AWS Snowcone User Guide

AWS Snowcone

aws

AWS Snowcone User Guide: AWS Snowcone

Table of Contents

What Is AWS Snowcone?

AWS Snowcone is a portable, rugged, and secure device for edge computing and data transfer. You can use a Snowcone device to collect, process, and move data to the AWS Cloud, either offline by shipping the device to AWS, or online by using AWS DataSync.

It can be challenging to run applications in austere (non–data center) edge environments, or where there is a lack of consistent network connectivity. These locations often lack the space, power, and cooling needed for data center IT equipment.

Snowcone is available in two flavors:

- **Snowcone** – Snowcone has two vCPUs, 4 GB of memory, and 8 TB of hard disk drive (HDD) based storage.
- **Snowcone SSD** – Snowcone SSD has two vCPUs, 4 GB of memory, and 14 TB of solid state drive (SSD) based storage.

With two CPUs and terabytes of storage, a Snowcone device can run edge computing workloads that use Amazon Elastic Compute Cloud (Amazon EC2) instances, and store data securely.

Snowcone devices are small (8.94" x 5.85" x 3.25" / 227 mm x 148.6 mm x 82.65 mm), so they can be placed next to machinery in a factory to collect, format, and transport data back to AWS for storage and analysis. A Snowcone device weighs about 4.5 lbs. (2 kg), so you can carry one in a backpack, use it with battery-based operation, and use the Wi-Fi interface to gather sensor data.

Snowcone devices offer a file interface with Network File System (NFS) support. Snowcone devices support data transfer from on-premises Windows, Linux, and macOS servers and file-based applications through the NFS interface.

Like AWS Snowball, AWS Snowcone has multiple layers of security encryption capabilities. You can use either of these services to collect, process, and transfer data to AWS, and run edge computing workloads that use Amazon EC2 instances. Snowcone is designed for data migration needs up to dozens of terabytes. It can be used in space-constrained environments where Snowball Edge devices don't fit.

Use Cases

You can use AWS Snowcone devices for the following use cases:

- For edge computing applications, to collect data, process the data to gain immediate insight, and then transfer the data online to AWS.
- To transfer data that is continuously generated by sensors or machines online to AWS in a factory or at other edge locations.
- To distribute media, scientific, or other content from AWS storage services to your partners and customers.
- To aggregate content by transferring media, scientific, or other content from your edge locations to AWS.
- For one-time data migration scenarios where your data is ready to be transferred, Snowcone offers a quick and low-cost way to transfer up to 8 TB or 14 TB of data to the AWS Cloud by shipping the device back to AWS.

For mobile deployments, a Snowcone device can run on specified battery power. For a light workload at 25 percent CPU usage, the device can run on a battery for up to approximately 6 hours. You can use the Wi-Fi interface on your Snowcone device to collect data from wireless sensors. An AWS Snowcone device is low power, portable, lightweight, and vibration resistant, so you can use it in a wide variety of remote and austere locations.

Pricing

You can order a Snowcone device for pay per use and keep the device for up to four years. For information about AWS Snowcone pricing and fees, see AWS Snowcone pricing.

How AWS Snowcone Works

AWS Snowcone is a portable device used for edge computing and data transfer. To get started, you request one or more Snowcone devices in the AWS Management Console based on how much data you need to transfer and the compute performance required. The Amazon Simple Storage Service (Amazon S3) buckets, data, and Amazon Elastic Compute Cloud (Amazon EC2) Amazon Machine Images (AMIs) that you choose are automatically configured, encrypted, and pre-installed on your devices. The AWS DataSync agent is also pre-installed before your devices are shipped to you.

You typically receive your Snowcone devices within approximately 4–6 days. To receive multiple AWS Snowcone devices, you must set up a job order on the console for each Snowcone device.

When your device arrives, you connect it to your on-premises network and set the IP address either manually or automatically with Dynamic Host Configuration Protocol (DHCP). You must download and install AWS OpsHub for Snow Family, a graphical user interface (GUI) application for managing your Snowcone device. You can install it on any Windows or macOS client machine, such as a laptop.

When you open AWS OpsHub and unlock the device, you see a dashboard showing your device and its system metrics. You can then launch instances to deploy your edge applications or migrate your data to the device with just a few clicks in AWS OpsHub.

When your compute or data transfer job is completed and the device is ready to be returned, the E Ink shipping label automatically updates the return address, ensuring that the Snowcone device is delivered to the correct AWS facility. When the device ships, you can receive tracking status through messages sent by Amazon Simple Notification Service (Amazon SNS), generated texts and emails, or directly from the console.

Topics

AWS Snowcone Workflow

You can create three different job types. Although the job types differ in their use cases, they all have the following workflow for ordering, receiving, and returning the device.

The workflow

1. **Create the job** – You create each job on the AWS Snow Family Management Console or programmatically through the job management API and choose a device type—**Snowcone** or **Snowcone SSD**—depending on your use case. You can track the status of the job on the AWS Management Console or through the Snowcone API.
2. **A device is prepared for your job** – AWS prepares an AWS Snowcone device for your job, and the status of your job changes to **Preparing Snowcone**.
3. **A device is shipped to you by your Region's carrier** – The shipping carrier takes over from here, and the status of your job now changes to **In transit to you**. You can find your tracking number and a

link to the tracking website on the AWS Snow Family Management Console console or with the job management API. For information about who your Region's carrier is, see Shipping Considerations for AWS Snowcone (p. 111).

4. **Get a Snowcone power supply** – To maintain the smallest footprint, Snowcone devices do not ship with a power supply. Snowcone uses a 45 watt USB-C connected power supply. It can also be powered by a portable battery. For more information, see AWS Snowcone Power Supply and Accessories (p. 13).

