now complete. For information on what to do next, see Where Do I Go from Here? (p. 28).

Next: Monitor the Import Status (p. 27)

Monitor the Import Status

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 appliance arrives at AWS, 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 AWS 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 Get Your Job Completion Report and Logs in the Console (p. 27).

Next: Get Your Job Completion Report and Logs in the Console (p. 27)

Get Your Job Completion Report and Logs in the Console

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 very end of the import process. For export jobs,

your job report typically becomes available for you while the AWS Snowball Edge appliance 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 appliance 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 the log you want to 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.

Next: Where Do I Go from Here? (p. 28)

Where Do I Go from Here?

Now that you've read through one of the previous getting started sections and begun your first job, you can learn more about using the AWS Snowball Edge appliance tools and interfaces in detail from the following topics:

- Using an AWS Snowball Edge (p. 34)
- Using the Snowball Client (p. 36)
- Using the Amazon S3 Adapter (p. 46)

We also recommend that you learn about 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 the AWS Snowball Edge Appliance

To help get the maximum benefit from and satisfaction with your AWS Snowball Edge appliance, we recommend that you follow these best practices.

Security

- If you notice anything that looks suspicious about the AWS Snowball Edge appliance, don't connect it to your internal network. Instead, contact AWS Support, and a new AWS Snowball Edge appliance 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 these separately helps prevent unauthorized parties from gaining access to the AWS Snowball Edge appliance. For example, you can save a copy of the manifest to your local server, and email the code to a user that unlocks the appliance. This approach limits access to the AWS Snowball Edge appliance to individuals who have access to files saved on the server and also that user's email address.
- The credentials displayed when you run the Snowball client command `snowballEdge credentials` are a pair of keys: an access key and a secret key. These keys are only associated with the job and the local resources on the appliance. They don't map to your AWS account or any other AWS account. If you try to use these keys to access services and resources in the AWS Cloud, they fail, because they work only for the local resources associated with your job.

Network

- We recommend that you only use one method of reading and writing data to a local bucket on an AWS Snowball Edge appliance at a time. Using both the file interface and the Amazon S3 Adapter for Snowball on the same bucket at the same time can result in read/write conflicts.
- To prevent corrupting your data, don't disconnect an AWS Snowball Edge appliance or change its network settings while transferring data.
- Files should be in a static state while being written to the appliance. Files that are modified while they are being written can result in read/write conflicts.
- For more information about improving performance of your AWS Snowball Edge appliance, see Performance (p. 29).

Resource Management

- The 10 free days for performing your on-premises data transfer start the day after the AWS Snowball Edge appliance arrives at your data center.
- 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. This functionality is also true for clusters.
- 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.

Performance

Following, you can find information about AWS Snowball Edge appliance 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 Edge with data. Transferring smaller files reduces your transfer speed due to decreased overhead. If you have many small files, we recommend that you zip them up into larger archives before transferring them onto a Snowball.

Rate (MB/s)	82 TB Transfer Time
800	1.22 days
450	2.11 days
400	2.37 days
300	3.16 days
277	3.42 days
200	4.75 days
100	9.49 days
60	15.53 days
30	31.06 days
10	85.42 days

To provide meaningful guidance about performance, the following sections describe how to determine when to use the AWS Snowball Edge appliance and how to get the most out of the service.

Performance Recommendations

The following recommendations are highly suggested, because they have the largest impact in improving the performance of your data transfer:

- We recommend that you have no more than 500,000 files or directories within each directory.
- We recommend that all files transferred to a Snowball be no smaller than 1 MB in size.
- If you have many files smaller than 1 MB in size each, we recommend that you zip them up into larger archives before transferring them onto a Snowball.

Speeding Up Data Transfer

One of the major ways that you can improve the performance of an AWS Snowball Edge appliance is to speed up the transfer of data going to and from an appliance. In general, you can improve the transfer speed from your data source to the appliance in the following ways, ordered from largest to smallest positive impact on performance:

1. **Perform multiple write operations at one time** – You can perform multiple write operations at one time. You can do this by running each command from multiple terminal windows on a computer with a network connection to a single AWS Snowball Edge appliance.
2. **Transfer small files in batches** – Each copy operation has some overhead because of encryption. To speed the process up, batch files together in a single archive. When you batch files together, they can

be auto-extracted when they are imported into Amazon S3. For more information, see Batching Small Files (p. 47).

3. **Write from multiple computers** – A single AWS Snowball Edge appliance can be connected to many computers on a network. Each computer can connect to any of the three network interfaces at once.
4. **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 negative impact on transfer performance. We recommend that your files remain in a static state while you transfer them.
5. **Reduce local network use** – Your AWS Snowball Edge appliance communicates across your local network. Because of this, reducing other local network traffic between the AWS Snowball Edge appliance, the switch it's connected to, and the computer that hosts your data source can improve data transfer speeds.
6. **Eliminate unnecessary hops** – We recommend that you set up your AWS Snowball Edge appliance, your data source, and the computer running the terminal connection between them so that they're the only machines communicating across a single switch. Doing so can result in improved data transfer speeds.

How to Transfer Petabytes of Data Efficiently

When transferring petabytes of data, we recommend that you plan and calibrate your data transfer between the AWS Snowball Edge appliances you have on-site and your servers according to the following guidelines.

Planning Your Large Transfer

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

Topics

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

Before you create your first job using AWS Snowball Edge appliances, you should make sure 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, this administrative housekeeping should make your life much easier when your AWS Snowball Edge appliances 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 AWS 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 should know the total amount of data that you're going to move into the cloud.

Step 2: Calculate Your Target Transfer Rate

It's important to estimate how quickly you can transfer data to the AWS Snowball Edge appliances connected to each of your servers. This estimated speed equals your target transfer rate. This rate is the

rate at which you can expect data to move into an AWS Snowball Edge appliance given the realities of your local network architecture.

> **Note**
> For large data transfers, we recommend using the Amazon S3 Adapter for Snowball to transfer your data.

To calculate your target transfer rate, transfer a representative subset of your data to a Snowball Edge. During the transfer, run the `snowballEdge status` command to track the progress of the transfer. When the transfer is complete, compare the size of the transferred data against the time it took the transfer to complete to get an estimate of your current transfer speed.

While determining your target transfer speed, keep in mind that you can change the speed through changes to network speed, the size of the files being transferred, and the speed at which data can be read from your local servers. The Amazon S3 Adapter for Snowball copies data to the AWS Snowball Edge appliance as fast as conditions allow.

Step 3: Determine How Many AWS Snowball Edge Appliances You Need

Using the total amount of data you're going to move into the cloud, found in step 1, the transfer speed you estimated from step 2, and the number of days in which you want to move the data into AWS, determine how many AWS Snowball Edge appliances you need to finish your large-scale data migration. Remember that AWS Snowball Edge appliances have about 73 TB of usable space. So if you want to move 300 TB of data into AWS in 10 days, and you have a transfer speed of 250 MB/s, you would need five AWS Snowball Edge appliances.

Step 4: Create Your Jobs

Now that you know how many AWS Snowball Edge appliances you need, you can create an import job for each appliance. Because each AWS Snowball Edge appliance import job involves a single AWS Snowball Edge appliance, you have to create multiple import jobs. For more information, see Create Your First Job (p. 21).

Step 5: 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 AWS Snowball Edge appliance 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 9 segments of 10 TB each for an AWS Snowball Edge appliance.
- For large files, each file can be an individual segment, keeping in mind the 5 TB size limit for objects in Amazon S3.
- Each segment can be a different size, and each individual segment can be made of the same kind of data—for example, small files in one segment, compressed archives in another, 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 compressing small files into a larger bundle, batching your files, or transferring larger individual files.

Creating these data transfer segments makes it easier for you to quickly resolve any transfer issues, because trying to troubleshoot a large, heterogeneous 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 AWS Snowball Edge appliance from your server to calibrate your speed and total transfer time.

Calibrating a Large Transfer

You can calibrate a large transfer by transferring 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 an AWS Snowball Edge appliance. At the same time, make a record of the transfer speed and total transfer time for each operation.

While the calibration is being performed, monitor the information that comes from the `snowballEdge status` command. If the calibration's results are less than the target transfer rate, you might be able to copy multiple parts of your data transfer at the same time. In this case, repeat the calibration with additional data transfer segment.

Continue adding additional parallel copy operations during calibration until you see diminishing returns in the sum of the transfer speed of all instances currently transferring data. At this point, you can end the last active instance and make a note of your new target transfer rate.

Sometimes the fastest way to transfer data with the AWS Snowball Edge appliance is to transfer data in parallel in one of the following scenarios:

- Using multiple instances of the Amazon S3 Adapter for Snowball on a single AWS Snowball Edge appliance.
- Using multiple instances of the Amazon S3 Adapter for Snowball on multiple AWS Snowball Edge appliances.

When you complete these steps, you should know how quickly you can transfer data to an AWS Snowball Edge appliance.

Using an AWS Snowball Edge

Following, you can find an overview of the AWS Snowball Edge appliance, the physically rugged appliance protected by AWS Key Management Service (AWS KMS) that you'll use for local storage and compute, or to transfer data between your on-premises servers and Amazon Simple Storage Service (Amazon S3).

For information on unlocking an AWS Snowball Edge appliance, see Using the Snowball Client (p. 36).

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

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

The following image shows what the AWS Snowball Edge appliance looks like.

It has three doors, a front, a back, and a top that all can be opened by latches. The top door contains the power cable for the appliance. The other two doors can be opened and slid inside the appliance so they're out of the way while you're using it.

Doing this gives you access to the LCD E Ink display embedded in the front side of the appliance, and the power and network ports in the back.

Plug the AWS Snowball Edge appliance into power, and into your local network using one of your network cables. Each AWS Snowball Edge appliance has been engineered to support data transfer over RJ45, SFP+ , or QSFP+, and is capable of network speeds of over 10 GB/second. Power on the AWS Snowball Edge appliance by pressing the power button above the LCD display.

You'll hear the AWS Snowball Edge appliance internal fans start up, and the display starts up with a small video. The fans will get quieter after a short time. Wait a few moments, and a screen appear that

indicates that the appliance is ready. When that happens, the AWS Snowball Edge appliance is ready to communicate with the Snowball client, the tool used to unlock your access to it.

You use the LCD display to get some of the local management information for the services on the appliance, and also to manage the IP address that the AWS Snowball Edge appliance 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 AWS Snowball Edge appliance

1. On the LCD display, tap **CONNECTION**. You'll see a screen that shows you the current network settings for the AWS Snowball Edge appliance.
2. The IP address below this drop-down box is updated to reflect the DHCP address that the AWS Snowball Edge appliance requested. You can change it to a static IP address, or leave it as is.

Using the Snowball Client

Following, you can find information about how to get and use the Snowball client with your AWS Snowball Edge appliance. The Snowball client is a standalone terminal application that you run on your local server to unlock the appliance and get credentials, logs, and status information. You can also use the client for administrative tasks for a cluster. While using the Snowball client, you can get additional support information by running the `snowballEdge help` command.

When you read and write data to the AWS Snowball Edge appliance, you use the Amazon S3 Adapter for Snowball or the file interface.

> **Note**
> In January 2018, there was a feature update for clusters, making them leaderless. The cluster update is backward-compatible with older clusters. However, if you're looking for documentation for the original Snowball client for Snowball Edge, see Using the Snowball Client (p. 101) in the Appendices.

Downloading and Installing the Snowball Client

You can download and install the Snowball client from the AWS Snowball Tools Download page. On that page, you can 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:

- **Microsoft Windows** – When the client has been installed, you can run it from any directory without any additional preparation.
- **Linux** – The Snowball client must be run from the ~/snowball-client-linux-*build_number*/bin/ directory.
- **Mac** – The `install.sh` script copies 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 if /usr/local/bin is a path in your bash_profile. You can verify your path with the `echo $PATH` command.

Commands for the Snowball Client

Following, you can find information on the Snowball client commands, including examples of use and sample outputs.

Topics

Configuring a Profile for the Snowball Client

Every time you run a command for the Snowball client, you need to provide your manifest file, unlock code, and an IP address. You can get the first two of these from the AWS Snowball Management Console or the job management API. For more information on getting your manifest and unlock code, see Get Your Credentials and Tools (p. 24).

You have the option of using the `snowballEdge configure` command to store the path to the manifest, the 29-character unlock code, and the endpoint as a profile. After configuration, you can use other Snowball client commands without having to manually type in these values for a particular job. After you configure the Snowball client, the information is saved in a plain-text JSON format to *home directory*`/.aws/snowball/config/snowball-edge.config`.

The endpoint is the IP address, with `https://` appended to it. You can locate the IP address for the AWS Snowball Edge appliance on the AWS Snowball Edge appliance's LCD display. When the AWS Snowball Edge appliance is connected to your network for the first time, it automatically gets a DHCP IP address, if a DHCP server is available. If you want to use a different IP address, you can change it from the LCD display. For more information, see Using an AWS Snowball Edge (p. 34).

> **Important**
> Anyone who can access the configuration file can access the data on your Snowball Edge devices or clusters. Managing local access control for this file is one of your administrative responsibilities.

Usage

You can use this command in two ways: inline, or when prompted. This usage example shows the prompted method.

```
snowballEdge configure
```

Example Output

```
Configuration will stored at home directory\.aws\snowball\config\snowball-edge.config
Snowball Edge Manifest Path: Path/to/manifest/file
Unlock Code: 29 character unlock code
Default Endpoint: https://192.0.2.0
```

You can have multiple profiles if you have multiple jobs at once, or if you want the option of managing a cluster from different endpoints. For more info on multiple AWS CLI profiles, see Named Profiles in the AWS Command Line Interface User Guide.

Unlocking AWS Snowball Edge Devices

To unlock a standalone AWS Snowball Edge appliance, run the `snowballEdge unlock-device` command. To unlock a cluster, use the `snowballEdge unlock-cluster` command. These commands authenticate your access to the AWS Snowball Edge appliance.

> **Note**
> To unlock the devices associated with your job, the devices must be onsite, plugged into power and network, and turned on. In addition, the LCD display on the AWS Snowball Edge appliance's front must indicate that the appliance is ready for use.

Usage (Snowball client not configured)

```
snowballEdge unlock-device --endpoint https://ip address --manifest-file Path/to/manifest/
file --unlock-code
 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge unlock-device
```

Example Single Device Unlock Input

```
snowballEdge unlock-device
```

Example Single Device Unlock Output

```
Your Snowball Edge device is unlocking. You may determine the unlock state of your device
 using the describe-device command. Your Snowball Edge device will be available for use
 when it is in the UNLOCKED state.
```

Cluster Usage

When you unlock a cluster, you need to provide the endpoint for one of your nodes, and all the IP addresses for the other devices in your cluster.

```
snowballEdge unlock-cluster --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/
file --unlock-code 01234-abcde-ABCDE-01234 --device-ip-addresses 192.0.2.0 192.0.2.1
 192.0.2.2 192.0.2.3 192.0.2.4
```

Example Cluster Unlock Output

```
Your Snowball Edge Cluster is unlocking. You may determine the unlock state of your cluster
 using the describe-device command. Your Snowball Edge Cluster will be available for use
 when your Snowball Edge devices are in the UNLOCKED state.
```

Getting Credentials

Using the `snowballEdge list-access-keys` and `snowballEdge get-secret-access-key` commands, you can get your local credentials. You use these to authenticate your requests when using the AWS CLI or with an AWS SDK. These credentials are only associated with an individual job

for Snowball Edge, and you can use them only on the device or cluster of devices. The appliance or appliances don't have any AWS Identity and Access Management (IAM) permissions in the AWS Cloud.

> **Note**
> If you're using the AWS CLI with the Snowball Edge, you must use these credentials when you configure the CLI. For information on configuring credentials for the CLI, see Quick Configuration in the *AWS Command Line Interface User Guide*.