5. **Receive the device** – A few days later, your Region's shipping carrier delivers the AWS Snowcone device to the address that you provided when you created the job. The status of your job changes to **Delivered to you**. The device does not arrive in a box because the device is its own shipping container.

6. **Get your credentials and download the AWS OpsHub or Snowball Edge client for Snow Family application** – Get ready to start transferring data by getting your credentials, your job manifest, and the manifest's unlock code, and then downloading the Snowball Edge client.

   • Get the manifest for your device from the console or with the job management API when the device is on-premises at your location. The manifest is used to authenticate your access to the device. The manifest is encrypted, so only the unlock code can decrypt it.

   • 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. To prevent unauthorized access to the device while it's at your facility, we recommend that you keep the unlock code in a safe location that is different from the location of the manifest.

   • AWS OpsHub for Snow Family is an application for managing Snow Family Devices, including Snowcone. The AWS OpsHub for Snow Family GUI helps you set up and manage Snowcone devices so that you can quickly run your edge compute workloads and migrate data to the AWS Cloud. With just a few clicks, you can use AWS OpsHub to unlock and configure your Snowcone device, drag and drop data, launch applications, or monitor device metrics. You can download and install it on Windows or macOS client machines, such as a laptop. There is no cost to use AWS OpsHub.

     Download AWS OpsHub from AWS Snowball resources. For more information about AWS OpsHub, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

   • The Snowball Edge client is the tool that you use to manage the flow of data from the device to your on-premises data destination.

7. **Position the hardware** – Move the device into your data center and open it following the instructions on the case. Connect the device to a power supply and your local network.

8. **Power on the device** – Power on the device by pressing the power button above the LCD display. Wait a few minutes, and the **Ready** screen appears.

9. **Get the IP address for the device** – The LCD display has a **CONNECTION** tab on it. Tap this tab and get the IP address for the AWS Snowcone device.

10. **Use AWS OpsHub to unlock the device** – To unlock the AWS Snowcone device, you enter the IP address of the device, upload your manifest, and the unlock code. AWS OpsHub decrypts the manifest and uses it to authenticate your access to the device. For more information about AWS OpsHub, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

11. **Use the device** – Use AWS OpsHub to set up and manage AWS Snowcone devices so that you can quickly run your edge compute workloads and transfer data to the AWS Cloud. With just a few clicks, you can use AWS OpsHub to unlock and configure your Snowcone device, drag and drop data, launch applications, or monitor device metrics. For details, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

12. **Don't unplug the Ethernet or power supply cables** – Don't unplug the Ethernet cable or the USB-C power supply or battery during data transfer or computing operations. To turn off the Snowcone device after your data transfer or compute job is complete, press the power button.

13. **Prepare the device for its return trip** – After you're done with the device in your on-premises location, and the file interface status changes to **Complete**, press the power button above the LCD display to power off the device. Unplug the device and store its power cables in the cable nook on top of the device, and shut all three of the device's doors. The device is now ready to be returned.

14.**Your Region's carrier returns the device to AWS** – When the carrier has the AWS Snowcone device, the status for the job changes to **In transit to AWS**.

How Import and Export Jobs Work

You can use Snowcone to transfer data online between your device and AWS storage services by using AWS DataSync. You can also transfer data offline from your on-premises storage devices to your Snowcone device.

Online Data Transfer Between Snowcone and AWS Storage Services with DataSync

AWS DataSync is an online data transfer service that simplifies, automates, and accelerates copying large amounts of data to and from AWS storage services over the internet or AWS Direct Connect. An AWS DataSync agent is pre-installed on your Snowcone device and is used to transfer data between the device and Amazon S3 buckets, Amazon Elastic File System (Amazon EFS) file systems, and Amazon FSx for Windows File Server. DataSync automatically handles moving files and objects, scheduling data transfers, monitoring the progress of transfers, encrypting data, verifying data transfers, and notifying customers of any issues.

The DataSync agent is preinstalled on your Snowcone device as an Amazon Machine Image (AMI) during the Snowcone job preparation. To transfer data online to AWS, connect the Snowcone device to the external network and use AWS OpsHub for Snow Family or the AWS Command Line Interface (AWS CLI) to launch the DataSync agent AMI. Activate the DataSync agent using the AWS Management Console or the AWS CLI. Then set up your online data transfer task between the AWS Snowcone Network File System (NFS) store and Amazon S3, Amazon EFS, or Amazon FSx.

You can use DataSync running on Snowcone for the following:

- For edge computing applications, to collect data, process the data to gain immediate insight, and then transfer the data online to AWS.
- To transfer data that is continuously generated by sensors or machines online to AWS in a factory or at other edge locations.
- To distribute media, scientific, or other content online from AWS storage services to your partners and customers.
- To aggregate content by transferring media, scientific, or other content online from your edge locations to AWS.

For one-time edge compute or data transfer workflows or for Snowcone workflows in edge locations without a wide area network (WAN) link or inadequate WAN bandwidth, we recommend shipping the Snowcone device back to AWS to complete the data transfer.

Offline Data Transfer Between Snowcone and Amazon S3

For offline data import jobs, you connect the Snowcone device to your on-premises network and then use AWS OpsHub to unlock the device. Download AWS OpsHub from the AWS Snowball resources page. You can copy data from on-premises storage devices to your Snowcone device through the NFS file interface. After you copy the data to your Snowcone device, the E Ink shipping label on the device helps ensure that the device is automatically sent to the correct AWS facility. You can track the Snowcone device by using Amazon SNS–generated text messages or emails and the console.

Exporting Jobs from Amazon S3

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

You can export up to 50 million objects for each export job, using as many devices as it takes to complete the transfer. If you need to transfer more than 50 million objects, you must create multiple export jobs. Each Snowcone device for an export job's job parts is delivered one after another, with subsequent devices 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 a device, 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.

When AWS receives a returned device, we completely erase it, following National Institute of Standards and Technology (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 object keys.

How Export Jobs Work

Each export job can use any number of Snowcone devices. If the listing contains more data than can fit on a single device, multiple devices are provided to you. Each job part has exactly one device associated with it. After your job parts are created, your first job part enters the **Preparing Snowcone** status.

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

Soon after that, we start exporting your data onto a device. Typically, exporting data takes one business day. However, this process can take longer depending on the amount and type of data. When the export is done, AWS gets the device ready for pickup by your Region's carrier. When it arrives, you connect the Snowcone device to your network and transfer the data that you want to import from Amazon S3 onto the device.

When you're done transferring data, ship the device back to AWS. When we receive the device for your export job part, we erase it completely. This erasure follows NIST 800-88 standards. This step marks the completion of that particular job part.

> **Important**
>
> - For keylisting
>
>   Before we export the objects in the S3 bucket, we scan the bucket. If the bucket is altered after the scan, the job could encounter delays because we scan for missing or altered objects.
>
> - For Amazon S3 Glacier
>
>   Snowcone cannot export objects in the S3 Glacier storage class. These objects must be restored to S3 buckets before Snowcone can successfully export the objects in the bucket.

Using Export Ranges

When you create an export job in the AWS Snow Family Management Console or with the job management API, you can export an entire Amazon S3 bucket or a specific range of object 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 about the specifics of UTF-8, see UTF-8 on Wikipedia.

Export Range Examples

Assume that 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

Specified range beginning	Specified range ending	Objects in the range that will be exported
		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
		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 after *Car* according to UTF-8 binary values.
Aar	Aarrr	Aardvark
		Aardwolf

For Import Job Storage

Internally, a Snowcone device contains 8 TB or 14 TB of disk storage that can be used with the internal Network File System (NFS) service or local Amazon EC2 instances through a local Amazon Elastic Block Store (Amazon EBS) volume presentation. You can use 8 TB or 14 TB for the NFS storage volume and 150 GB for the capacity-optimized HDD (`sbg1`) Amazon EBS storage volume.

For Compute Job Storage

If the job type is local compute, you might create a total of 8 TB or 14 TB of local capacity-optimized HDD (`sbg1`) Amazon EBS volumes and attach them to Amazon EC2 instances. Using Amazon EBS volumes allows the local Amazon EC2 instances to access more local capacity than the root volume alone. Because this is local storage only, data written to the Amazon EBS volumes is lost when the device is returned to AWS because it can't be imported into Amazon S3.

Note
The NFS server is not available for compute jobs. If you need to import or export data to or from the AWS Cloud, don't choose the local compute job type when you place your order.

AWS Snowcone Device Specifications

This section provides information about AWS Snowcone device specifications and requirements for hardware, network, and power supply.

Topics

- Features and Specifications Summary (p. 10)
- Hardware and Network (p. 11)
- Ruggedization Specifications (p. 13)
- AWS Snowcone Power Supply and Accessories (p. 13)

Features and Specifications Summary

The following table summarizes the features and specifications for the Snowcone device.

Item	Specification
Usage scenario	Industrial Internet of Things (IoT), transportation, healthcare IoT, content distribution, tactical edge computing, logistics, autonomous vehicle, data migration
Device size	9 inches long, 6 inches wide, and 3 inches tall (227 mm x 148.6 mm x 82.65 mm)
Device weight	4.5 lbs. (2.1 kg) for Snowcone and 4.0 lbs (1.81 kg) Snowcone SSD
Storage capacity	8 TB usable for Snowcone and 14 TB useable for Snowcone SSD
Onboard computing options	Amazon EC2 Amazon Machine Images (AMIs)
Encryption	Yes, 256-bit
Transfers through Network File System (NFS)	Yes
Transfers through Amazon S3 API	No
Portability	Battery-based operation
Number of usable vCPUs	2 vCPUs
Available memory	4 GB
Network interfaces	2x 1/10 gigabit (Gb) - RJ45
AWS DataSync agent pre-installed	Yes

Item	Specification
Typical job lifetime	Offline or online data transfer: Days to weeks Edge compute: Weeks to months
Max job length	Edge compute or on-going data transfer: Up to 1,460 days

Hardware and Network

A Snowcone device provides 8 TB or 14 TB of available storage. It runs specific Amazon Elastic Compute Cloud (Amazon EC2) instances with two available CPUs and 4 GB of available memory to support your applications and AWS IoT Greengrass functions. In this section, you can find information about the physical device, such as the interfaces, power button, and power requirements as they appear in different views of the device.

Top View

The Snowcone device's top surface includes an integrated E-Ink touch display that is used as an operator interface to set up both wired and wireless networking. It also serves as a display for an electronic shipping label. The electronic shipping label is preprogrammed with both outbound and inbound shipping labels that automatically change after the device is first powered on.
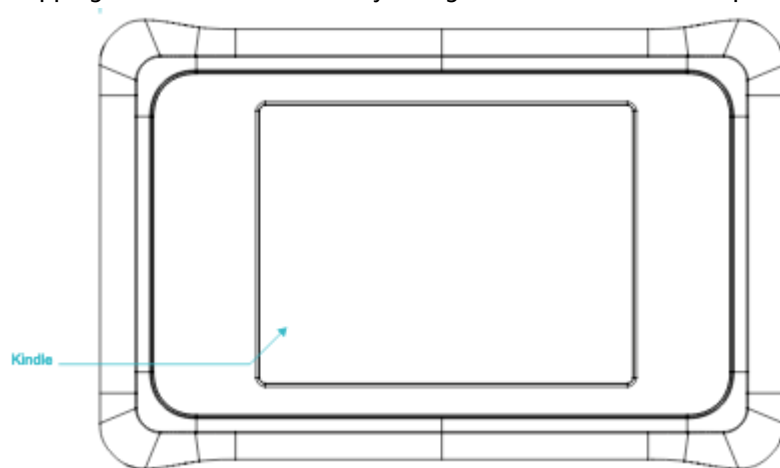
Rear Panel

You make all cable connections on the rear panel. This section describes each connector.
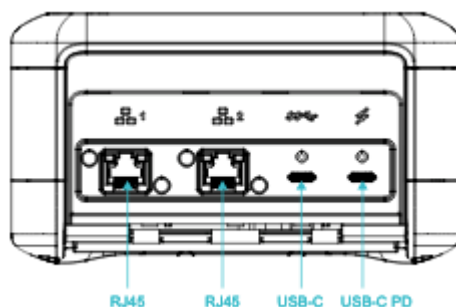
Power

Power is supplied to the device through the rightmost USB-C connection using a suitable power adapter that can supply at least 45 W.