Usage (Snowball client not configured)

```
snowballEdge list-access-keys --endpoint https://192.0.2.0 --manifest-file Path/to/
manifest/file --unlock-code 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge list-access-keys
```

Example Output

```
{
  "AccessKeyIds" : [ "AKIAIOSFODNN7EXAMPLE" ]
}
```

Usage (Snowball client not configured)

```
snowballEdge get-secret-access-key --access-key-id Access Key --endpoint https://192.0.2.0
 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge get-secret-access-key --access-key-id Access Key
```

Example Output

```
[snowballEdge]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Getting Your Certificate for Transferring Data

To transfer data to a Snowball Edge, you need to use the Amazon S3 Adapter for Snowball. To use the adapter over the HTTPS protocol, you must provide a certificate. The certificates are generated by each Snowball Edge device. If you unlock your Snowball Edge device with a different IP address, a new certificate is generated and the old certificate is no longer valid to use with the endpoint. You'll have to get the new, updated certificate from the Snowball Edge again using the `get-certificate` command.

You can list these certificates and download them from your Snowball Edge device with the following commands:

- `list-certificates` – Lists the Amazon Resource Names (ARNs) for the certificates available for use.

  Usage (Snowball client not configured)

  ```
  snowballEdge list-certificates --endpoint https://192.0.2.0 --manifest-file Path/to/
  manifest/file --unlock-code 29 character unlock code
  ```

Usage (configured Snowball client)

```
snowballEdge list-certificates
```

Example Output

```
{
  "Certificates" : [ {
    "CertificateArn" : "arn:aws:snowball-
device:::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7",
    "SubjectAlternativeNames" : [ "192.0.2.0" ]
  } ]
}
```

- `get-certificate` – Gets a specific certificate, based on the ARN provided.

Usage (Snowball client not configured)

```
snowballEdge get-certificate --certificate-arn arn:aws:snowball-
device:::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7 --endpoint https://192.0.2.0 --
manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge get-certificate --certificate-arn arn:aws:snowball-
device:::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7
```

Example Output

```
-----BEGIN CERTIFICATE-----
Certificate
-----END CERTIFICATE-----
```

For information on configuring your certificate, see Specifying the Adapter as the AWS CLI Endpoint (p. 47).

AWS Snowball Edge Logs

When you transfer data between your on-premises data center and an Snowball Edge, logs are automatically generated. If you encounter unexpected errors during data transfer to the device, you can use the following commands to save a copy of the logs to your local server.

There are three commands related to logs:

- `list-logs` – Returns a list of logs in JSON format. This list reports the size of the logs in bytes, the ARN for the logs, the service ID for the logs, and the type of logs.

Usage (Snowball client not configured)

```
snowballEdge list-logs --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file
 --unlock-code 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge list-logs
```

Example Output

```
{
  "Logs" : [ {
    "LogArn" : "arn:aws:snowball-device:::log/s3-storage-JIEXAMPLE2f-1234-4953-a7c4-
dfEXAMPLE709",
    "LogType" : "SUPPORT",
    "ServiceId" : "s3",
    "EstimatedSizeBytes" : 53132614
  }, {
    "LogArn" : "arn:aws:snowball-device:::log/fileinterface-JIDEXAMPLEf-1234-4953-a7c4-
dfEXAMPLE709",
    "LogType" : "CUSTOMER",
    "ServiceId" : "fileinterface",
    "EstimatedSizeBytes" : 4446
  }]
}
```

- `get-log` – Downloads a copy of a specific log from the Snowball Edge to your server at a specified path. `CUSTOMER` logs are saved in the `.zip` format, and you can extract this type of log to view its contents. `SUPPORT` logs are encrypted and can only be read by AWS Support engineers. You have the option of specifying a name and a path for the log.

  Usage (Snowball client not configured)

  ```
  snowballEdge get-log --log-arn arn:aws:snowball-device:::log/fileinterface-
  JIDEXAMPLEf-1234-4953-a7c4-dfEXAMPLE709 --endpoint https://192.0.2.0 --manifest-
  file Path/to/manifest/file --unlock-code 29 character unlock code
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge get-log --log-arn arn:aws:snowball-device:::log/fileinterface-
  JIDEXAMPLEf-1234-4953-a7c4-dfEXAMPLE709
  ```

  Example Output

  ```
  Logs are being saved to download/path/snowball-edge-logs-1515EXAMPLE88.bin
  ```

- `get-support-logs` – Downloads a copy of all the `SUPPORT` type of logs from the Snowball Edge to your service at a specified path.

  Usage (Snowball client not configured)

  ```
  snowballEdge get-support-logs --endpoint https://192.0.2.0 --manifest-file Path/to/
  manifest/file --unlock-code 29 character unlock code
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge get-support-logs
  ```

Example Output

```
Logs are being saved to download/path/snowball-edge-logs-1515716135711.bin
```

Important
`CUSTOMER` type might contain sensitive information about your own data. To protect this potentially sensitive information, we strongly suggest that you delete these logs once you're done with them.

Getting Device Status

You can determine the status and general health of your Snowball Edge devices with the following Snowball client commands:

- `describe-device`

  Usage (Snowball client not configured)

  ```
  snowballEdge describe-device --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge describe-device
  ```

  Example Output

  ```
  {
    "DeviceId" : "JIDEXAMPLEf-31a7-4953-a7c4-dfcEXAMPLE09",
    "UnlockStatus" : {
      "State" : "UNLOCKED"
    },
    "ActiveNetworkInterface" : {
      "IpAddress" : "192.0.2.0"
    }
  }
  ```

- `describe-cluster`

  Usage (Snowball client not configured)

  ```
  snowballEdge describe-cluster --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge describe-cluster
  ```

  Example Output

  ```
  {
    "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5",
    "Devices" : [ {
      "DeviceId" : "JIDEXAMPLE2-bc53-4618-a538-917EXAMPLE94",
  ```

```
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.0.2.0"
  },
  "ClusterAssociation" : {
    "State" : "ASSOCIATED",
    "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5"
  },
  "NetworkReachability" : {
    "State" : "REACHABLE"
  }
}, {
  "DeviceId" : "JIDEXAMPLE2-bc53-4618-a538-917EXAMPLE94",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.0.2.1"
  },
  "ClusterAssociation" : {
    "State" : "ASSOCIATED",
    "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5"
  },
  "NetworkReachability" : {
    "State" : "REACHABLE"
  }
}, {
  "DeviceId" : "JIDEXAMPLE2-bc53-4618-a538-917EXAMPLE94",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.0.2.2"
  },
  "ClusterAssociation" : {
    "State" : "ASSOCIATED",
    "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5"
  },
  "NetworkReachability" : {
    "State" : "REACHABLE"
  }
}, {
  "DeviceId" : "JIDEXAMPLE2-bc53-4618-a538-917EXAMPLE94",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.0.2.3"
  },
  "ClusterAssociation" : {
    "State" : "ASSOCIATED",
    "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5"
  },
  "NetworkReachability" : {
    "State" : "REACHABLE"
  }
}, {
  "DeviceId" : "JIDEXAMPLE2-bc53-4618-a538-917EXAMPLE94",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.0.2.4"
  },
```

```
    "ClusterAssociation" : {
      "State" : "ASSOCIATED",
      "ClusterId" : "CIDEXAMPLE7-5402-4c19-9feb-7c9EXAMPLEd5"
    },
    "NetworkReachability" : {
      "State" : "REACHABLE"
    }
  } ]
}
```

Getting Service Status

You can determine the status and general health of the services running on Snowball Edge devices with the `describe-service` command. You can first run the `list-services` command to see what services are running.

- `list-services`

  Usage (Snowball client not configured)

  ```
  snowballEdge list-services --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge list-services
  ```

  Example Output

  ```
  {
    "ServiceIds" : [ “greengrass”, "fileinterface", "s3" ]
  }
  ```

- `describe-service`

  Usage (Snowball client not configured)

  ```
  snowballEdge describe-service --endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code --service-id service-id
  ```

  Usage (configured Snowball client)

  ```
  snowballEdge describe-service --service-id service-id
  ```

  Example Output

  ```
  {
  "ServiceId" : "s3",
  "Storage" : {
  "TotalSpaceBytes" : 99608745492480,
  "FreeSpaceBytes" : 99608744468480
  },
  "Endpoints" : [ {
  "Protocol" : "http",
  "Port" : 8080,
  "Host" : "192.0.2.0"
  ```

```
}, {
"Protocol" : "https",
"Port" : 8443,
"Host" : "192.0.2.0",
"CertificateAssociation" : {
"CertificateArn" : "arn:aws:snowball-
device:::certificate/6d955EXAMPLEdb71798146EXAMPLE3f0"
}
} ]
}
```

Removing a Node from a Cluster

The `disassociate-device` command removes a node from a Snowball Edge cluster. If you want to replace an unhealthy node, use this command. For more information on clusters, see Using an AWS Snowball Edge Cluster (p. 63).

> **Important**
> Use the `disassociate-device` command only when you are removing an unhealthy node. This command fails and returns an error if you try to remove a healthy node.

Don't use this command to remove a node that was accidentally powered off or disconnected from the network and is therefore temporarily unavailable to the rest of the cluster. Nodes removed with this command can't be added to any cluster, and must be returned to AWS.

If a node that was accidentally powered off or disconnected from the network simply plug the previously unavailable node back into power and the network and use the `associate-device` command. You can't use the `disassociate-device` command to disassociate a node if it's powered on and healthy.

Usage (Snowball client not configured)

```
snowballEdge disassociate-device --device-id Job ID for the Device --
endpoint https://192.0.2.0 --manifest-file Path/to/manifest/file --unlock-code 29 character
 unlock code
```

Usage (configured Snowball client)

```
snowballEdge disassociate-device --device-id Job ID for the Device
```

Example Output

```
Disassociating your Snowball Edge device from the cluster. Your Snowball Edge device will
 be disassociated from the cluster when it is in the "DISASSOCIATED" state. You can use the
 describe-cluster command to determine the state of your cluster.
```

Adding a Node to a Cluster

The `associate-device` command adds a node to a cluster of Snowball Edge devices. If you power off a node, then it reverts from being unlocked to being locked. To unlock that node, you can use this command. You can use this command to replace an unavailable node with a new node that you ordered as a replacement. For more information on clusters, see Using an AWS Snowball Edge Cluster (p. 63).

Usage (Snowball client not configured)

```
snowballEdge associate-device --device-ip-address IP Address --endpoint https://192.0.2.0
 --manifest-file Path/to/manifest/file --unlock-code 29 character unlock code
```

Usage (configured Snowball client)

```
snowballEdge associate-device --device-ip-address IP Address
```

Example Output

```
Associating your Snowball Edge device with the cluster. Your Snowball Edge device will be
 associated with the cluster when it is in the ASSOCIATED state. You can use the describe-
cluster command to determine the state of your cluster.
```

Using the Amazon S3 Adapter

Following, you can find an overview of the Amazon S3 Adapter for Snowball, which allows you to programmatically transfer data to and from the AWS Snowball Edge appliance using Amazon S3 REST API actions. This Amazon S3 REST API support is limited to a subset of actions. You can use this subset of actions with one of the AWS SDKs to transfer data programmatically. You can also use the subset of supported AWS Command Line Interface (AWS CLI) commands for Amazon S3 to transfer data programmatically.

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.

> **Important**
> We recommend that you only use one method of reading and writing data to a local bucket on an AWS Snowball Edge appliance at a time. Using both the file interface and the Amazon S3 Adapter for Snowball on the same bucket at the same time can result in read/write conflicts.

Getting and Using Local Amazon S3 Credentials

Every interaction with a Snowball Edge is signed with the AWS Signature Version 4 algorithm. For more information on the algorithm, see Signature Version 4 Signing Process.

You can get the local Amazon S3 credentials to sign your requests to the Snowball Edge device by running the `snowballEdge list-access-keys` and `snowballEdge get-secret-access-key` Snowball client commands. For more information, see Getting Credentials (p. 38). These local Amazon S3 credentials include a pair of keys: an access key ID and a secret key. These credentials are only valid for the appliances associated with your job. They can't be used in the AWS Cloud because they have no AWS Identity and Access Management (IAM) counterpart.

You can add these credentials to the AWS credentials file on your server. The default credential profiles file is typically located at `~/.aws/credentials`, but the location can vary per platform. This file is shared by many of the AWS SDKs and by the AWS CLI. You can save local credentials with a profile name as in the following example.

```
[snowballEdge]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Specifying the Adapter as the AWS CLI Endpoint

When you use the AWS CLI to issue a command to the AWS Snowball Edge appliance, you specify that the endpoint is the Amazon S3 Adapter for Snowball. You have the choice of using the HTTPS endpoint, or an unsecured HTTP endpoint, as shown following.

HTTPS secured endpoint

```
aws s3 ls --profile snowballEdge --endpoint https://192.0.2.0:8443 --ca-bundle path/to/certificate
```

HTTP unsecured endpoint

```
aws s3 ls --profile snowballEdge --endpoint http://192.0.2.0:8080
```

If you use the HTTPS endpoint of `8443`, your data is securely transferred from your server to the Snowball Edge. This encryption is ensured with a certificate that's generated by the Snowball Edge whenever it gets a new IP address. After you have your certificate, you can save it to a local `ca-bundle.pem` file. Then you can configure your AWS CLI profile to include the path to your certificate, as described following.

To associate your certificate with the adapter endpoint

1. Connect the Snowball Edge to power and network, and turn it on.
2. After the device has finished booting up, make a note of its IP address on your local network.
3. From a terminal on your network, make sure you can ping the Snowball Edge.
4. Run the `snowballEdge get-certificate` command in your terminal. For more information on this command, see Getting Your Certificate for Transferring Data (p. 39).
5. Save the output of the `snowballEdge get-certificate` command to a file, for example `ca-bundle.pem`.
6. Run the following command from your terminal.

   ```
   aws configure set snowballEdge.ca_bundle /path/to/ca-bundle.pem
   ```

After you complete the procedure, you can run CLI commands with these local credentials, your certificate, and your specified endpoint, as in the following example.

```
aws s3 ls --profile snowballEdge --endpoint https://192.0.2.0:8443
```

Batching Small Files

Each copy operation has some overhead because of encryption. To speed up the process of transferring small files, you can batch them together in a single archive. When you batch files together, they can be auto-extracted when they are imported into Amazon S3, if they were batched in one of the supported archive formats.

Typically, files that are 1 MB or smaller should be included in batches. There's no hard limit on the number of files you can have in a batch, though we recommend that you limit your batches to about 10,000 files. Having more than 100,000 files in a batch can affect how quickly those files import into Amazon S3 after you return the device. We recommend that the total size of each batch be no larger than 100 GB.

Batching files is a manual process, which you manage. After you batch your files, transfer them to a Snowball Edge device using the AWS CLI `cp` command with the `--metadata snowball-auto-extract=true` option. Specifying `auto-extract=true` automatically extracts the contents of the archived files when the data is imported into Amazon S3, so long as the size of the batched file is no larger than 100 GB.

Any batches larger than 100 GB, even if they were created with the `--metadata snowball-auto-extract=true` option, will be imported as a batch. They will not be automatically extracted, and you'll have to manually manage extracting those files once the batches are in Amazon S3.

To batch small files

1. Decide on what format you want to batch your small files in. The auto-extract feature supports the `TAR`, `ZIP`, and `tar.gz` formats.
2. Identify which small files you want to batch together, including their size and the total number of files that you want to batch together.
3. If you're using Linux, you can batch the files in the same command line used to transfer your files to the device, as in the following example.

```
tar -cf - /Logs/April | aws s3 cp - s3://mybucket/batch01.tar --metadata snowball-auto-extract=true --endpoint http://192.0.2.0:8089
```

   Note
   Alternatively, you can use the archive utility of your choice to batch the files into one or more large archives. However, this approach requires extra local storage to save the archives before you transfer them to the Snowball.
4. Repeat until you've archived all the small files that you want to transfer to Amazon S3 using a Snowball Edge.
5. Transfer the archived files to the Snowball. If you want the data to be auto-extracted, and you used one of the supported archive formats mentioned above in step 1, use the AWS CLI `cp` command with the `– –metadata snowball-auto-extract=true` option.

Supported AWS CLI Commands

Following, you can find information about how to specify the Amazon S3 Adapter for Snowball as the endpoint for applicable AWS Command Line Interface (AWS CLI) commands. You can also find the list of AWS CLI commands for Amazon S3 that are supported for transferring data to the AWS Snowball Edge appliance 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 AWS Command Line Interface User Guide.

Supported AWS CLI Commands for Amazon S3

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 AWS Snowball Edge appliance. The following are options for this command:
  - `--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. 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.
- `--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.
- `--storage-class` (string) The type of storage to use for the object. Valid choices are `STANDARD`, `REDUCED_REDUNDANCY`, and `STANDARD_IA`. The option defaults to `STANDARD`.
- `--page-size` (integer) The number of results to return in each response to a list operation. The default value is 1000 (the maximum allowed). Using a lower value might help if an operation times out.
- `--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 haven'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.