> **Note**
> AWS Snowcone does not include a power supply because it ships with the smallest possible form factor. For details, see AWS Snowcone Power Supply and Accessories (p. 13).

USB

The first USB-C connection is not active.

Ethernet connectors 1 and 2

For wired networking, the Snowcone device provides two ports that auto-negotiate for 1 gigabit (Gb) or 10 Gb Ethernet networks.

RJ45 10/1G Base-T Ethernet ports – These ports auto-negotiate between 10 Gb and 1 Gb based on the far end connection capability. They don't negotiate speeds lower than 1 Gb (for example, 100 Mb or 10 Mb). The link LED is located to the right of each connector, and the activity LED is on the left.
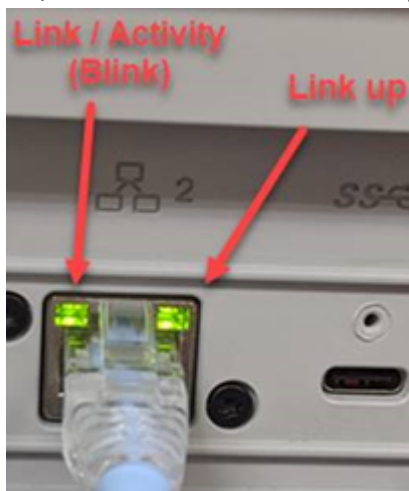
Front Panel

The front panel contains the power button and status LED displays.
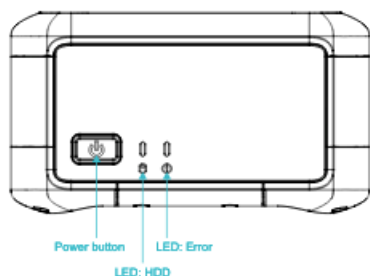
Power switch

To turn on the device, momentarily press the power button. The button illuminates, the E-Ink display changes to a progress bar, and **Please wait** is displayed.

To turn off the device, hold the power switch for two seconds, or until the E-Ink display shows **Please wait**.

> **Note**
> Holding the power button for more than five seconds forces the device to power off. Doing this is not recommended because it might cause data in buffers to be lost. After AC power loss, the device automatically restores power to the last operating state.

Status LEDs

Two status LEDs are located next to the power button. The left LED flashes with disk activity and the right LED illuminates if there is a fault condition.

Ruggedization Specifications

AWS Snowcone devices are designed to meet stringent standards for ruggedization, including ISTA-3A, ASTM D4169, and MIL-STD-810G for free-fall shock, operational vibration, and more. They are designed to tolerate falls up to 3.8 feet (1.15 meters). They also meet the IP65 International Protection Marking IEC standard, meaning they are both dust-tight (allowing no dust inside the enclosure when sealed) and water resistant (including protection from water jets on all sides).

The devices have a wide operating temperature range from freezing (0 degrees C or 32 degrees F) to desert-like conditions (Snowcone: 38 degrees C or 100 degrees F; Snowcone SSD: 45 degrees C or 113 degrees F). When in storage or being shipped, Snowcone devices withstand even harsher temperatures (-32 degrees C or -25.6 degrees F to 63 degrees C or 145.4 degrees F).

AWS Snowcone Power Supply and Accessories

AWS Snowcone devices do not include a power supply or an Ethernet cable (RJ45) because they ship with the smallest possible form factor. You have the option to run your Snowcone device through a plug-in power source or a battery. Here are the details to guide you when ordering a power supply and Ethernet cable:

USB-C power adapter – Use a USB-C power adapter with the Snowcone device for plugged-in power or for stationary (non-mobile) operating environments. To power your Snowcone device, you can purchase one of the following AWS-tested USB-C power adapters:

- Apple 61W USB-C Power Adapter
- Lenovo USB-C 65W Standard AC Adapter

Or, you can use any USB-C power adapter that is rated for 45 W+ and your environment temperature.

USB-C battery – Use a USB-C battery to power the Snowcone device in mobile or portable operating environments. To power your Snowcone device, you can purchase one of the following AWS-tested USB-C batteries:

- Dell Notebook Power Bank Plus – USB C, 65Wh - PW7018LC
- Omni 20c+

Or, you can use any USB-C battery that is rated for 45 W+.

Ethernet cable (RJ45) – To connect the Snowcone device to your local network, use an Ethernet cable (RJ45). If you don't have one, you should purchase one.

Setting Up AWS for AWS Snowcone

Before you create your first AWS Snowcone job, follow these instructions to ensure that you adequately prepare your environment.

Topics

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your account is automatically signed up for all AWS services. AWS only charges you for the services that you use. After you set up your account, you can order, configure, and manage your AWS Snowcone device through the AWS Snow Family Management Console. For more information about pricing and fees for Snowcone, see AWS Snowcone pricing.

If you already have an AWS account, note your AWS account number. If you don't have an AWS account, follow these steps:

To create an AWS account

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

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

Note your AWS account number, which you'll need for the next step.

Before You Order a Snowcone Device

AWS Snowcone is a region-specific service, so make sure that the service is available in your region before you plan the job. Ensure that your location and Amazon S3 bucket are within the same AWS Region because it will impact your ability to order the device. There are limitations on shipping the Snowcone device outside of a region's country borders. For more information, see Region-Based Shipping Restrictions (p. 111).

International shipments to locations outside of your AWS Region are supported on the AWS Snow Family Management Console for whitelisted customers for a select set of locations from specific regions (such as US to Mexico). You should discuss the target destination, costs, and timing to accommodate these requests with your account team.

As part of the order process, you create an AWS Identity and Access Management (IAM) role and AWS Key Management Service (AWS KMS) key. The KMS protects the encryption keys used to protect data on each device. For more information, see Creating an AWS Snowcone Job.

Topics
- Questions About the Local Environment (p. 15)
- Working with Files with Special Characters (p. 15)
- Using Amazon EC2 on Snowcone (p. 16)

Questions About the Local Environment

Understanding your dataset and how the local environment is set up will help you complete your data transfer. Consider the following before placing your order.

Will the data be accessed during the transfer?

We recommend that if you are using the file interface, you use only one method of transferring data to the AWS Snowcone. To prevent corrupting your data, don't disconnect a Snowcone device or change its network settings while transferring data. Files should be in a static state while being written to the device. Files that are modified while they are being written to the device can result in read/write conflicts.

Working with Files with Special Characters

It is important to note that if your files contain special characters, you might encounter errors. Although Amazon S3 allows special characters, we highly recommend that you avoid the following characters:

- Backslash ("\")
- Left curly brace ("{")
- Right curly brace ("}")
- Left square bracket ("[")
- Right square bracket ("]")
- 'Less than' symbol ("<")
- 'Greater than' symbol (">")
- Non-printable ASCII characters (128–255 decimal characters)
- Caret ("^")
- Percent character ("%")

- Grave accent / back tick ("`")
- Quotation marks
- Tilde ("~")
- 'Pound' character ("#")
- Vertical bar / pipe ("|")

If your files have one or more of these characters, rename them before you copy them to the AWS Snowcone device. Windows users who have spaces in their file names should be careful when copying individual objects or running a recursive command. Surround individual objects that contain spacing in the name with quotation marks. The following are examples of such files.

Operating System	File Name: `test file.txt`
Windows	"C:\Users\<username>\desktop\test file.txt"
Mac	/Users/<username>/test\ file.txt
Linux	/home/<username>/test\ file.txt

Note
The only object metadata that is transferred is the object name and size. If you want additional metadata to be copied, you can use the file interface or other tools to copy the data to Amazon S3. However, AWS DataSync preserves access control lists (ACLs). For information, see How DataSync Handles Metadata and Special Files in the *AWS DataSync User Guide*.

Using Amazon EC2 on Snowcone

This section provides an overview of using Amazon EC2 compute instances on an AWS Snowcone device.

You should use the Amazon EC2 compatible instances when you have an application running on the edge that is managed and deployed as a virtual machine (an Amazon Machine Image, or AMI). Snowcone supports the SNC1 instance type with three instances, including snc1.micro (1 CPU and 1 GB RAM), snc1.small (1 CPU and 2 GB RAM), and snc1.medium (2 CPU and 4 GB RAM). The support for EC2-compatible instances on Snowcone enables you to build and test your application on Amazon EC2. You can enable and provision EC2 AMIs during AWS Snowcone job creation using either the AWS Management Console, AWS Snowball SDK, or AWS CLI.

Supported Amazon EC2 instance types

Use the following Amazon EC2 instance types for your compute jobs.

snc1.micro—1 CPU core, 1 GB RAM

snc1.small—1 CPU core, 2 GB RAM

snc1.medium—2 CPU cores, 4 GB RAM

Use AWS OpsHub to manage your instances on Snowcone. Download AWS OpsHub from the AWS Snowball resources website. After you unlock the device using AWS OpsHub, navigate to the Amazon EC2 page. Choose **Create instance** to create an EC2 instance based on the AMI that you had preloaded onto the device when you created the job. You can then connect to the instances and run your edge application. AWS OpsHub also provides single-click buttons to start, stop, terminate, and reboot your EC2 instances. For more information, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

When you're done with your device, return it to AWS. If the device was used in an import job, the data transferred using the file interface is imported into Amazon S3 by using the Snowcone NFS interface. Otherwise, we perform a complete erasure of the device when it is returned to AWS. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

> **Important**
> Data in compute instances running on a Snowcone isn't imported into AWS.

Pricing for Compute Instances on Snowcone

There are additional costs associated with using compute instances. For more information, see AWS Snowcone pricing.

Prerequisites

Before creating your job, keep the following information in mind:

- Before you can add any AMIs to your job, you must have an AMI in your AWS account and it must be a supported image type. Currently, supported AMIs are based on the Amazon Linux 2, CentOS 7 (x86_64) - with Updates HVM, or Ubuntu 16.04 LTS - Xenial (HVM) images. You can get these images from the AWS Marketplace.
- All AMIs must be based on Amazon Elastic Block Store (Amazon EBS), with a single volume.
- If you are connecting to a compute instance running on a Snowcone, you must use Secure Shell (SSH). To do so, you first add the key pair.

Creating a Linux AMI from an Instance

You can create an AMI using the console or the command line. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally, launch an instance of your new AMI.