- ls Lists objects on the AWS Snowball Edge appliance. The following are options for this command:
  - `--human-readable` (boolean) File sizes are displayed in human-readable format.
  - `--summarize` (boolean) Summary information is displayed. This information is the number of objects and their total size.
  - `--recursive` (boolean) The command is performed on all files or objects under the specified directory or prefix.
  - `--page-size` (integer) The number of results to return in each response to a list operation. The default value is 1000 (the maximum allowed). Using a lower value might help if an operation times out.
- rm Deletes an object on the AWS Snowball Edge appliance. The following are options for this command:
  - `--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.
  - `--recursive` (boolean) The command is performed on all files or objects under the specified directory or prefix.
  - `--page-size` (integer) The number of results to return in each response to a list operation. The default value is 1000 (the maximum allowed). Using a lower value might help if an operation times out.
  - `--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.
- sync Syncs directories and prefixes. This command copies new and updated files from the source directory to the destination. This command only creates folders in the destination if they contain one or more files.

  Important
  Syncing from one directory to another directory on the same Snowball Edge isn't supported.

Syncing from one AWS Snowball device to another AWS Snowball device isn't supported. You can only use this option to sync the contents between your on-premises data storage and a Snowball Edge.

- `--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. 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.
- `--follow-symlinks` or `--no-follow-symlinks` (boolean) Symbolic links (symlinks) are followed only when uploading to S3 from the local file system. 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.
- `--storage-class` (string) The type of storage to use for the object. Valid choices are `STANDARD`, `REDUCED_REDUNDANCY`, and `STANDARD_IA`. The option defaults to `STANDARD`.
- `--only-show-errors` (boolean) Only errors and warnings are displayed. All other output is suppressed.
- `--no-progress` (boolean) File transfer progress is not displayed. This option is only applied when the `--quiet` and `--only-show-errors` options are not provided.
- `--page-size` (integer) The number of results to return in each response to a list operation. The default value is 1000 (the maximum allowed). Using a lower value might help if an operation times out.
- `--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 haven'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 to another directory on the same Snowball Edge isn't supported.
  Syncing from one AWS Snowball device to another AWS Snowball device isn't supported. You can only use this option to sync the contents between your on-premises data storage and a Snowball Edge.
- `--size-only` (boolean) With this option, the size of each key is the only criterion used to decide whether to sync from source to destination.
- `--exact-timestamps` (boolean) When syncing from S3 to local storage, same-sized items are ignored only when the timestamps match exactly. The default behavior is to ignore same-sized items unless the local version is newer than the S3 version.
- `--delete` (boolean) Files that exist in the destination but not in the source are deleted during sync.

You can work with files or folders with spaces in their names, such as `my photo.jpg` or `My Documents`. However, make sure that you handle the spaces properly in the AWS CLI commands. For more information, see Specifying Parameter Values for the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

Supported REST API Actions

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

Topics

Supported REST API Actions for Snowball Edge

HEAD Snowball Edge

Description

Currently, there's only one Snowball Edge REST API operation, which you can use to return status information for a specific device. This operation returns the status of a Snowball Edge. 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 following example, the IP address for the Snowball Edge 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 AWS Snowball Edge appliance counterpart. All responses coming back from an AWS Snowball Edge appliance 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
```

Amazon S3 REST API calls require SigV4 signing. If you're using the AWS CLI or an AWS SDK to make these API calls, the SigV4 signing is handled for you. Otherwise, you need to implement your own SigV4 signing solution. For more information, see Authenticating Requests (AWS Signature Version 4) in the *Amazon Simple Storage Service Developer Guide.*

- GET Bucket (List Objects) version 1 – Supported. However, the only supported delimiter is a forward slash. Only version 1 is supported. GET Bucket (List Objects) version 2 is not supported.

- GET Service
- HEAD Bucket
- HEAD Object
- GET Object – When an object is uploaded to an AWS Snowball Edge appliance 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 an AWS Snowball Edge appliance 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 AWS Snowball Edge appliance first deletes that object. It then copies it in parts to the AWS Snowball Edge appliance.
- 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.

Using the File Interface for the AWS Snowball Edge

Following, you can find information about using the file interface for the AWS Snowball Edge. Using this file interface, you can drag and drop files from your computer onto Amazon S3 buckets on the Snowball Edge.

Topics

Overview of the File Interface

The file interface exposes a Network File System (NFS) mount point for each bucket on your AWS Snowball Edge appliance. You can mount the file share from your NFS client using standard Linux, Microsoft Windows, or Mac commands. You can use standard file operations to access the file share.

After the file share has been mounted, a new **file interface** tab appears on the LCD screen on the front of the Snowball Edge. From this tab, you can get transfer status information, see your NFS point IP addresses, secure NFS client access to specific buckets, and open a support channel to AWS Support if a problem occurs with the file interface.

You can use the local LCD display on the AWS Snowball Edge appliance to disable or enable the file interface. By unlocking the AWS Snowball Edge appliance, you have all the permissions necessary to read and write data through the file interface.

Topics

Benefits of the File Interface

You might want to use the file interface to read and write data because of the following benefits:

- You can more easily read, write, and delete files by using the file interface.
- You can use the local LCD display on the AWS Snowball Edge appliance to monitor the file interface status.
- The file interface preserves user-defined metadata in objects. This metadata includes permissions, ownership, and time stamps and can be useful for tracking.
- Because files are written to the buckets on the appliance, adding files can trigger associated AWS Lambda powered by AWS Greengrass functions.

Prerequisites for Using the File Interface

Before you can use the file interface, the following steps must occur:

- You must create a job for your Snowball Edge.
- Your Snowball Edge device must arrive at your location.
- You must unlock your device by using the Snowball client.

If one or more of those steps haven't occurred, see the following topics:

- For more information about creating a job to use a Snowball Edge, see Getting Started with an AWS Snowball Edge Appliance (p. 20).
- For more information about unlocking a Snowball Edge, see Using the Amazon S3 Adapter (p. 46).

Considerations for Using the File Interface

While using the file interface, keep the following considerations in mind:

- The maximum size of a file that you can transfer to the file interface on a Snowball Edge is 150 GB. If you try to transfer a file larger than 150 GB, the file interface will write the first 150 GB of that file, and then return an error message indicating that the file is too large.
- We recommend that you use only one method of reading and writing data to each bucket on a Snowball Edge device. Using both the file interface and the Amazon S3 Adapter for Snowball on the same bucket might result in undefined behavior.
- File interface supports all NFS file operations, except truncate, rename, or changing ownership. Requests that use these unsupported file operations are rejected with error messages sent to your NFS client. Attempts to change a file's permissions after the file has been created on the Snowball Edge are ignored without error.
- If the Snowball Edge has a power failure or is rebooted, data in the file interface buffer persists. On reboot, this buffered data is uploaded to buckets on the device. When **Write status** on the **File**

interface tab shows 100 percent with a green progress bar, all data in the file interface buffer is uploaded to the buckets on the device.

- Don't write data to a Snowball Edge that is full, or write more data than the size of the remaining available storage. Either action causes errors that might corrupt your data. We recommend that you use the Snowball client's `snowballEdge status` command to determine the remaining amount of space on the Snowball Edge. Then compare that to the amount of data you want to copy over using the file interface before copying the data.
- When you've finished copying data to the Snowball Edge using the file interface, you must disable the file interface to avoid losing any data that might be in the buffer but not yet written to the Amazon S3 bucket. For more information, see Disabling the File Interface (p. 57).
- We recommend that you keep a local copy of all data that is written to the file interface until the Snowball Edge has been shipped back to AWS and the data has been ingested to Amazon S3.

Mounting a Bucket with the File Interface

The following contains guidance on mounting a file share on Snowball Edge to the NFS client on your computer using the file interface. It includes information about the supported NFS clients and procedures for enabling those clients on Linux, Mac, and Windows operating systems.

Topics

Supported NFS Clients for the File Interface

The file interface supports the following NFS clients:

Clients with NFSv4 support

- Amazon Linux
- macOS
- Red Hat Enterprise Linux (RHEL) 7
- Ubuntu 14.04

Clients with NFSv3 support

- Windows 10, Windows Server 2012, and Windows Server 2016
- Windows 7 and Windows Server 2008. For these clients, the maximum supported NFS I/O size is 32 KB. Because of this factor, you might experience degraded performance on these versions of Windows.

Getting the IP Address for the File Share of a Bucket on a Snowball Edge

You can mount the file shares with a simple command, if you have the IP address for the file share on a Snowball Edge. You can find the file share's IP address on the LCD display in the **CONNECTION** tab. You can't use the file interface if this IP address is blank. Ensure that the file interface gets an IP address.

Important
The IP address for the file interface is not the IP address that you used to unlock the Snowball Edge device. The IP address for the file interface can either be a static IP or one issued by your DHCP server.

To get the IP address for the file interface

1. Access the LCD display on the front of the AWS Snowball Edge appliance.
2. Tap **CONNECTION** at the top of the LCD display to open the network connection tab.
3. From the drop-down list in the center of the page, choose **file interface**.

   The IP address below this list updates to reflect the DHCP address that the AWS Snowball Edge appliance requested for the file interface. You can change it to a static IP address, or leave it as is.

Now that you have your IP address, you're ready to mount a bucket on the Snowball Edge using the appropriate mount command for your computer's operating system.

Mounting a File Share with the File Interface on Linux

When you mount file shares on your Linux server, we recommend that you first update your NFS client with the following command.

```
$ sudo nfs-utils
```

When the file interface is enabled, it exposes an NFS mount point for each local bucket on the appliance. The file interface supports NFS versions 3, 4.0, and 4.1. You can mount the file shares with a simple command with the IP address for the file interface. For more information, see Getting the IP Address for the File Share of a Bucket on a Snowball Edge (p. 54).

When you have the IP address, you can mount a bucket with the following command.

```
mount -t nfs -o nolock IP Address:/BucketName local/mount/directory
```

For example, suppose that the IP address for the file interface is 192.0.1.0 and your bucket name is **`test-bucket`**. You want to mount your bucket to the `mnt/test-bucket` directory on your local Linux server. In this case, your command looks like the following.

```
mount -o nolock 192.0.1.0:/test-bucket mnt/test-bucket
```

Mounting a File Share with the File Interface on a Mac

You can mount the file shares with a simple command with the IP address for the file interface. For more information, see Getting the IP Address for the File Share of a Bucket on a Snowball Edge (p. 54).
When you mount file shares on your Mac, you need to declare the version of the NFS protocol that you're using when you run the mount command. For example, if you're using the NFSv3.0 protocol, you use the `vers=3` option.

```
mount -t nfs -o vers=3,nolock IP Address:/BucketName local mount directory
```

For example, suppose the IP address for the file interface is 192.0.1.0, your bucket name is **`test-bucket`**, and you want to mount your bucket to the `private/mybucket` directory on your Mac. In this case, your command looks like the following.

```
sudo mount_nfs -o vers=3,nolock -v 192.0.1.0:/test-bucket private/mybucket
```

Mounting a File Share with the File Interface on Microsoft Windows

When you mount file shares on your Windows server, you need to turn on the Windows Client for NFS. You also need to assign the mount point a drive letter with the `mount` command.

> **Note**
> For a Windows 7 or Windows Server 2008 server, the maximum supported NFS I/O size is 32 KB. Because of this limit, you might experience degraded performance for the file interface on these versions of Windows.

To turn on Windows Client for NFS

1. In Windows, from **Start**, search for **Turn Windows features on or off** and choose the application of the same name that appears in the search results.
2. In the **Windows Features** dialog box that appears, scroll through the list of features until you find **Services for NFS**.
3. Expand **Services for NFS**, and select the **Client for NFS** check box.
4. Choose **OK** to close the dialog box with Client for NFS activated.

You can mount the file shares with a simple command with the IP address for the file interface. For more information, see Getting the IP Address for the File Share of a Bucket on a Snowball Edge (p. 54). You can now mount the file shares on the AWS Snowball Edge appliance to an unused drive letter on your Windows server as in the following example.