To create an AMI from an instance using the console

1. Choose an appropriate EBS-backed AMI as a starting point for your new AMI, and configure it as needed before launch. For more information, see Launching an instance using the Launch Instance Wizard.
2. Choose **Launch** to launch an instance of the EBS-backed AMI that you've selected. Accept the default values as you step through the wizard. For more information, see Launching an instance using the Launch Instance Wizard.
3. While the instance is running, connect to it. You can perform the following actions on your instance to customize it for your needs:

   - Install software and applications
   - Copy data
   - Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space
   - Attach additional Amazon EBS volumes

4. (Optional) Create snapshots of all the volumes attached to your instance. For more information about creating snapshots, see Creating Amazon EBS snapshots.
5. In the navigation pane, choose **Instances**, and select your instance. For **Actions**, choose **Image**, **Create Image**

   > **Tip**
   > If this option is disabled, your instance isn't an Amazon EBS-backed instance.

6. In the **Create Image** dialog box, specify the following information, and then choose **Create Image.**

- **Image name** - A unique name for the image.
- **Image description** - An optional description of the image, up to 255 characters.
- **No reboot** - This option is not selected by default. Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Choose **No reboot** to avoid having your instance shut down.

    > **Warning**
    > If you choose **No reboot**, we can't guarantee the file system integrity of the created image.

- **Instance Volumes** - The fields in this section enable you to modify the root volume, and add other Amazon EBS and instance store volumes. For information about each field, pause on the **i** icon next to each field to display field tooltips. Some important points are listed following.

    - To change the size of the root volume, locate **Root** in the **Volume Type** column. For **Size (GiB)**, enter the required value.
    - If you select **Delete on Termination**, when you terminate the instance created from this AMI, the Amazon EBS volume is deleted. If you clear **Delete on Termination**, when you terminate the instance, the Amazon EBS volume is not deleted. For more information, see Preserving Amazon EBS volumes on instance termination in the *Amazon EC2 User Guide for Linux Instances*.
    - To add an Amazon EBS volume, choose **Add New Volume** (which adds a new row). For **Volume Type**, choose **EBS**, and fill in the fields in the row. When you launch an instance from your new AMI, additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
    - To add an instance store volume, see Adding Instance Store Volumes to an AMI. When you launch an instance from your new AMI, additional volumes are automatically initialized and mounted. These volumes do not contain data from the instance store volumes of the running instance on which you based your AMI.

7. To view the status of your AMI while it is being created, in the navigation pane, choose **AMIs**. Initially, the status is pending but should change to available after a few minutes.

   (Optional) To view the snapshot that was created for the new AMI, choose **Snapshots**. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

8. Launch an instance from your new AMI. For more information, see Launching an instance using the Launch Instance Wizard.

9. The new running instance contains all of the customizations that you applied in previous steps.

To Create an AMI from an Instance Using the Command Line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2.

- create-image (AWS CLI)
- New-EC2Image (AWS Tools for Windows PowerShell)

Creating a Linux AMI from a Snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot using the AWS Management Console or the command line.

To create an AMI from a snapshot using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots**.

3. Choose the snapshot and choose **Actions**, **Create Image**.

4. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, and then choose **Create**. If you're re-creating a parent instance, choose the same options as the parent instance.

   - **Architecture:** Choose **i386** for 32-bit or **x86_64** for 64-bit.
   - **Root device name:** Enter the appropriate name for the root volume. For more information, see Device naming on Linux instances.
   - **Virtualization type:** Choose whether instances launched from this AMI use paravirtual (PV) or hardware virtual machine (HVM) virtualization. For more information, see Linux AMI Virtualization Types.
   - (PV virtualization type only) **Kernel ID** and **RAM disk ID:** Choose the AKI and ARI from the lists. If you choose the default AKI or don't choose an AKI, you must specify an AKI every time you launch an instance using this AMI. In addition, your instance might fail the health checks if the default AKI is incompatible with the instance.
   - (Optional) **Block Device Mappings:** Add volumes or expand the default size of the root volume for the AMI. For more information about resizing the file system on your instance for a larger volume, see Extending a Linux file system after resizing a volume.

To Create an AMI from a Snapshot Using the Command Line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2.

- register-image (AWS CLI)
- Register-EC2Image (AWS Tools for Windows PowerShell)

Getting Started

This section provides general instructions for creating and completing your first AWS Snowcone job in the AWS Snow Family Management Console. For an overview of the AWS Snowcone device, see How AWS Snowcone Works (p. 3).

This getting started documentation assumes that you use the AWS Snow Family Management Console to create your job, and you use the Snowball Edge client or the AWS OpsHub for Snow Family application to unlock the AWS Snowcone device. 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 Snowcone 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). For more information, see Setting Up AWS for AWS Snowcone (p. 14).

To get started with AWS Snowcone, see Creating an AWS Snowcone Job.

Creating an AWS Snowcone Job

Regardless of the type of job that you're creating for Snow Family Devices, there is a set of common tasks that you perform in the AWS Snow Family Management Console. These tasks include the following:

- Creating a job
- Monitoring the status of your job
- Setting up the device
- Unlocking the device
- Transferring data
- Returning the device

The following sections provide step-by-step instructions for performing these tasks. When there is a job type-specific consideration, it is called out specifically in a note.

You can use the AWS Snow Family Management Console to create and manage jobs. Depending on your use case, and regardless of your job type, the process for creating a new job in the AWS Snow Family Management Console follows a standard workflow.

You can also create and manage jobs using the job management API. For more information, see the *AWS Snowball API Reference*.

Topics

Step 1: Plan Your Job

The first step in creating a job is to determine the type of job that you need and to start planning it using the AWS Snow Family Management Console.

To plan your job

1. Sign in to the AWS Management Console, and open the AWS Snow Family Management Console. If this is your first time creating a job in this AWS Region, you will see the **AWS Snow Family** page. Otherwise you will see the **Plan your job** page.

2. On the **AWS Snow Family** page, choose **Order an AWS Snow family device** to open the **Plan your job** page.

3. Depending on your use case, choose one of the following job types:

   - **Import into Amazon S3** – Choose this option to have AWS ship an empty Snowball Edge device to you. You connect the device to your local network and run the Snowball Edge client. You copy data onto the device, ship it back to AWS, and your data is uploaded to the AWS Cloud.