```
mount -o nolock 192.0.1.0:/test-bucket Z:
```

Monitoring the File Interface

When you use the file interface, it's important to keep an eye on its overall health and current status. You can perform these tasks by using the **file interface** tab on the LCD display on the front of the AWS Snowball Edge appliance.

Topics

Getting the Status of the File Interface

On the **file interface** tab, there are two health indicators, **Status** and **Write status**. The following list describes how to work with these indicators:

- **Status** indicates the operational status of the file interface as a whole. It has the following possible values:
  - **Enabled** – The file interface is up and running normally.
  - **Disabling** – The file interface is stopped, and nothing can be written to it.

  - **Disabled** – The file interface has stopped and the mount point is no longer available. In addition, all the data in the appliance's memory buffer has been encrypted and written to the local Amazon S3 buckets.
  - **Error** – An error has occurred. If you see this status, contact AWS Support.
- **Write status** shows you the progress of the current write operation executed on the AWS Snowball Edge appliance using a progress bar:
  - At 0–99 percent, a write operation is actively happening on the appliance and data is in the buffer. Don't disconnect the appliance before writing has completed.
  - At 100 percent, with a green progress bar, the last write operation completed successfully. There is no data in the buffer, and no new write operations have begun.

Securing Your NFS Connection

When a job for an AWS Snowball Edge appliance is created on the AWS console, all Amazon S3 buckets selected for the job are enabled by default as active file shares. When the appliance arrives at your site, and you set up, connect, and unlock it, anyone on your network who can see the IP address for the file interface can access the file shares for each bucket.

Therefore, we recommend that you secure the buckets by specifying which NFS clients are allowed to access your buckets. You can do this from the LCD screen on the front of the Snowball Edge with the following procedure.

To allow only certain NFS clients to access the file shares for your buckets on a Snowball Edge

1. On the LCD display, tap **File interface** to open its tab.
2. From **Allowed clients**, choose your bucket from the drop-down list.
3. Tap **Edit** to reveal the text boxes that you can enter your IP addresses into.
4. In the top box, use the onscreen keyboard to type in the IP address of a computer that you want to mount the file share for that bucket to.
5. If you have other computers connected to this same bucket, type in their IP addresses in the subsequent text boxes.

You have now secured the file share for one of your buckets on the Snowball Edge. You can repeat this process for all the file shares for the buckets on the Snowball Edge to secure access to the data in your device.

Once you specify an IP address for an allowed client, that file share return to unrestricted again by changing the IP address to `0.0.0.0`. If the IP address of the computer connected to it ever changes, you need to update the IP address for that allowed client.

Disabling the File Interface

When you're done using the file interface, you should disable the file interface after the **Write Status** on the AWS Snowball Edge appliance is set to **Complete**. Disabling the file interface helps you avoid data loss by ensuring that all files have been written to the appliance.

To stop the file interface

1. Access the LCD display on the front of the appliance.
2. On the LCD display, tap **File interface** to open its tab.
3. Press the **Disable** button on the **file interface** tab on the LCD display.

After you press the **Disable** button, the status changes to **Disabling** and any remaining buffered data is written to the appliance. When the write operation is complete, the status changes to **Disabled**.

You have now stopped the file interface. If you are otherwise finished with the AWS Snowball Edge appliance, it can now be safely powered off.

Opening a Support Channel for AWS Support

If the file interface ever enters into an error state, you might need to contact AWS Support. When you do so, AWS Support might ask you to open the support channel on your Snowball Edge.

The support channel is a special channel through which AWS Support can get information about the state of the file interface for troubleshooting purposes, if the Snowball Edge is connected to the internet. The following procedure outlines how to open a support channel when you're asked to do so by AWS Support.

To open a support channel for AWS Support

1. Access the LCD display on the front of the appliance.
2. On the LCD display, tap **File interface** to open its tab.
3. Scroll down to the bottom of the page.
4. When instructed by AWS Support, tap **Open support channel**.
5. After the channel opens, a port number appears that is labeled **on port: *xxxx*. Give this port number to AWS Support.** where *xxxx* is your port number.

At this point, you can work with AWS Support through this channel to troubleshoot the issue.

Using AWS Lambda with an AWS Snowball Edge

Following, you can find an overview of AWS Lambda powered by AWS Greengrass as used in an AWS Snowball Edge appliance. With this feature, you can run Lambda functions locally on a Snowball Edge. There is no charge for AWS Lambda powered by AWS Greengrass functions executed locally on an appliance. However, there might be charges for functions that create or use resources in the AWS Cloud.

> **Note**
> To use AWS Lambda powered by AWS Greengrass functions with Snowball Edge, you must create your jobs in an AWS Region supported by AWS Greengrass. For a list of valid regions, see AWS Greengrass in the *AWS General Reference*.

Before You Start

Before you create a Python-language Lambda function to run on your Snowball Edge, we recommend that you familiarize yourself with the following services, concepts, and related topics.

Prerequisites for AWS Greengrass

AWS Greengrass is software that extends AWS Cloud capabilities to local devices. AWS Greengrass makes it possible for local devices to collect and analyze data closer to the source of information, while also securely communicating with each other on local networks. More specifically, developers who use AWS Greengrass can author serverless code (Lambda functions) in the AWS Cloud. They can then conveniently deploy this code to devices for local execution of applications.

The following AWS Greengrass concepts are important to understand when using AWS Greengrass with a Snowball Edge:

- **AWS Greengrass Requirements** – For a full list of AWS Greengrass requirements, see Requirements in the *AWS Greengrass Developer Guide*. AWS Greengrass supports Python version 2.7, and each Lambda function has a minimum RAM requirement of 128 MB.
- **AWS Greengrass Core** – Each Snowball Edge has the AWS Greengrass core software. For more information on the AWS Greengrass core software, see Greengrass Core Software in the *AWS Greengrass Developer Guide*.
- **AWS Greengrass Group** – A Snowball Edge is part of an AWS Greengrass group as the group's core device. For more information on groups, see AWS Greengrass Groups in the *AWS Greengrass Developer Guide*.
- **MQTT** – AWS Greengrass uses the industry-standard, lightweight Message Queue Telemetry Transport (MQTT) protocol to communicate within a group. Within a Snowball Edge, there is an IoT device associated with the Amazon S3 Adapter for Snowball. When data is written using Amazon S3 PUT object operations on buckets specified at job creation, these operations trigger MQTT messages. These messages in turn trigger any associated Lambda functions. In addition, any MQTT-compatible device or software in your AWS Greengrass group can trigger the Lambda functions, if you define the related MQTT message to do so.
- **Associated service role** – Before you can use AWS Greengrass with a Snowball Edge as a core device, you must associate an AWS Greengrass service role with your account. This association allows AWS Greengrass to access your Lambda functions and AWS IoT resources. For more information, see Associating an AWS Greengrass Service Role with Your Account (p. 60).

Prerequisites for AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. The following Lambda concepts are important to understand when using Lambda with a Snowball Edge:

- **Lambda functions** – Your custom code, uploaded and published to Lambda and used on a Snowball Edge. For more information, see Lambda Functions in the *AWS Lambda Developer Guide*.
- **Lambda console** – The management console in which you'll upload, update, and publish your Python-language Lambda functions for use on a Snowball Edge. For a sample on how to use the Lambda console, see Step 2: Create a HelloWorld Lambda Function and Explore the Console in the *AWS Lambda Developer Guide*.
- **Python** – The high-level programming language used for your Lambda functions powered by AWS Greengrass on a Snowball Edge. AWS Greengrass supports Python version 2.7.

Related Topics

The following topics are related to running AWS Lambda powered by AWS Greengrass functions on a Snowball Edge:

- Limitations for Lambda Powered by AWS Greengrass (p. 96)
- Customer Managed Policy Examples (p. 84)

Next:

Getting Started with Lambda Powered by AWS Greengrass (p. 59)

Getting Started with Lambda Powered by AWS Greengrass

To get started with AWS Lambda powered by AWS Greengrass functions on a Snowball Edge, take the following steps:

1. Associating an AWS Greengrass Service Role with Your Account (p. 60)
2. Configuring AWS Greengrass with a Snowball Edge (p. 60)
3. Creating Your Job (p. 61)
4. Using Lambda Powered by AWS Greengrass Functions on a Snowball Edge (p. 61)

Associating an AWS Greengrass Service Role with Your Account

Before you can use AWS Greengrass with a Snowball Edge as a core device, you must associate an AWS Greengrass service role with your account. This association allows AWS Greengrass to access your Lambda functions and AWS IoT resources. If you try to create a job for a Snowball Edge before you associate the service role, the job creation request fails.

The following procedures show how to configure and store your AWS Greengrass settings in the cloud. This configuration means that you can push Lambda function changes to the Snowball Edge and changes to other devices in your AWS Greengrass group. The first procedure uses the AWS Identity and Access Management (IAM) console, and the second procedure uses the AWS Command Line Interface (AWS CLI).

To create an AWS Greengrass service role in the IAM console

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. For **Role Type**, choose **AWS Greengrass Role**.
3. Select **AWSGreengrassResourceAccessPolicy**, and then choose **Next Step**.
4. Type a name for your role, and then choose **Create Role**.

After creating the role, make a note of the role Amazon Resource Name (ARN), and use it in the next procedure.

To associate the AWS Greengrass service role with your account

1. Install the AWS CLI on your computer, if you haven't already. For more information, see Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.
2. Configure the AWS CLI on your computer, if you haven't already. For more information, see Configuring the AWS CLI in the *AWS Command Line Interface User Guide*.
3. Open a terminal on your computer and run the following AWS CLI command, with the AWS Greengrass role ARN you created in the first procedure.

```
aws greengrass associate-service-role-to-account --role-
arn arn:aws:iam::123EXAMPLE12:role/GreengrassRole
```

Now you can create a job for a Snowball Edge and use AWS Lambda powered by AWS Greengrass.

Configuring AWS Greengrass with a Snowball Edge

When you create a compute job for a Snowball Edge, the service automatically configures the following AWS Greengrass elements:

- **AWS Greengrass Core Software** – The AWS Greengrass distributable has been pre-installed on the Snowball Edge.
- **AWS Greengrass Group** – When you create a compute job, an AWS Greengrass group named *JobID*_group is created and provisioned. The core of this group is named *JobID*_core, and it has a

device in it named *JobID*_s3adapter. The Amazon S3 Adapter for Snowball is registered as an IoT device in your AWS Greengrass group. This registration is because the adapter sends MQTT messages for every Amazon S3 PUT object action for the buckets on the Snowball Edge.

If you have other configuration changes that you want to make for the AWS Greengrass group associated with a Snowball Edge, you can make them after you unlock the device and connect it to the internet.

Creating Your Job

Create a job in the AWS Snowball Management Console and associate the Amazon Resource Name (ARN) for at least one published Lambda function with a bucket. For a walkthrough on creating your first job, see Getting Started with AWS Snowball Edge: Your First Job (p. 20).

All the Lambda functions that you choose during job creation are triggered by MQTT messages sent by the IoT device associated with the Amazon S3 Adapter for Snowball. These MQTT messages are triggered whenever an Amazon S3 PUT object action is made against the bucket on the AWS Snowball Edge appliance.

Connecting to the Internet to Update AWS Greengrass Group Certificates

When the Snowball Edge arrives, unlock the device, and then connect it to the internet for at least one minute. Connecting the device to the internet allows the AWS Greengrass service in the cloud to send the AWS Greengrass certificates that are needed to operate the Snowball Edge. After that, you can disconnect the device from the internet. The associated AWS Greengrass group then functions in offline mode.

Important
When the IP address for any device in the local AWS Greengrass group changes, reconnect the Snowball Edge to the internet so it can get new certificates.

Using Lambda Powered by AWS Greengrass Functions on a Snowball Edge

Unless programmed otherwise, Lambda functions are triggered by MQTT messages sent by the IoT device associated with the Amazon S3 Adapter for Snowball. These MQTT messages are in turn triggered by Amazon S3 PUT object actions. Amazon S3 PUT object actions occur through the file interface (with a write operation), the AWS CLI (using the Amazon S3 Adapter for Snowball), or programmatically through one of the SDKs or a REST application of your own design.

You can use your AWS Lambda powered by AWS Greengrass functions to execute Python code against public endpoints among AWS services in the cloud. For this Python code execution to work, your AWS Snowball Edge appliance needs to be connected to the internet. For more information, see the AWS Lambda Developer Guide.

Updating Existing Functions

You can update existing Lambda functions in the console. If you do, and if your AWS Snowball Edge appliance is connected to the internet, a deployment agent notifies each Lambda function of the updated AWS Greengrass group configuration. For more information, see Create a Deployment in the *AWS Greengrass Developer Guide*.

Adding New Functions

After your Snowball Edge arrives and you unlock it and connect it to the internet, you can add or remove additional Lambda functions. These new Lambda functions don't have to be triggered by Amazon S3

PUT object actions. Instead, the event that you programmed to trigger the new functions performs function triggering, as with typical Lambda functions running in an AWS Greengrass group.

Testing Lambda Powered by AWS Greengrass Functions

While testing your Python-coded function in the Lambda console, you might encounter errors. If that happens, see Retries on Errors in the *AWS Lambda Developer Guide*.

Using an AWS Snowball Edge Cluster

A cluster is a logical grouping of AWS Snowball Edge devices, in groups of 5 to 10 devices. A cluster is created as a single job, which offers increased durability and storage capacity. In the following topic, you can find information about Snowball Edge clusters. This information includes conceptual, usage, and administrative information, in addition to walkthroughs for common Snowball Edge procedures.

> **Note**
> In January 2018, there was a feature update for clusters, making them leaderless. The cluster update is backward-compatible with older clusters. However, if you're looking for the original cluster documentation, see Additional Information for Snowball Edge (p. 101).

Topics

Clustering Overview

For the AWS Snowball service, a cluster is a collective of Snowball Edges, used as a single logical unit, for local storage and compute purposes.

A cluster offers two primary benefits over a standalone Snowball Edge for local storage and compute purposes:

- **Increased Durability** – The data stored in a cluster of Snowball Edge appliances enjoys increased data durability over a single device. In addition, the data on the cluster remains as safe and viable as it was previously, despite possible Snowball Edge outages in the cluster. Clusters can withstand the loss of two nodes before the data is in danger. You can also add or replace nodes.
- **Increased Storage** – The total available storage is 45 terabytes of data per node in the cluster. Thus, in a five-node cluster there are 225 terabytes of available storage space. In contrast, there are about 82 terabytes of available storage space in a standalone Snowball Edge. Clusters that have more than five nodes have even more storage space.

A cluster of Snowball Edge devices is made of leaderless nodes. Any node can write data to and read data from the entire cluster, and all nodes are capable of performing the behind-the-scenes management of the cluster.

Snowball Edge Cluster Quorums

A quorum represents the minimum number of Snowball Edge devices in a cluster that must be communicating with each other to maintain some level of operation. There are two levels of quorum for Snowball Edge clusters—a read/write quorum and a read quorum.

Let's say you upload your data to a cluster of Snowball Edge devices. With all devices healthy, you have a read/write quorum for your cluster.

If one of those nodes goes offline, you reduce the operational capacity of the cluster. However, you can still read and write to the cluster. In that sense, with the cluster operating all but one node, the cluster still has a read/write quorum.

If two nodes in your cluster are down, any additional or ongoing write operations fail. But any data that was successfully written to the cluster can be accessed and read. This is called a read quorum.

Finally, suppose that a third node loses power. Then the cluster is offline, and the data in the cluster is unavailable. You might be able fix this, or the data might be permanently lost, depending on the severity of the event. If it is a temporary external power event, and you can power the three Snowball Edges back on and unlock all the nodes in the cluster, then your data is available again.

Important
If a minimum quorum of healthy nodes doesn't exist, contact AWS Support.

You can determine the quorum state of your cluster by determining your node's lock state and network reachability. The `snowballEdge describe-cluster` command reports back the lock and network reachability state for every node in an unlocked cluster. Ensuring that the appliances in your cluster are healthy and connected is an administrative responsibility that you take on when you create the cluster job. For more information on the different client commands, see Commands for the Snowball Client (p. 37).

Considerations for Cluster Jobs for AWS Snowball Edge

Keep the following considerations in mind when planning to use a cluster of Snowball Edges:

- We recommend that you have a redundant power supply to reduce potential performance and stability issues for your cluster.
- As with standalone local storage and compute jobs, the data stored in a cluster can't be imported into Amazon S3 without ordering additional appliances as a part of separate import jobs. Then you can transfer the data from the cluster to those appliances and import the data when you return the appliances for the import jobs.
- To get data onto a cluster from Amazon S3, create a separate export job and copy the data from the appliances of the export job onto the cluster.
- You can create a cluster job from the console, the AWS CLI, or one of the AWS SDKs. For a guided walkthrough of creating a job, see Getting Started with an AWS Snowball Edge Appliance (p. 20).
- Cluster nodes have node IDs. A *node ID* is the same as the job ID for a device that you can get from the console, the AWS CLI, the AWS SDKs, and the Snowball client. You can use node IDs to remove old nodes from clusters. You can get a list of node IDs by using the `snowballEdge describe-device` command on an unlocked device or the `describe-cluster` on an unlocked cluster.
- The lifespan of a cluster is limited by the security certificate granted to the cluster devices when the cluster is provisioned. By default, Snowball Edge devices can be used for up to 120 days before they need to be returned. At the end of that time, the devices stop responding to read/write requests. If you need to keep one or more devices for longer than 120 days, contact AWS Support.
- When AWS receives a returned appliance that was part of a cluster, we perform a complete erasure of the appliance. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

Related Topics

Beyond the content presented in this topic, you can find other topics in this guide that are relevant to clusters:

- Getting Started with an AWS Snowball Edge Appliance (p. 20) – This section outlines how to get started creating your first job. The techniques in this section work for all job types, including cluster jobs.

- Commands for the Snowball Client (p. 37) – This section contains a list of commands for the Snowball client tool. These commands include the Snowball Edge administrative commands to unlock a cluster, get the status information for the nodes and the cluster as a whole, remove unavailable nodes, and add new nodes.
- Administrating a Cluster (p. 65) – This section contains information about the administrative tasks you perform with a cluster, like adding and removing nodes, and includes helpful procedures.

Administrating a Cluster

Following, you can find information about administrative tasks to operate a healthy cluster of Snowball Edge devices. The primary administrative tasks are covered in the following topics.

Topics

Most administrative tasks require that you use the Snowball client and its commands that perform the following actions:

- Unlocking AWS Snowball Edge Devices (p. 38)
- Getting Device Status (p. 42) of a cluster
- Removing a Node from a Cluster (p. 45)
- Adding a Node to a Cluster (p. 45)

Reading and Writing Data to a Cluster

After you've unlocked a cluster, you're ready to read and write data to it. You can use the Amazon S3 Adapter for Snowball to read and write data to a cluster. For more information, see Using the Amazon S3 Adapter (p. 46).

To write data to a cluster, you must have a read/write quorum with no more than one unavailable node. To read data from a cluster, you must have a read quorum of no more than two unavailable nodes. For more information on quorums, see Snowball Edge Cluster Quorums (p. 63).

Reconnecting an Unavailable Cluster Node

A node can become temporarily unavailable due to an issue (like power or network loss) without damaging the data on the node. When this happens, it affects the status of your cluster. A node's network reachability and lock status is reported in the Snowball client by using the `snowballEdge describe-cluster` command.

We recommend that you physically position your cluster so you have access to the front, back, and top of all nodes. This way, you can access the power and network cables on the back, the shipping label on the top to get your node ID, and the LCD screen on the front of the device for the IP address and other administrative information.

When you detect that a node is unavailable, we recommend that you try one of the following procedures, depending on the scenario that caused the unavailability.

To reconnect an unavailable node

1. Ensure that the node has power.
2. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
3. Wait for the node to finish powering up, if it needed to be powered up.
4. Run the `snowballEdge unlock-cluster` command, or the `snowballEdge associate-device` command. For an example, see Unlocking AWS Snowball Edge Devices (p. 38).

To reconnect an unavailable node that lost network but didn't lose power

1. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
2. Run the `snowballEdge describe-device` command to see when the previously unavailable node is added back to the cluster. For an example, see Getting Device Status (p. 42).

When you have performed the preceding procedures, your nodes should be working normally. You should also have a read/write quorum. If that's not the case, then one or more of your nodes might have a more serious issue and might need to be removed from the cluster.

Removing an Unhealthy Node from a Cluster

Rarely, a node in your cluster might become unhealthy. If the node is unavailable, we recommend going through the procedures listed in Reconnecting an Unavailable Cluster Node (p. 65) first.

If doing so doesn't resolve the issue, then the node might be unhealthy. An unhealthy node can occur if the node has taken damage from an external source, if there was an unusual electrical event, or if some other unlikely event occurs. If this happens, you need to remove the node from the cluster before you can add a new node as a replacement.

When you detect that a node is unhealthy and needs to be removed, we recommend that you do so with the following procedure.

To remove an unhealthy node

1. Ensure that the node is unhealthy and not just unavailable. For more information, see Reconnecting an Unavailable Cluster Node (p. 65).
2. Disconnect the unhealthy node from the network and power it off.
3. Run the `snowballEdge dissassociate-device` Snowball client command. For more information, see Removing a Node from a Cluster (p. 45).
4. Order a replacement node using the console, the AWS CLI, or one of the AWS SDKs.
5. Return the unhealthy node to AWS. When we have the node, we perform a complete erasure of the device. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

After you successfully remove a node, your data is still available on the cluster if you still have a read quorum. To have read quorum, a cluster must have no more than two unavailable nodes. Therefore, we recommend that you order replacement nodes as soon as you remove an unavailable node from the cluster.

Adding or Replacing a Node in a Cluster

You can add a new node after you have removed an unhealthy node from a cluster. You can also add a new node to increase local storage.

To add a new node, you first need to order a replacement. You can order a replacement node from the console, the AWS CLI, or one of the AWS SDKs. If you're ordering a replacement node from the console, you can order replacements for any job that hasn't been canceled or completed.

To order a replacement node from the console

1. Sign in to the AWS Snowball Management Console.
2. Find and choose a job for a node that belongs to the cluster that you created from the Job dashboard.
3. For **Actions**, choose **Replace node**.

   Doing this opens the final step of the job creation wizard, with all settings identical to how the cluster was originally created.
4. Choose **Create job**.

Your replacement Snowball Edge is now on its way to you. When it arrives, use the following procedure to add it to your cluster.

To add a replacement node

1. Position the new node for the cluster such that you have access to the front, back, and top of all nodes.
2. Ensure that the node has power.
3. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
4. Wait for the node to finish powering up, if it needed to be powered on.
5. Run the `snowballEdge associate-device` command. For an example, see Adding a Node to a Cluster (p. 45).

Shipping Considerations for AWS Snowball

Following, you can find information about how shipping is handled for an AWS Snowball Edge appliance, and a list that shows each AWS Region that is supported. The shipping rate you choose for a job apply to both sending and receiving the AWS Snowball Edge appliance or AWS Snowball Edge appliances used for that job. For information on shipping charges, see AWS Snowball Edge 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 AWS Snowball Edge appliance 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, which depends on factors including job type (exports take longer than imports, typically) and job size (cluster jobs take longer than individual jobs, typically). Also, carriers generally only pick up outgoing AWS Snowball Edge appliances once a day. Thus, processing before shipping can take a day or more.

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

Preparing an AWS Snowball Edge for Shipping

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

To prepare an AWS Snowball Edge appliance for shipping

1. Make sure that you've finished transferring all the data for this job to or from the AWS Snowball Edge appliance.
2. Press the power button above the LCD display. It takes about 20 seconds for the appliance to power off.

   > **Note**
   > If you've powered off and unplugged the AWS Snowball Edge appliance, and your shipping label doesn't appear after about a minute on the E Ink display on top of the appliance, contact AWS Support.
3. Disconnect and stow the power cable the AWS Snowball Edge appliance was sent with in the cable nook on top of the appliance.
4. Close the three doors on the back, the top, and the front of the AWS Snowball Edge appliance, pressing in on each one at a time until you hear and feel them click.

You don't need to pack the AWS Snowball Edge appliance in a container, because it is its own physically rugged shipping container. The E Ink display on the top of the AWS Snowball Edge appliance changes to your return shipping label when the AWS Snowball Edge appliance 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 or Amazon Glacier data is housed in. A few shipping restrictions apply, as follows:

- For data transfers in US regions, we don't ship AWS Snowball Edge appliances outside of the United States.
- We don't ship AWS Snowball Edge appliances 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 AWS Snowball Edge appliances within Australia.
- For data transfers in the EU regions, we only ship AWS Snowball Edge appliances to the 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.

Note
AWS doesn't ship AWS Snowball Edge appliances to post office boxes.

Shipping an AWS Snowball Edge

The prepaid shipping label contains the correct address to return the AWS Snowball Edge appliance. For information on how to return your AWS Snowball Edge appliance, see Shipping Carriers (p. 69). The AWS Snowball Edge appliance 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 AWS Snowball Edge appliance. Always use the shipping label that is displayed on the AWS Snowball Edge appliance E Ink display.

Shipping Carriers

When you create a job, you provide the address that you want the AWS Snowball Edge appliance shipped to. The carrier that supports your region handles the shipping of AWS Snowball Edge appliances from AWS to you, and from you back to AWS. Whenever an AWS Snowball Edge appliance 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 AWS Snowball Edge appliances by region:

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

AWS Snowball Pickups in the EU, US, and Canada

In the EU and US and Canadian regions, keep the following information in mind for UPS to pick up an AWS Snowball Edge appliance:

- You arrange for UPS to pick up the AWS Snowball Edge appliance by scheduling a pickup with UPS directly, or take the AWS Snowball Edge appliance 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 AWS Snowball Edge appliance.
- The AWS Snowball Edge appliance 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 AWS Snowball Edge appliance. Always use the shipping label that is displayed on the AWS Snowball Edge appliance's E Ink display.

AWS Snowball Pickups in Brazil

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

- When you're ready to return a Snowball Edge, call 0800-770-9035 to schedule a pickup with UPS.
- Snowball Edge 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 Edge 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 the declaration: one for your records, the other three for transport.

AWS Snowball Pickups in Australia

In Australia, if you're shipping an AWS Snowball Edge appliance back to AWS, send an email to snowball-pickup@amazon.com with *Snowball Edge 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 AWS Snowball Edge appliance 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 AWS Snowball Edge appliance to you.
- **Earliest Pickup Time** (your local time) – The earliest time of day that you want the AWS Snowball Edge appliance picked up.
- **Latest Pickup Time** (your local time) – The latest time of day that you want the AWS Snowball Edge appliance 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 AWS Snowball Edge appliance.

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 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 Edge 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 either of the displays.
- The Snowball Edge 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 an AWS Snowball Edge appliance. Shipping speeds are as follows:

- **Australia** – When shipping within Australia, you have access to express shipping. Typically, AWS Snowball Edge appliances 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 that within the EU, you have access to express shipping. Typically, AWS Snowball Edge appliances 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.
- **Japan** – When shipping within Japan, you have access to standard shipping speed.
- **United States of America (US) and Canada** – When shipping in the US or in Canada, you have access to one-day shipping and two-day shipping.

Security for AWS Snowball Edge

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

Topics

Encryption for AWS Snowball Edge

When you're using a Snowball Edge to import data into S3, all data transferred to a device is protected by SSL encryption over the network. To protect data at rest, AWS Snowball uses server side-encryption (SSE).

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, AWS 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 Edge

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 set 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 the **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.

AWS Key Management Service in AWS 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 AWS Snowball, AWS KMS protects the encryption keys used to protect data on each AWS Snowball Edge appliance. When you create your job, you also choose an existing KMS key. Specifying the ARN for

an AWS KMS key tells AWS Snowball which AWS KMS master key to use to encrypt the unique keys on the AWS Snowball Edge appliance.

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. 72).

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 AWS KMS CMKs.
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 AWS 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 on the appliance. Unencrypted data is never stored on the appliance.

When using the adapter, keep the following in mind:

- To get the local Amazon S3 credentials to sign your requests to the AWS Snowball Edge appliance, run the `snowballEdge list-access-keys` and `snowballEdge get-secret-access-keys` Snowball client commands. For more information, see Using the Snowball Client (p. 36). These local Amazon S3 credentials include a pair of keys: an access key and a secret key. These keys are only valid

for the appliances associated with your job. They can't be used in the AWS Cloud as they have no IAM counterpart.

- 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.

Other Security Considerations for AWS Snowball

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

We recommend the following security approaches:

- When the appliance first arrives, inspect it for damage or obvious tampering. If you notice anything that looks suspicious about the appliance, don't connect it to your internal network. Instead, contact AWS Support, and a new appliance 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 appliance sent for that job.
- Don't leave the appliance sitting on a loading dock. Left on a loading dock, it can be exposed to the elements. Although each AWS Snowball appliance is rugged, weather can damage the sturdiest of hardware. Report stolen, missing, or broken appliances as soon as possible. The sooner such an issue is reported, the sooner another one can be sent to complete your job.

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

We perform the following security steps:

- When transferring data with the Amazon S3 Adapter for Snowball, object metadata is not persisted. 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 transferring data with the file interface, object metadata is persisted.
- When an appliance arrives at AWS, we inspect it 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.

Authentication and Access Control for AWS Snowball Edge

As with all AWS services, access to AWS Snowball requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such an Amazon S3 bucket or an AWS Lambda function. AWS Snowball differs in two ways:

1. Jobs in AWS Snowball do not have Amazon Resource Names (ARNs).
2. Physical and network access control for an appliance on-premises is your responsibility.

The following sections provide details on how you can use AWS Identity and Access Management (IAM) and AWS Snowball to help secure your resources by controlling who can access them in the AWS Cloud, and also local access control recommendations.

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.
- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a job in AWS Snowball). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Snowball supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the *AWS General Reference*.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing user identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the *IAM User Guide*.

- **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account's resources. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the *IAM User Guide*.

Access Control in the AWS Cloud

You can have valid credentials to authenticate your requests in AWS. However, unless you have permissions you cannot create or access AWS resources. For example, you must have permissions to create a job for AWS Snowball.

The following sections describe how to manage cloud-based permissions for AWS Snowball. We recommend that you read the overview first.

- Overview of Managing Access Permissions to Your Resources in the AWS Cloud (p. 77)
- Using Identity-Based Policies (IAM Policies) for AWS Snowball (p. 80)

Overview of Managing Access Permissions to Your Resources in the AWS Cloud

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

- Resources and Operations (p. 78)
- Understanding Resource Ownership (p. 78)
- Managing Access to Resources in the AWS Cloud (p. 78)

Resources and Operations

In AWS Snowball the primary resource is a *job*. AWS Snowball also has appliances like the Snowball and the AWS Snowball Edge appliance, however, you can only use those appliances in the context of an existing job. Amazon S3 buckets and Lambda functions are resources of Amazon S3 and Lambda respectively.

As mentioned previously, jobs don't have Amazon Resource Names (ARNs) associated with them. However, other services' resources, like Amazon S3 buckets These do have unique (ARNs) associated with them as shown in the following table.

AWS Snowball provides a set of operations to create and manage jobs. For a list of available operations, see the AWS Snowball API Reference.

Understanding Resource Ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the root account, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a file system, your AWS account is the owner of the resource (in AWS Snowball, the resource is the job).
- If you create an IAM user in your AWS account and grant permissions to create a job to that user, the user can create a job. However, your AWS account, to which the user belongs, owns the job resource.
- If you create an IAM role in your AWS account with permissions to create a job, anyone who can assume the role can create a job. Your AWS account, to which the role belongs, owns the job resource.

Managing Access to Resources in the AWS Cloud

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of AWS Snowball. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices) and policies attached to a resource are referred to as *resource-based* policies. AWS Snowball supports only identity-based policies (IAM policies).

Topics

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create a job, you can attach a permissions policy to a user or group that the user belongs to.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

  For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

The following is an example policy that allows a user to perform the `CreateJob` action for your AWS account.

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

For more information about using identity-based policies with AWS Snowball, see Using Identity-Based Policies (IAM Policies) for AWS Snowball (p. 80). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Snowball doesn't support resource-based policies.

Specifying Policy Elements: Actions, Effects, and Principals

For each job (see Resources and Operations (p. 78)), the service defines a set of API operations (see AWS Snowball API Reference) to create and manage said job. To grant permissions for these API

operations, AWS Snowball defines a set of actions that you can specify in a policy. For example, for a job, the following actions are defined: `CreateJob`, `CancelJob`, and `DescribeJob`. Note that, performing an API operation can require permissions for more than one action.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see Resources and Operations (p. 78).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, depending on the specified `Effect`, `snowball:*` either allows or denies the user permissions to perform all operations.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Snowball doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

For a table showing all of the AWS Snowball API actions, see AWS Snowball API Permissions: Actions, Resources, and Conditions Reference (p. 87).

Specifying Conditions in a Policy

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS Snowball. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the *IAM User Guide*.

Using Identity-Based Policies (IAM Policies) for AWS Snowball

This topic provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Snowball resources in the AWS Cloud.

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Snowball resources. For more information, see Overview of Managing Access Permissions to Your Resources in the AWS Cloud (p. 77).

The sections in this topic cover the following:

- Permissions Required to Use the AWS Snowball Console (p. 81)

- AWS Managed (Predefined) Policies for AWS Snowball (p. 84)
- Customer Managed Policy Examples (p. 84)

The following shows an example of a permissions policy.

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

The policy has two statements:

- The first statement grants permissions for three Amazon S3 actions (`s3:GetBucketLocation`, `s3:GetObject`, and `s3:ListBucket`) on all Amazon S3 buckets using the *Amazon Resource Name (ARN)* of `arn:aws:s3:::*`. The ARN specifies a wildcard character (*) so the user can choose any or all Amazon S3 buckets to export data from.
- The second statement grants permissions for all AWS Snowball actions. Because these actions don't support resource-level permissions, the policy specifies the wildcard character (*) and the `Resource` value also specifies a wild card character.

The policy doesn't specify the `Principal` element because in an identity-based policy you don't specify the principal who gets the permission. When you attach policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table showing all of the AWS Snowball job management API actions and the resources that they apply to, see AWS Snowball API Permissions: Actions, Resources, and Conditions Reference (p. 87).

Permissions Required to Use the AWS Snowball Console

The permissions reference table lists the AWS Snowball job management API operations and shows the required permissions for each operation. For more information about job management API operations, see AWS Snowball API Permissions: Actions, Resources, and Conditions Reference (p. 87).

To use the AWS Snowball Management Console, you need to grant permissions for additional actions as shown in the following permissions policy:

```
{
```

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

                ",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketLocation",
            "s3:GetBucketPolicy",
            "s3:ListBucket",
            "s3:ListBucketMultipartUploads",
            "s3:ListAllMyBuckets"
        ],
         "Resource": "arn:aws:s3:::*"
    },
   {
        "Effect": "Allow",
        "Action": [
            "s3:CreateBucket",
            "s3:PutObject",
            "s3:AbortMultipartUpload",
            "s3:ListMultipartUploadParts",
            "s3:PutObjectAcl",
        ],
         "Resource": "arn:aws:s3:::*"
    },

    {
        "Effect": "Allow",
        "Action": [
            "lambda:GetFunction",
            "lambda:GetFunctionConfiguration"
        ],
       "Resource": "arn:aws:lambda:::function:*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "lambda:ListFunctions"
        ],
       "Resource": "arn:aws:lambda:::*"
    },
    {
       "Effect": "Allow",
       "Action": [
            "kms:CreateGrant",
            "kms:GenerateDataKey",
            "kms:Decrypt",
            "kms:Encrypt",
            "kms:RetireGrant",
            "kms:ListKeys",
            "kms:DescribeKey",
            "kms:ListAliases"
       ],
       "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:ListTopics",
                "sns:GetTopicAttributes",
                "sns:SetTopicAttributes",
                "sns:ListSubscriptionsByTopic",
                "sns:Subscribe"
            ],
            "Resource": [
                "*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "greengrass:getServiceRoleForAccount"
            ],
            "Resource": [
                "*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "snowball:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

The AWS Snowball console needs these additional permissions for the following reasons:

- `lambda:` – These allow the user to select Lambda functions for local compute purposes. For more information, see Using AWS Lambda with an AWS Snowball Edge (p. 58).
- `kms:` – These allow the user to create or choose the KMS key that will encrypt your data. For more information, see AWS Key Management Service in AWS Snowball (p. 73).
- `iam:` – These allow the user to create or choose an IAM role ARN that AWS Snowball will assume to access the AWS resources associated with job creation and processing.
- `sns:` – These allow the user to create or choose the Amazon SNS notifications for the jobs they create. For more information, see Notifications for the AWS Snowball Edge (p. 90).

AWS Managed (Predefined) Policies for AWS Snowball

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the *IAM User Guide*. There are no AWS managed policies for AWS Snowball.

You can create your own custom IAM policies to allow permissions for AWS Snowball job management API actions. You can attach these custom policies to the IAM users or groups that require those permissions.

Customer Managed Policy Examples

In this section, you can find example user policies that grant permissions for various AWS Snowball job management actions. These policies work when you are using AWS SDKs or the AWS CLI. When you are using the console, you need to grant additional permissions specific to the console, which is discussed in Permissions Required to Use the AWS Snowball Console (p. 81).

> **Note**
> All examples use the us-west-2 region and contain fictitious account IDs.

Examples

Example 1: Role Policy That Allows a User to Create a Job with the API

The following permissions policy is a necessary component of any policy that is used to grant job or cluster creation permission using the job management API. The user also needs some or all of the permissions specified in Permissions Required to Use the AWS Snowball Console (p. 81), depending on the type of job created.

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

Example 2: Role Policy for Creating Import Jobs

You use the following role trust policy for creating import jobs for Snowball Edge that use AWS Lambda powered by AWS Greengrass functions.