   - **Local compute and storage only** – Choose this option to perform compute and storage workloads on the device without transferring any data.

4. Choose **Next** to continue.

Step 2: Choose Your Shipping Preferences

Receiving and returning a Snow Family device involves shipping the device back and forth, so it's important that you provide accurate shipping information.

To provide shipping details

1. On the AWS Snow Family Management Console, in the **Provide shipping information** section, choose an existing address or add a new address.

   - If you choose **Use recent address**, the addresses on file are displayed. Carefully choose the address that you want from the list.

   - If you choose **Add a new address**, provide the requested address information. The AWS Snow Family Management Console saves your new shipping information.

     Note
     The country that you provide in the address must match the destination country for the device and must be valid for that country.

2. In the **Shipping speed** section, choose a shipping speed for the job. This speed shows how quickly the device ships between destinations and doesn't reflect how soon it will arrive after today's date. The shipping speeds you can choose are:

   - **One-Day Shipping (1 business day)**
   - **Two-Day Shipping (2 business days)**

3. Choose **Next**.
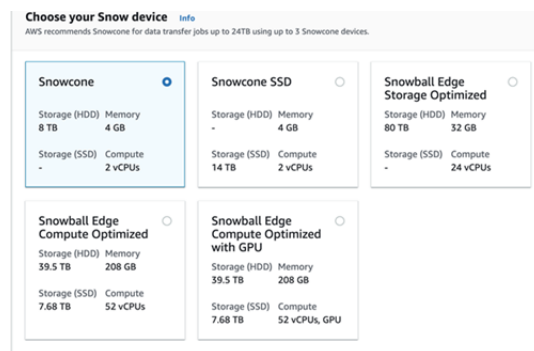
Step 3: Choose Your Job Details

In this step, you provide details for your AWS Snow Family device job, including the job name, AWS Region, device type, Amazon S3 bucket name, and Amazon Machine Image (AMI).

To add job details

1. On the AWS Snow Family Management Console, in the **Name your job** section, provide a name for your job in the **Job name** box.

2. In the **Choose your Snow device** section, choose the AWS Snowcone device type that you want to use.

   AWS Snowcone device options

   - **Snowcone** – 8 TB storage (HDD), 4 GB memory, 2 vCPUs
   - **Snowcone SSD** – 14 TB storage (SSD), 4 GB memory, 2 vCPUs

   Choose your Snow device Info
   AWS recommends Snowcone for data transfer jobs up to 24TB using up to 3 Snowcone devices.

Snowcone ●	Snowcone SSD ○	Snowball Edge Storage Optimized ○
Storage (HDD) Memory 8 TB 4 GB	Storage (HDD) Memory - 4 GB	Storage (HDD) Memory 80 TB 32 GB
Storage (SSD) Compute - 2 vCPUs	Storage (SSD) Compute 14 TB 2 vCPUs	Storage (SSD) Compute - 24 vCPUs

Snowball Edge Compute Optimized ○	Snowball Edge Compute Optimized with GPU ○
Storage (HDD) Memory 39.5 TB 208 GB	Storage (HDD) Memory 39.5 TB 208 GB
Storage (SSD) Compute 7.68 TB 52 vCPUs	Storage (SSD) Compute 7.68 TB 52 vCPUs, GPU

3. In the **Snowcone power supply** section, choose **I will provide my own power supply and Ethernet cable**. For information about power supplies, see AWS Snowcone Power Supply and Accessories.

4. In the **Choose your data transfer mechanism** section, choose either **S3** or **NFS**.

5. Depending on the transfer mechanism that you chose, do one of the following:

   - In the **Choose your S3 storage** section, choose to create a new Amazon S3 bucket, or choose the S3 bucket that you want to use in the **Bucket name** list. You can include additional S3 buckets. These buckets appear on your device as local S3 buckets.

   - In the **Choose your NFS storage** section, choose to create a new S3 bucket, or choose the S3 bucket that you want to use in the **Bucket name** list. You can include additional S3 buckets. These buckets appear on your device as Network File System (NFS) mount points.

6. In the **Compute using EC2 instances - *optional*** section, choose **EC2 AMI name**. This option enables you to use compute Amazon EC2 instances in a cluster. It loads your Snowball Edge device with Amazon EC2 AMIs, and enables your device to function as a mobile data center.

   For more information, see Using Amazon EC2 Compute Instances.

   This feature incurs additional charges. For more information, see AWS Snowball Edge Pricing.

7. Choose an AMI in the **Source AMI name** list. Or, search for an AMI in the **Source AMI name** box and choose **Next**.

Step 4: Choose Your Security Preferences

Setting security adds the permissions and encryption settings for your AWS Snow Family Devices job to help protect your data while in transit.

To set security for your job

1. In the **Encryption** section, choose the **KMS key** that you want to use.

- If you want to use the default AWS Key Management Service (AWS KMS) key, choose **aws/importexport (default)**. This is the default key that protects your import and export jobs when no other key is defined.

- If you want to provide your own AWS KMS key, choose **Enter a key ARN**, provide the Amazon Resource Name (ARN) in the **key ARN** box, and choose **Use this KMS key**. The key ARN will be added to the list.

2. In the **Service access** section, choose **Create service role** to grant AWS Snow Family permissions to use Amazon S3 and Amazon Simple Notification Service (Amazon SNS) on your behalf.