```
                {
"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": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:AttachPrincipalPolicy",
            "iot:AttachThingPrincipal",
            "iot:CreateKeysAndCertificate",
            "iot:CreatePolicy",
            "iot:CreateThing",
            "iot:DescribeEndpoint",
            "iot:GetPolicy"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "lambda:GetFunction"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
```

```
                "greengrass:CreateCoreDefinition",
                "greengrass:CreateDeployment",
                "greengrass:CreateDeviceDefinition",
                "greengrass:CreateFunctionDefinition",
                "greengrass:CreateGroup",
                "greengrass:CreateGroupVersion",
                "greengrass:CreateLoggerDefinition",
                "greengrass:CreateSubscriptionDefinition",
                "greengrass:GetDeploymentStatus",
                "greengrass:UpdateGroupCertificateConfiguration",
                "greengrass:CreateGroupCertificateAuthority",
                "greengrass:GetGroupCertificateAuthority",
                "greengrass:ListGroupCertificateAuthorities",
                "greengrass:ListDeployments",
                "greengrass:GetGroup",
                "greengrass:GetGroupVersion",
                "greengrass:GetCoreDefinitionVersion"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Example 3: Role Policy for Creating Export Jobs

You use the following role trust policy for creating export jobs for Snowball Edge that use AWS Lambda powered by AWS Greengrass functions.

```
                    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
           "Effect": "Allow",
           "Action": [
                "snowball:*"
           ],
           "Resource": [
                "*"
           ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:AttachPrincipalPolicy",
                "iot:AttachThingPrincipal",
                "iot:CreateKeysAndCertificate",
                "iot:CreatePolicy",
                "iot:CreateThing",
                "iot:DescribeEndpoint",
                "iot:GetPolicy"
```

```
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetFunction"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "greengrass:CreateCoreDefinition",
                "greengrass:CreateDeployment",
                "greengrass:CreateDeviceDefinition",
                "greengrass:CreateFunctionDefinition",
                "greengrass:CreateGroup",
                "greengrass:CreateGroupVersion",
                "greengrass:CreateLoggerDefinition",
                "greengrass:CreateSubscriptionDefinition",
                "greengrass:GetDeploymentStatus",
                "greengrass:UpdateGroupCertificateConfiguration",
                "greengrass:CreateGroupCertificateAuthority",
                "greengrass:GetGroupCertificateAuthority",
                "greengrass:ListGroupCertificateAuthorities",
                "greengrass:ListDeployments",
                "greengrass:GetGroup",
                "greengrass:GetGroupVersion",
                "greengrass:GetCoreDefinitionVersion"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

AWS Snowball API Permissions: Actions, Resources, and Conditions Reference

When you are setting up Access Control in the AWS Cloud (p. 77) and writing a permissions policy that you can attach to an IAM identity (identity-based policies), you can use the following list as a reference. The list includes each AWS Snowball job management API operation, the corresponding actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

You can use AWS-wide condition keys in your AWS Snowball policies to express conditions. For a complete list of AWS-wide keys, see Available Keys in the *IAM User Guide*.

Note
To specify an action, use the `snowball:` prefix followed by the API operation name (for example, `snowball:CreateJob`).

Data Validation with Snowball Edge Jobs

Following, you'll find information on how Snowball Edge 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 Amazon S3 Adapter for Snowball to the Snowball Edge, 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). For the Snowball Edge, these checksums are visible when you run the AWS CLI `s3 ls` command against a bucket on the device. The checksums are used to validate the integrity of your data throughout the transfers 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 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 Edge that is already at full data storage capacity.
- Attempting to copy files to a Snowball Edge 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 Edge During Transfer (p. 89) – Outlines how to check for failed files while you still have the Snowball Edge on-premises.
- Manual Data Validation for Snowball Edge 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 Edge 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.

When you run the Amazon S3 Adapter for Snowball to copy data with the AWS CLI, logs are generated. One option for manual validation is to check these logs. These 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>`

Manual Data Validation for Snowball Edge 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 Get Your Job Completion Report and Logs in the Console (p. 27).

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 comma-separated value (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 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.

Notifications for the AWS Snowball Edge

Like the standard Snowball, the AWS Snowball Edge appliance 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. 11).

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 Edge Specifications

The following table outlines hardware specifications for the AWS Snowball Edge appliance.

Item	Specification
Storage capacity	100 TB Snowballs have about 82 TB of usable space.
Data and network connections	Network connections: • 10 GBase-T – RJ45 • 25 GB – SFP+ • 40 GB – QSFP+ To use the AWS Snowball Edge appliance, you need your own network cables. For RJ45 cables, there are no specific recommendations. SFP+ and QSFP+ cables and modules from Mellanox and Finisar have been verified to be compatible with the appliance.
Cables	Each AWS Snowball Edge appliance ships country-specific power cables. No other cables or optics are not provided.
Thermal requirements	AWS Snowball Edge appliances are designed for office operations, and are ideal for data center operations.
Decibel output	On average, an AWS Snowball Edge appliance produces 68 decibels of sound, typically quieter than a vacuum cleaner or living-room music.
Weight	49.5 pounds (22.6 Kg)
Height	15.25 inches (386 mm)
Width	10.375 inches (259 mm)
Length	26.00 inches (671 mm)
Power	In the US regions: NEMA 5–15p 100–220 volts. In all regions, a power cable is included.
Power consumption	400 watts
Voltage	100 – 240V AC
Frequency	47/63 Hz
Power conversion efficiency	89 – 92% at 25C, 230Vac
Temperature range	0 – 40°C (operational)
Non-Operational Vibration	ASTM D4169 Truck Level I 0.73 GRMS
Non-Operational Shock	Drop Test (12” all sides + 24” 1 side)
Non-operational Altitude	0 – 12,000 meters

Item	Specification
Operational Altitude	0 to 3,000m (0 to 10,000')

Supported Network Hardware

After you open the back panel of the AWS Snowball Edge appliance, you'll see the network ports shown in the following photograph.

These ports support the following network hardware.

SFP

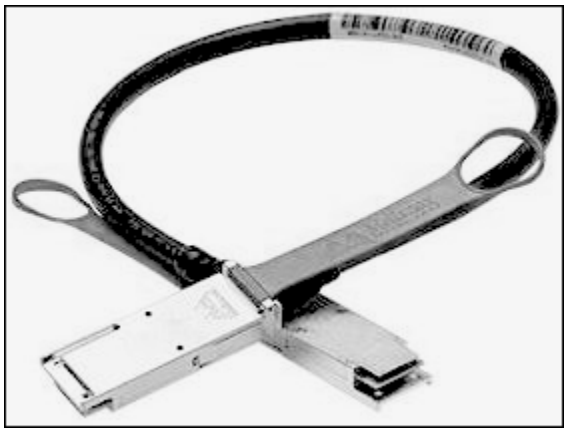
This port provides a 10G/25G SFP28 interface compatible with SFP28 and SFP+ transceiver modules and direct-attach copper (DAC) cables. You need to provide your own transceivers or DAC cables.

- For 10G operation, you can use any SFP+ option. Examples include:
  - 10Gbase-LR (single mode fiber) transceiver
  - 10Gbase-SR (multi-mode fiber) transceiver
  - SFP+ DAC cable
- For 25G operation, you can use any SFP28 option. Examples include:
  - 25Gbase-LR (single mode fiber) transceiver
  - 25Gbase-SR (multi-mode fiber) transceiver
  - SFP28 DAC cable

QSFP

This port provides a 40G QSFP+ interface compatible with QSFP+ transceiver modules and DAC cables. You need to provide your own transceivers or DAC cables. Examples include the following:

- 40Gbase-LR4 (single mode fiber) transceiver
- 40Gbase-SR4 (multi-mode fiber) transceiver
- QSFP+ DAC

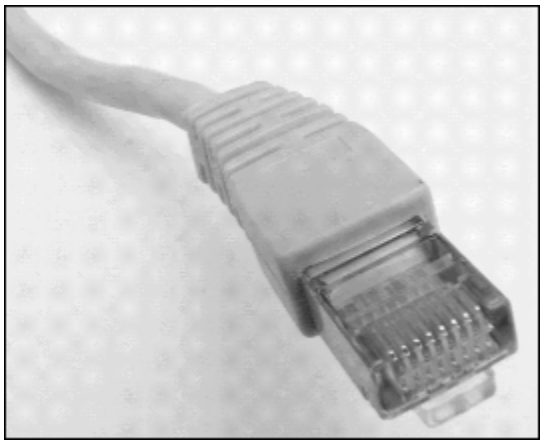


RJ45

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 Edge device, as it dramatically increases the time it takes to transfer data.

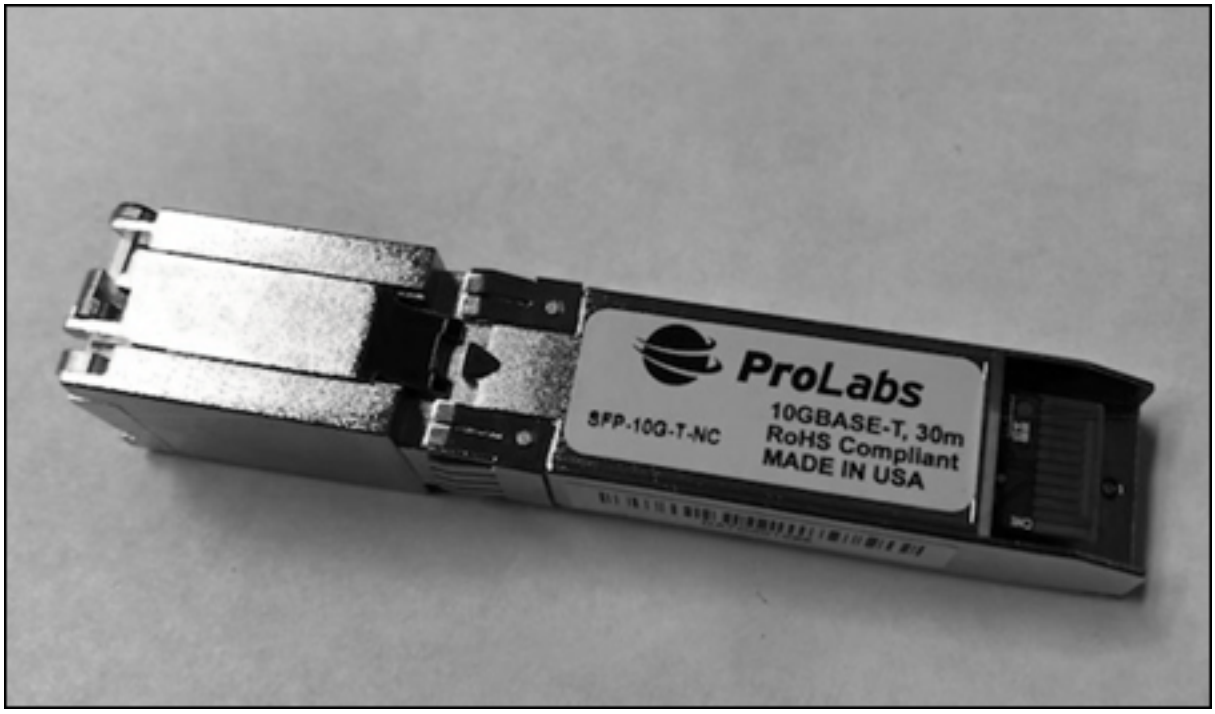
10G operation is indicated by a blinking green light. It requires a Cat6A UTP cable with a maximum operating distance of 180 feet (55 meters).



Network Converter for a Second Connection to the Snowball Edge

For applications requiring a second 10G BASE-T connection to the Snowball Edge, you can install an SFP + to 10G BASE-T media converter module in the SFP+ port.

AWS has qualified the ProLabs model SFP-10G-T-NC media converter for this purpose. You can purchase this converter from Anixter, Graybar, and other distributors.



The following procedure outlines how to set up this media converter for use with a Snowball Edge.

To set up the media converter for use with a Snowball Edge

1. Insert the module into the SFP+ port.
2. Connect a Cat6A cable to the Snowball Edge and to your network.
3. Restart the power for the Snowball Edge.

Important
Remove the media converter before returning your Snowball Edge to AWS.

AWS Snowball Edge Limits

Following, you can find information about limitations on using the AWS Snowball Edge appliance.

Important
When you transfer data into Amazon Simple Storage Service using a Snowball Edge, 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 Edge, which has 100 TB of storage space. If you are looking for documentation for the Snowball, see the AWS Snowball User 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 AWS Snowball Edge Jobs

The following limitations exist for creating AWS Snowball Edge appliance jobs:

- For security purposes, jobs must be completed within 120 days of the AWS Snowball Edge appliance being prepared. If you need to keep one or more devices for longer than 120 days, contact AWS Support.
- Currently, AWS Snowball Edge appliance doesn't support server-side encryption with AWS Key Management Service–managed keys (SSE-KMS) or server-side encryption with customer-provided keys

(SSE-C). AWS Snowball Edge appliance 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.*

- If you're using AWS Snowball Edge appliance to import data, and you need to transfer more data than will fit on a single AWS Snowball Edge appliance, create additional jobs. Each export job can use multiple AWS Snowball Edge appliances.
- The default service limit for the number of AWS Snowball Edge appliances you can have at one time is 1. If you want to increase your service limit or create a cluster job, contact AWS Support.
- Metadata for objects transferred to an appliance is not persisted, unless it's transferred with the file interface. 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 an AWS Snowball Edge Appliance

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

- Files must be in a static state while being written. 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 an AWS Snowball Edge appliance, not for downloading data from a AWS Snowball Edge appliance.

Limitations for Lambda Powered by AWS Greengrass

If you allocate the minimum recommendation of 128 MB of memory for each of your functions, you can have up to seven Lambda functions in a single job. This limitation occurs because the physical nature of the Snowball Edge limits the amount of memory available for running Lambda functions on the device.

Limitations on Shipping an AWS Snowball Edge

The following limitations exist for shipping a AWS Snowball Edge appliance:

- AWS will not ship a AWS Snowball Edge appliance to a post office box.
- AWS will not ship a AWS Snowball Edge appliance between non-US regions—for example, from EU (Ireland) to EU (Frankfurt), or to Asia Pacific (Sydney).
- Moving a AWS Snowball Edge appliance 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. 68).

Limitations on Processing Your Returned AWS Snowball Edge for Import

To import your data into AWS, the appliance must meet the following requirements:

- The AWS Snowball Edge appliance must not be compromised. Except for opening the three doors on the front, back, and top, or to add and replace the optional air filter, don't open the AWS Snowball Edge appliance for any reason.
- The appliance must not be physically damaged. You can prevent damage by closing the three doors on the AWS Snowball Edge appliance until the latches make an audible clicking sound.
- The AWS Snowball Edge appliance'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 AWS Snowball Edge appliance.

Note
All AWS Snowball Edge appliances returned that do not meet these requirements are erased without work performed on them.

Troubleshooting for an AWS Snowball Edge

Following, you can find information to help you troubleshoot problems with an AWS Snowball Edge appliance. If you have 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*. In addition, be aware of the following:

- Objects in Amazon S3 have a maximum file size of 5 TB.
- Objects transferred onto AWS Snowball Edge appliances have a maximum key length of 933 bytes. Key names that include characters that take up more than 1 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$7BXys3KHYePfuIOMNjY83dVxugPYlxVg/evpcQEJLT/rSwZc$MlVVf/$hwefVISRqwepB$/BiiD/PPF$twRAjrD/fIMp/0NY`	135 characters

- For security purposes, import and export jobs using an AWS Snowball Edge appliance must be completed within 120 days of the being prepared. If you need to keep one or more devices for longer than 120 days, contact AWS Support. Otherwise, after 120 days, the appliance becomes locked to additional on-premises data transfers. If the appliance becomes locked during a data transfer, return it and create a new job to transfer the rest of your data. If the AWS Snowball Edge appliance becomes locked during an import job, we can still transfer the existing data on the appliance into Amazon S3.
- If you encounter unexpected errors using an AWS Snowball Edge appliance, we want to hear about it. Make a copy of the relevant 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 Commands for the Snowball Client (p. 37).

Troubleshooting Connection Problems

The following can help you troubleshoot issues you might have with connecting to your AWS Snowball Edge appliance:

- Routers and switches that work at a rate of 100 megabytes per second won't work with an AWS Snowball Edge appliance. 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 Edge, see Performance (p. 29) 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 Edge:

- If you're using Linux and you can't upload files with UTF-8 characters to an AWS Snowball Edge appliance, it might be because your Linux server doesn't recognize UTF-8 character encoding. You can correct this issue by installing the `locales` package on your Linux server 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're communicating with the AWS Snowball Edge appliance 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*.
- When you use the Amazon S3 Adapter for Snowball with the AWS CLI, you can work with files or folders with spaces in their names, such as `my photo.jpg` or `My Documents`. However, make sure that you handle the spaces properly. For more information, see Specifying Parameter Values for 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 AWS Snowball Edge Logs (p. 40).
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.*

Additional Information for Snowball Edge

Following, you can find additional information for Snowball Edge, including a discussion of features that are still supported but not necessarily recommended.

Topics

Using the Snowball Client

> **Note**
> In January 2018, there was a feature update for clusters, making them leaderless. The cluster update is backward-compatible with older clusters. Following, you can find the documentation for the original Snowball client for Snowball Edge. If you're looking for the updated content, see Using the Snowball Client (p. 36).

Following, you can find information about how to get and use the Snowball client with your AWS Snowball Edge appliance. The Snowball client is a standalone terminal application that you run on your local server to unlock the appliance and get credentials, logs, and status information. You can also use the client for administrative tasks for a cluster. When you read and write data to the AWS Snowball Edge appliance, you use the Amazon S3 Adapter for Snowball or the file interface.

Downloading and Installing the Snowball Client

You can download and install the Snowball client from the AWS Snowball Tools Download page. When you've reached that page, 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:

- **Microsoft Windows** – When the client has been installed, you can run it from any directory without any additional preparation.
- **Linux** – The Snowball client 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.

Commands for the Snowball Client

Following, you can find information on the Snowball client commands, including examples of use and sample outputs.

Unlocking

The `unlock` command unlocks access to the AWS Snowball Edge appliance with the AWS Snowball Edge appliance's IP address and your credentials.

If you're unlocking a cluster, you use the `-i` option for your primary node and the `-s` option for each secondary node, as in the following example. All nodes are identical until you assign one to be the primary node. The primary node is the leader of the cluster and performs most of the behind-the-scenes management of the cluster. For more information on clusters, see Clustering Overview (p. 106).

Usage

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

Example Single Device Unlock Input

```
snowballEdge unlock -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
```

Example Single Device Unlock Output

```
The Snowball Edge unlock status is: UnlockSnowballResult(status=UNLOCKING)
```

Example Cluster Unlock Input

```
snowballEdge unlock -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234 -s
 192.0.2.1 -s 192.0.2.2 -s 192.0.2.3 -s 192.0.2.4
```

Example Cluster Unlock Output

```
Snowball Unlock Status: SUCCESS
Node Ip:  [Node Id, State]
192.0.2.0 : [JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab00, AVAILABLE]
192.0.2.1 : [JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab01, AVAILABLE]
192.0.2.2 : [JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab02, AVAILABLE]
192.0.2.3 : [JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab03, AVAILABLE]
192.0.2.4 : [JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab04, AVAILABLE]
Total Size: 225 TB
Free Space: 121 TB
Primary Node: 192.0.2.0
S3 Endpoint running at: http://192.0.2.0:8080
Durability Status: HEALTHY - The Snowball Edge cluster is highly durable.
```

Getting Credentials

The `credentials` command returns the set of local credentials (an access key and a secret key). You use these to sign your requests when using the AWS CLI or your own application with the Amazon S3 Adapter for Snowball to read and write data to the AWS Snowball Edge appliance. These credentials are only associated with an individual job for AWS Snowball, and they can only be used on the appliance or cluster of appliances. The appliance or appliances don't have any AWS Identity and Access Management (IAM) permissions in the AWS Cloud.

> **Note**
> If you're using the AWS CLI with the Snowball Edge, you must use these credentials when you configure the CLI. For information on configuring credentials for the CLI, see Quick Configuration in the *AWS Command Line Interface User Guide.*

Usage

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

Example Input

```
snowballEdge credentials -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-
ABCDE-01234
```

Example Output

```
[snowballEdge]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Getting Status

The `status` command returns the status of an AWS Snowball Edge appliance. When you run this command to check the status of a cluster, the IP address you should use is the IP address of the primary node. For more information on clusters, see Clustering Overview (p. 106).

Usage

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

Example Input

```
snowballEdge status -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
```

Example Output

```
Snowball Status: SUCCESS
S3 Endpoint running at: http://192.0.2.0:8080
Total Size: 82 TB
Free Space: 74 TB
```

Getting AWS Lambda Powered by AWS Greengrass and File Interface Logs

The `logs` command saves a copy of the Lambda and file interface logs to the specified path on your server in an archive format. You can specify the path with the `-o` option.

Usage

```
snowballEdge logs -i IP Address -m Path/to/manifest/file -u 29 character unlock code -
o Path/to/logs
```

Example Input

```
snowballEdge logs -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234 -o ~/
```

Example Output

```
Snowball logs written to: /home/dan/snowballLogs-14EXAMPLE8009.zip
```

Getting Device Logs

The `servicelogs` saves an encrypted blob of support logs from a Snowball Edge or cluster to the specified output path on your server. Give these logs to AWS Support when debugging issues.

Usage

```
snowballEdge servicelogs -o Path/to/service/logs/output/directory -i IP Address -m Path/to/
manifest/file -u 29 character unlock code
```

Example Input

```
snowballEdge servicelogs -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-
ABCDE-01234 -o ~/
```

Example Output

```
Snowball servicelogs written to: /home/dan/snowballLogs-14EXAMPLE8010.bin
```

Removing a Node from a Cluster

The `removenode` command removes a node from a cluster of Snowball Edge devices. Use this command if a node has become unavailable. For more information on clusters, see Clustering Overview (p. 106).

> **Important**
> Use the `removenode` command only when you are removing an unresponsive node. Don't use this command to remove a healthy node.

If a node was accidentally powered off or disconnected from the network and was therefore temporarily unavailable to the rest of the cluster, you don't have to use this command. In this case, simply plug the previously unavailable node back into power and the network. The node should then rejoin the cluster automatically.

Usage

```
snowballEdge removenode -i IP Address -m Path/to/manifest/file -u 29 character unlock code
 -n node id of unavailable node
```

Example Input

```
snowballEdge removenode -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
 -n JIDfEXAMPLE-1234-abcd-1234-4EXAMPLE8ae9
```

Example Output

```
The node: JIDfEXAMPLE-1234-abcd-1234-4EXAMPLE8ae9 was successfully removed from the
 cluster.
```

Adding a Node to a Cluster

The `addnode` command adds a node to a cluster of Snowball Edge devices. You can use this command to replace an unavailable node with a new node that you ordered as a replacement. For more information on clusters, see Clustering Overview (p. 106).

Important
Don't attempt to re-add any node that you previously removed from the cluster with the `snowballEdge removenode` command.

Usage

```
snowballEdge addnode -i IP Address -m Path/to/manifest/file -u 29 character unlock code -
a IP address of the node to be added
```

Example Input

```
snowballEdge addnode -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234 -a
 192.0.2.5
```

Example Output

```
The node: 192.0.2.5 was successfully added to the cluster.
```

Unlocking the AWS Snowball Edge Appliance

To unlock the AWS Snowball Edge appliance, run the `snowballEdge unlock` command. This command authenticates your access to the AWS Snowball Edge appliance. To run this command, the AWS Snowball Edge appliance you use for your job must be onsite, plugged into power and network, and turned on. In addition, the LCD display on the AWS Snowball Edge appliance's front must indicate that the appliance is ready for use.

To authenticate the Snowball client's access to an AWS Snowball Edge appliance

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 AWS Snowball Edge appliance when it was being prepared. This comparison verifies that you have the right AWS Snowball Edge appliance 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 AWS Snowball Edge appliance while it's at your facility.
2. Locate the IP address for the AWS Snowball Edge appliance on the AWS Snowball Edge appliance's LCD display. When the AWS Snowball Edge appliance 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 Edge (p. 34).
3. Execute the `snowballEdge unlock` command to authenticate your access to the AWS Snowball Edge appliance with the AWS Snowball Edge appliance's IP address and your credentials, as follows:

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

   Example

   ```
   snowballEdge unlock -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
   ```

Clustering Overview

Note
In January 2018, there was a feature update to for clusters, making them leaderless. The cluster update is backwards compatible with older clusters. This is the original cluster documentation for the Snowball Edge. If you're looking for the updated content, see

A cluster is a logical grouping of AWS Snowball Edge devices, in groups of 5 to 10 devices. A cluster is created as a single job, which offers increased durability and storage capacity. In the following topic, you can find information about Snowball Edge clusters. This information includes conceptual, usage, and administrative information, in addition to walkthroughs for common Snowball Edge procedures.

> **Note**
> In January 2018, there was a feature update for clusters, making them leaderless. The cluster update is backward-compatible with older clusters. However, if you're looking for the original cluster documentation, see Additional Information for Snowball Edge (p. 101).

Topics

Clustering Overview

For the AWS Snowball service, a cluster is a collective of Snowball Edges, used as a single logical unit, for local storage and compute purposes.

A cluster offers two primary benefits over a standalone Snowball Edge for local storage and compute purposes:

- **Increased Durability** – The data stored in a cluster of Snowball Edge appliances enjoys increased data durability over a single device. In addition, the data on the cluster remains as safe and viable as it was previously, despite possible Snowball Edge outages in the cluster. Clusters can withstand the loss of two nodes before the data is in danger. You can also add or replace nodes.
- **Increased Storage** – The total available storage is 45 terabytes of data per node in the cluster. Thus, in a five-node cluster there are 225 terabytes of available storage space. In contrast, there are about 82 terabytes of available storage space in a standalone Snowball Edge. Clusters that have more than five nodes have even more storage space.

A cluster of Snowball Edge devices is made of leaderless nodes. Any node can write data to and read data from the entire cluster, and all nodes are capable of performing the behind-the-scenes management of the cluster.

Snowball Edge Cluster Quorums

A quorum represents the minimum number of Snowball Edge devices in a cluster that must be communicating with each other to maintain some level

of operation. There are two levels of quorum for Snowball Edge clusters—a read/write quorum and a read quorum.

Let's say you upload your data to a cluster of Snowball Edge devices. With all devices healthy, you have a read/write quorum for your cluster.

If one of those nodes goes offline, you reduce the operational capacity of the cluster. However, you can still read and write to the cluster. In that sense, with the cluster operating all but one node, the cluster still has a read/write quorum.

If two nodes in your cluster are down, any additional or ongoing write operations fail. But any data that was successfully written to the cluster can be accessed and read. This is called a read quorum.

Finally, suppose that a third node loses power. Then the cluster is offline, and the data in the cluster is unavailable. You might be able fix this, or the data might be permanently lost, depending on the severity of the event. If it is a temporary external power event, and you can power the three Snowball Edges back on and unlock all the nodes in the cluster, then your data is available again.

> **Important**
> If a minimum quorum of healthy nodes doesn't exist, contact AWS Support.

You can determine the quorum state of your cluster by determining your node's lock state and network reachability. The `snowballEdge describe-cluster` command reports back the lock and network reachability state for every node in an unlocked cluster. Ensuring that the appliances in your cluster are healthy and connected is an administrative responsibility that you take on when you create the cluster job. For more information on the different client commands, see Commands for the Snowball Client (p. 37).

Considerations for Cluster Jobs for AWS Snowball Edge

Keep the following considerations in mind when planning to use a cluster of Snowball Edges:

- We recommend that you have a redundant power supply to reduce potential performance and stability issues for your cluster.
- As with standalone local storage and compute jobs, the data stored in a cluster can't be imported into Amazon S3 without ordering additional appliances as a part of separate import jobs. Then you can transfer the data from the cluster to those appliances and import the data when you return the appliances for the import jobs.
- To get data onto a cluster from Amazon S3, create a separate export job and copy the data from the appliances of the export job onto the cluster.
- You can create a cluster job from the console, the AWS CLI, or one of the AWS SDKs. For a guided walkthrough of creating a job, see Getting Started with an AWS Snowball Edge Appliance (p. 20).
- Cluster nodes have node IDs. A *node ID* is the same as the job ID for a device that you can get from the console, the AWS CLI, the AWS SDKs, and the Snowball client. You can use node IDs to remove old nodes from

clusters. You can get a list of node IDs by using the `snowballEdge describe-device` command on an unlocked device or the `describe-cluster` on an unlocked cluster.

- The lifespan of a cluster is limited by the security certificate granted to the cluster devices when the cluster is provisioned. By default, Snowball Edge devices can be used for up to 120 days before they need to be returned. At the end of that time, the devices stop responding to read/write requests. If you need to keep one or more devices for longer than 120 days, contact AWS Support.
- When AWS receives a returned appliance that was part of a cluster, we perform a complete erasure of the appliance. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

Related Topics

Beyond the content presented in this topic, you can find other topics in this guide that are relevant to clusters:

- Getting Started with an AWS Snowball Edge Appliance (p. 20) – This section outlines how to get started creating your first job. The techniques in this section work for all job types, including cluster jobs.
- Commands for the Snowball Client (p. 37) – This section contains a list of commands for the Snowball client tool. These commands include the Snowball Edge administrative commands to unlock a cluster, get the status information for the nodes and the cluster as a whole, remove unavailable nodes, and add new nodes.
- Administrating a Cluster (p. 65) – This section contains information about the administrative tasks you perform with a cluster, like adding and removing nodes, and includes helpful procedures.

Administrating a Cluster

Following, you can find information about administrative tasks to operate a healthy cluster of Snowball Edge devices. The primary administrative tasks are covered in the following topics.

Topics

Most administrative tasks require that you use the Snowball client and its commands that perform the following actions:

- Unlocking AWS Snowball Edge Devices (p. 38)
- Getting Device Status (p. 42) of a cluster
- Removing a Node from a Cluster (p. 45)
- Adding a Node to a Cluster (p. 45)

Reading and Writing Data to a Cluster

After you've unlocked a cluster, you're ready to read and write data to it. You can use the Amazon S3 Adapter for Snowball to read and write data to a cluster. For more information, see Using the Amazon S3 Adapter (p. 46).

To write data to a cluster, you must have a read/write quorum with no more than one unavailable node. To read data from a cluster, you must have a read quorum of no more than two unavailable nodes. For more information on quorums, see Snowball Edge Cluster Quorums (p. 63).

Reconnecting an Unavailable Cluster Node

A node can become temporarily unavailable due to an issue (like power or network loss) without damaging the data on the node. When this happens, it affects the status of your cluster. A node's network reachability and lock status is reported in the Snowball client by using the `snowballEdge describe-cluster` command.

We recommend that you physically position your cluster so you have access to the front, back, and top of all nodes. This way, you can access the power and network cables on the back, the shipping label on the top to get your node ID, and the LCD screen on the front of the device for the IP address and other administrative information.

When you detect that a node is unavailable, we recommend that you try one of the following procedures, depending on the scenario that caused the unavailability.

To reconnect an unavailable node

1. Ensure that the node has power.
2. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
3. Wait for the node to finish powering up, if it needed to be powered up.
4. Run the `snowballEdge unlock-cluster` command, or the `snowballEdge associate-device` command. For an example, see Unlocking AWS Snowball Edge Devices (p. 38).

To reconnect an unavailable node that lost network but didn't lose power

1. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
2. Run the `snowballEdge describe-device` command to see when the previously unavailable node is added back to the cluster. For an example, see Getting Device Status (p. 42).

When you have performed the preceding procedures, your nodes should be working normally. You should also have a read/write quorum. If that's not the case, then one or more of your nodes might have a more serious issue and might need to be removed from the cluster.

Removing an Unhealthy Node from a Cluster

Rarely, a node in your cluster might become unhealthy. If the node is unavailable, we recommend going through the procedures listed in Reconnecting an Unavailable Cluster Node (p. 65) first.

If doing so doesn't resolve the issue, then the node might be unhealthy. An unhealthy node can occur if the node has taken damage from an external source, if there was an unusual electrical event, or if some other unlikely event occurs. If this happens, you need to remove the node from the cluster before you can add a new node as a replacement.

When you detect that a node is unhealthy and needs to be removed, we recommend that you do so with the following procedure.

To remove an unhealthy node

1. Ensure that the node is unhealthy and not just unavailable. For more information, see Reconnecting an Unavailable Cluster Node (p. 65).
2. Disconnect the unhealthy node from the network and power it off.
3. Run the `snowballEdge dissassociate-device` Snowball client command. For more information, see Removing a Node from a Cluster (p. 45).
4. Order a replacement node using the console, the AWS CLI, or one of the AWS SDKs.
5. Return the unhealthy node to AWS. When we have the node, we perform a complete erasure of the device. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

After you successfully remove a node, your data is still available on the cluster if you still have a read quorum. To have read quorum, a cluster must have no more than two unavailable nodes. Therefore, we recommend that you order replacement nodes as soon as you remove an unavailable node from the cluster.

Adding or Replacing a Node in a Cluster

You can add a new node after you have removed an unhealthy node from a cluster. You can also add a new node to increase local storage.

To add a new node, you first need to order a replacement. You can order a replacement node from the console, the AWS CLI, or one of the AWS SDKs. If you're ordering a replacement node from the console, you can order replacements for any job that hasn't been canceled or completed.

To order a replacement node from the console

1. Sign in to the AWS Snowball Management Console.
2. Find and choose a job for a node that belongs to the cluster that you created from the Job dashboard.
3. For **Actions**, choose **Replace node**.

Doing this opens the final step of the job creation wizard, with all settings identical to how the cluster was originally created.

4. Choose **Create job**.

Your replacement Snowball Edge is now on its way to you. When it arrives, use the following procedure to add it to your cluster.

To add a replacement node

1. Position the new node for the cluster such that you have access to the front, back, and top of all nodes.
2. Ensure that the node has power.
3. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
4. Wait for the node to finish powering up, if it needed to be powered on.
5. Run the `snowballEdge associate-device` command. For an example, see Adding a Node to a Cluster (p. 45).

(p.).

For the AWS Snowball service, a cluster is a collective of AWS Snowball Edge appliances, used as a single logical unit, for local storage and compute purposes.

A cluster offers two primary benefits over a standalone Snowball Edge for local storage and compute purposes:

- **Increased Durability** – The data stored in a cluster of Snowball Edges enjoys increased data durability over a single device. In addition, the data on the cluster remains as safe and viable as it was previously, despite possible Snowball Edge outages in the cluster. Clusters can withstand the loss of two nodes before the data is in danger. You can also add or replace nodes.
- **Increased Storage** – The total available storage is 45 terabytes of data per node in the cluster. Thus, in a five-node cluster there's 225 terabytes of available storage space. In contrast, there's about 82 terabytes of available storage space in a standalone Snowball Edge. Clusters that have more than five nodes have even more storage space.

A cluster of Snowball Edge devices is made of nodes. There are two types of nodes: primary nodes and secondary nodes. When writing data to a cluster, data is written from your server, across your internal network, and to the primary node of the cluster. The primary node then writes the data to the secondary nodes.

The primary node is the leader of the cluster and performs most of the behind-the-scenes management of the cluster. You designate the primary node from among all the nodes of the cluster when you unlock the cluster for the first time. You can change the primary node if the current one becomes unavailable.

Snowball Edge Cluster Quorums

A quorum represents the minimum number of Snowball Edge devices in a cluster that must be communicating with each other to maintain some level of operation. There are two levels of quorum for Snowball Edge clusters—a read/write quorum and a read quorum.

Let's say you've uploaded your data to a cluster of Snowball Edge devices. With all devices healthy, you have a read/write quorum for your cluster.

If one of those nodes goes offline, you have reduced the operational capacity of the cluster. However, you can still read and write to the cluster. In that sense, with the cluster operating all but one node, the cluster still has a read/write quorum.

In this same example, if the external power failure took out two of the nodes in your cluster, any additional or ongoing write operations fail. But any data that was successfully written to the cluster can be accessed and read. This situation is called a read quorum.

Finally, in this example, suppose that a third node loses power. Then the cluster is offline, and the data in the cluster is unavailable. You might be able fix this, or the data might be permanently lost, depending on the severity of the event. If it is a temporary external power event, and you can power the three Snowball Edges back on and unlock all the nodes in the cluster, then your data will be available again.

Important
If a minimum quorum of healthy nodes doesn't exist, contact AWS Support.

You can check the quorum state of your cluster by running the `snowballEdge status` command. Ensuring that the appliances in your cluster are healthy and connected is an administrative responsibility that you take on when you create the cluster job.

Considerations for Cluster Jobs for AWS Snowball Edge

Keep the following considerations in mind when planning to use a cluster of Snowball Edges:

- We recommend that you have a redundant power supply to reduce potential performance and stability issues for your cluster.
- As with standalone local storage and compute jobs, the data stored in a cluster can't be imported into Amazon S3 without ordering additional appliances as a part of separate import jobs. Then you could transfer the data from the cluster to those appliances and import the data when you return the appliances for the import jobs.
- To get data onto a cluster from Amazon S3, you have to create a separate export job and copy the data from the appliances of the export job onto the cluster.
- You can create a cluster job from the console, the AWS CLI, or one of the AWS SDKs. For a guided walkthrough of creating a job, see Getting Started with an AWS Snowball Edge Appliance (p. 20).
- Cluster nodes have node IDs. A *node ID* is the same as the job ID for a device that you can get from the console, the AWS CLI, the AWS SDKs, and the Snowball client. You can use node IDs to remove old nodes from clusters. You can get a list of node IDs by using the `snowballEdge status` command.
- The lifespan of a cluster is limited by the security certificate granted to the cluster devices when the cluster is provisioned. By default, Snowball Edge devices can be used for up to 120 days before they need to be returned. At the end of that time, the devices stop responding to read/write requests. If you need to keep one or more devices for longer than 120 days, contact AWS Support.
- When AWS receives a returned appliance that was part of a cluster, we perform a complete erasure of the appliance. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

Related Topics

Beyond the content presented in this topic, you can find other topics in this guide that are relevant to clusters:

- Getting Started with an AWS Snowball Edge Appliance (p. 20) – This section outlines how to get started creating your first job. The techniques in this section work for all job types, including cluster jobs.

- Commands for the Snowball Client (p. 101) – This section contains a list of commands for the Snowball client tool. These commands include the Snowball Edge administrative commands to unlock a cluster, get the status information for the nodes and the cluster as a whole, remove unavailable nodes, and add new nodes.
- Administrating a Cluster (p. 113) – This section contains information about the administrative tasks you perform with a cluster like adding and removing nodes, and includes helpful procedures.

Administrating a Cluster

Following, you can find information about administrative tasks to operate a healthy cluster of Snowball Edge devices. The primary administrative tasks are covered in the following topics.

Topics

Most administrative tasks require that you use the Snowball client and its commands that perform the following actions:

- Unlocking (p. 101) a cluster
- Getting Status (p. 103) of a cluster
- Removing a Node from a Cluster (p. 104)
- Adding a Node to a Cluster (p. 104)

Reading and Writing Data to a Cluster

After you've unlocked a cluster, you're ready to read and write data to it. Currently, we recommend that you use the Amazon S3 Adapter for Snowball to read and write data to a cluster. For more information, see Using the Amazon S3 Adapter (p. 46).

To write data to a cluster, you must have a read/write quorum no more than one unavailable node. To read data from a cluster, you must have a read quorum of no more than two unavailable nodes. For more information on quorums, see Snowball Edge Cluster Quorums (p. 111).

Reconnecting an Unavailable Cluster Node

A node can become temporarily unavailable due to an issue (like power or network loss) without damaging the data on the node. When this happens, it affects the status of your cluster. A node's temporary unavailability is reported in the Snowball client by using the `snowballEdge status` command.

Because of this functionality, we recommend that you physically position your cluster so you have access to the front, back, and top of all nodes. This way, you can access the power and network cables on the back, the shipping label on the top to get your node ID, and the LCD screen on the front of the device for the IP address and other administrative information.

You can detect that a node is unavailable with the Snowball client `snowballEdge status` command. This command reports back the quorum status of the entire cluster and also the status of each available node. If a primary node is unavailable, then the status command returns an error. If a secondary node is not available, then it should be listed as unavailable.

When you detect that a node is unavailable, we recommend that you try one of the following procedures, depending on the scenario that caused the unavailability.

To reconnect an unavailable primary node

1. Ensure that the node has power.
2. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
3. Wait for the node to finish powering up, if it needed to be powered up.
4. Run the `snowballEdge unlock` command. For an example, see Unlocking (p. 101).

To reconnect an unavailable secondary node that was powered off

The following procedure shows you how to add a healthy node to a cluster.

1. Power the unavailable node back on.
2. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
3. Wait for the node to finish powering up.
4. Run the `snowballEdge addnode` command. For an example, see Adding a Node to a Cluster (p. 45).

To reconnect an unavailable secondary node that lost network but didn't lose power

1. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
2. Run the `snowballEdge status` command to see when the previously unavailable node is added back to the cluster. For an example, see Getting Device Status (p. 42).

When you have performed the preceding procedures, your nodes should be working normally. You should also have a read/write quorum. If that's not the case, then one or more of your nodes might have a more serious issue and might need to be removed from the cluster.

Changing the Primary Node on a Cluster

The primary node is the leader of a cluster and performs most behind-the-scenes management of the cluster. You designate the primary node from among all the nodes of the cluster when you unlock the cluster for the first time. You can also change the primary node if the current one becomes unavailable.

To change which node is primary because the old primary node is unavailable

1. Power off all cluster nodes by pressing the power button above the LCD screen for no longer than two seconds.
2. Power on all the cluster nodes.
3. Wait for the node to finish powering up, if it needed to be powered up.
4. Make sure that the nodes are connected to the same internal network and make a note of their IP addresses.
5. Run the `snowballEdge unlock` Snowball client command with a different node as the primary node.

Removing an Unhealthy Node from a Cluster

Rarely, a node in your cluster might become unhealthy. If the node is unavailable, we recommend going through the procedures listed in Reconnecting an Unavailable Cluster Node (p. 113) first.

If doing so doesn't resolve the issue, then the node might be unhealthy. An unhealthy node can occur if the node has taken damage from an eternal source, if there was an unusual electrical event, or if some other unlikely event occurs. If this happens, you need to remove the node from the cluster before you can add a new node as a replacement.

If the node that needs to be removed is the primary node, before going through the following procedure you should change which node is your primary node. For more information, see Changing the Primary Node on a Cluster (p. 114).

When you detect that a secondary node is unhealthy and needs to be removed, we recommend that you do so with the following procedure.

To remove an unhealthy node

1. Ensure that the node is unhealthy and not just unavailable. For more information, see Reconnecting an Unavailable Cluster Node (p. 113).
2. Disconnect the unhealthy node from the network and power it off.
3. Run the `snowballEdge removenode` Snowball client command. For more information, see Removing a Node from a Cluster (p. 104).
4. Turn the power off on each node in the cluster once again.
5. Power each node in the cluster back on again.
6. Unlock the cluster once again, but without the unhealthy node.
7. Order a replacement node using the console, the AWS CLI, or one of the AWS SDKs.
8. Return the unhealthy node to AWS. When we have the node, we perform a complete erasure of the device. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

After you successfully remove a node, your data is still available on the cluster if you still have a read quorum. To have read quorum, a cluster must have no more than two unavailable nodes. Therefore, we recommend that you order replacement nodes as soon as you remove an unavailable node from the cluster.

Adding or Replacing a Node in a Cluster

You can add a new node after you have removed an unhealthy node from a cluster. You can also add a new node to increase local storage.

To add a new node, you first need to order a replacement. You can order a replacement node from the console, the AWS CLI, or one of the AWS SDKs. If you're ordering a replacement node from the console, you can order replacements for any job that hasn't been canceled or completed.

To order a replacement node from the console

1. Sign in to the AWS Snowball Management Console.
2. Find and choose a job for a node that belongs to the cluster that you created from the Job dashboard.
3. For **Actions**, choose **Replace node**.

   Doing this opens the final step of the job creation wizard, with all settings identical to how the cluster was originally created.
4. Choose **Create job**.

Your replacement Snowball Edge is now on its way to you. When it arrives, use the following procedure to add it to your cluster.

To add a replacement node

1. Position the new node for the cluster such that you have access to the front, back, and top of all nodes.
2. Ensure that the node has power.
3. Ensure that the node is connected to the same internal network that the rest of the cluster is on.
4. Wait for the node to finish powering up, if it needed to be powered up.
5. Run the `snowballEdge add` command. For an example, see Adding a Node to a Cluster (p. 104).

Document History

The following table describes documentation releases for the AWS Snowball Edge appliance.

- **API version: 1.0**
- **Latest documentation update:** March 20, 2018

Change	Description	Date
Automatically extracted batches of small files are now supported	You can now batch many small files together into a larger archive, and specify that those batches are automatically extracted when the data is imported into Amazon S3. Batching small files together can significantly improve your transfer performance when moving data from your on-premises server to a Snowball Edge device. For more information, see Batching Small Files (p. 47).	March 20, 2018
Major feature revision to the Snowball client, and cluster update	The new major feature revision for the Snowball client includes performance improvements, profiles, and support for the cluster update. For more information, see Using the Snowball Client (p. 36). Clusters are now leaderless. All nodes can read and write data to the cluster. For more information, see Using an AWS Snowball Edge Cluster (p. 63).	February 5, 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. 68).	December 18, 2017
Improved AWS CLI support for the Amazon S3 Adapter for Snowball	You can now use the `s3 sync` command with the Amazon S3 Adapter for Snowball to sync data between a Snowball Edge and your local computer. For more information, see Supported AWS CLI Commands for Amazon S3 (p. 48).	November 10, 2017

Change	Description	Date
Updated file interface file size support	The file interface can now support files up to 150 GB in size. For more information, see Using the File Interface for the AWS Snowball Edge (p. 52).	October 4, 2017
New AWS Region supported	AWS Snowball Edge is now supported in the Asia Pacific (Tokyo) region, with region-specific shipping options. For more information, see Shipping Considerations for AWS Snowball (p. 68).	September 19, 2017
New AWS Region supported	AWS Snowball Edge 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. 68).	August 8, 2017
Updated AWS Greengrass and Lambda functionality	Lambda functions running on AWS Snowball Edge devices can now be added, updated, removed, or replaced, once the devices are on-premises. In addition, AWS Snowball Edge devices can now be used as AWS Greengrass core devices. For more information, see Using AWS Lambda with an AWS Snowball Edge (p. 58).	July 25, 2017
New AWS Region supported	AWS Snowball Edge is now supported in the Canada (Central) region, with region-specific shipping options. For more information, see Shipping Considerations for AWS Snowball (p. 68).	June 29, 2017
Updated file interface functionality	With the file interface, you can now choose the Network File System (NFS) clients that are allowed to access the file share on the Snowball Edge, in addition to accessing other support and troubleshooting features. For more information, see Using the File Interface for the AWS Snowball Edge (p. 52).	June 21, 2017

Change	Description	Date
Updated cluster functionality	Clusters can now be created in groups of 5–10 AWS Snowball Edge appliances. For more information, see Using an AWS Snowball Edge Cluster (p. 63).	June 5, 2017
Documentation update	Documentation 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
Updated compute information	The following updates have been made for AWS Lambda powered by AWS Greengrass functions: • Event objects are now JSON objects like their cloud-based counterparts. • When you choose a function for a job, you also choose a specific version of the function. Each version of a function now has a separate Amazon Resource Name (ARN). • To improve latency, functions are loaded in memory when a job is created. When creating a compute job, keep in mind that you have a total of 3.744 GB of memory available for all the functions. If you need more functions than the memory can support, you need to create more jobs.	December 6, 2016
Introducing AWS Snowball Edge	The AWS Snowball service now has two appliances, the standard Snowball and the AWS Snowball Edge appliance. With the new Snowball Edge, you can now create local storage and compute jobs, harnessing the power of the AWS Cloud locally, without an internet connection.	November 30, 2016

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.