3. Choose **View details** to choose the **IAM role** that you want, or you can use the default role.

4. For **Policy name**, choose the import policy that you want to use.

Example Example policies for Snowcone devices

Import-only role policy example

The following is an example of an S3 import-only role policy.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetBucketPolicy",
                    "s3:GetBucketLocation",
                    "s3:ListBucketMultipartUploads",
                    "s3:ListBucket",
                    "s3:HeadBucket",
                    "s3:PutObject",
                    "s3:AbortMultipartUpload",
                    "s3:ListMultipartUploadParts",
                    "s3:PutObjectAcl",
                    "s3:GetObject"
                ],
                "Resource": [
                    "arn:aws:s3:::YourS3BucketARN",
                    "arn:aws:s3:::YourS3BucketARN/*"
                 ]
            }
        ]
    }
```

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

Using server-side encryption to encrypt the Amazon S3 bucket

If you use server-side encryption with AWS KMS–managed keys (SSE-KMS) to encrypt the Amazon S3 buckets associated with your import job, you also must add the following statement to your IAM role.

```
{
    "Effect": "Allow",
     "Action": [
       "kms:GenerateDataKey","kms:Decrypt"

     ],
     "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-
efgh-111111111111"
}
```

5. Choose **Allow**.

6. Choose **Next**.

Step 5: Choose Your Notification Preferences

Notifications update you on the latest status of your AWS Snow Family Devices jobs. You create an SNS topic and receive emails from Amazon Simple Notification Service (Amazon SNS) as your job status changes.

To set up notifications

1. In the **Set notifications** section, do one of the following:
   - If you want to use an existing SNS topic, choose **Use an existing SNS topic**, and choose the topic Amazon Resource Name (ARN) from the list.
   - If you want to create a new SNS topic, choose **Create a new SNS topic**. Enter a name for your topic and provide an email address.

2. Choose **Next**.

The notification will be about one of the following states of your job:

- Job created
- Preparing device
- Preparing shipment
- In transit to you
- Delivered to you
- In transit to AWS
- At sorting facility
- At AWS
- Importing
- Completed
- Canceled

Step 6: Download AWS OpsHub

The AWS Snow Family Devices offer a user-friendly tool, AWS OpsHub for Snow Family, that you can use to manage your devices and local AWS services.

With AWS OpsHub installed on your client computer, you can perform tasks such as the following:

- Unlocking and configuring single or clustered devices
- Transferring files
- Launching and managing instances running on Snow Family Devices.

For more information, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

To download and install AWS OpsHub for Snow Family

1. On the AWS Snow Family Management Console, in the **Download AWS OpsHub** section, choose **Get AWS OpsHub**. You are redirected to the AWS Snowball resources website.
2. In the **AWS OpsHub** section, choose **Download** for your operating system, and follow the installation steps. When you are finished, choose **Next**.

Step 7: Review and Create Your Job

After you provide all the necessary job details for your AWS Snow Family Devices job, review the job and create it.

1. On the AWS Snow Family Management Console, in the **Review job order** page, review all the sections before you create the job. If you want to make changes, choose **Edit** for the appropriate section, and edit the information.
2. When you are done reviewing and editing, choose **Create job**. After you create a job, you can cancel it within an hour without incurring any charges.

   Jobs are subject to export control laws in specific countries and might require an export license. US export and re-export laws also apply. Diversion from the country and US laws and regulations is prohibited.

   > **Note**
   > Snow Family Devices jobs are not fulfilled with a power supply, and one must be provided separately.

After your job is created, you can see the status of the job in the **Job status** section. For detailed information about job statuses, see Job Statuses.

Using AWS OpsHub for Snow Family to Manage Devices

The Snow Family Devices now offer a user-friendly tool, AWS OpsHub for Snow Family, that you can use to manage your devices and local AWS services. You use AWS OpsHub on a client computer to perform tasks such as unlocking and configuring single or clustered devices, transferring files, and launching and managing instances running on Snow Family Devices. You can use AWS OpsHub to manage both the Storage Optimized and Compute Optimized device types and the Snow device. The AWS OpsHub application is available at no additional cost to you.

AWS OpsHub takes all the existing operations available in the Snowball API and presents them as a graphical user interface. This interface helps you quickly migrate data to the AWS Cloud and deploy edge computing applications on Snow Family Devices.

AWS OpsHub provides a unified view of the AWS services that are running on Snow Family Devices and automates operational tasks through AWS Systems Manager. With AWS OpsHub, users with different levels of technical expertise can manage a large number of Snow Family Devices. With a few clicks, you can unlock devices, transfer files, manage Amazon EC2 instances, and monitor device metrics.

When your Snow device arrives at your site, you download, install, and launch the AWS OpsHub application on a client machine, such as a laptop. After installation, you can unlock the device and start managing it and using supported AWS services locally. AWS OpsHub provides a dashboard that summarizes key metrics such as storage capacity and active instances on your device. It also provides a selection of AWS services that are supported on the Snow Family Devices. Within minutes, you can begin transferring files to the device.

After you download the AWS OpsHub application and install it on a client machine, AWS OpsHub can connect to the AWS Snowcone device on the same network, whether the device is connected via Wi-Fi or a physical cable. Then you open AWS OpsHub and unlock the device. You are then presented with a dashboard that shows your device and its system metrics. You can then begin deploying your edge applications or migrating your data to the device. AWS OpsHub makes data transfers to your Snowcone device simple by allowing you to drag-and-drop files or folders onto the device. With AWS OpsHub, you can also easily see what is stored on the device.

Topics

Unlocking a device

When your device arrives at your site, the first step is to connect and unlock it. AWS OpsHub lets you sign in, unlock, and manage devices using the following methods:

- **Locally** – To sign in to a device locally, you must power on the device and connect it to your local network. Then provide an unlock code and a manifest file.
- **Remotely** – To sign in to a device remotely, you must power on the device and make sure that it can connect to `device-order-region`.amazonaws.com through your network. Then provide the AWS

Identity and Access Management (IAM) credentials (access key and secret key) for the AWS account that is linked to your device.

For information on enabling remote management and creating an associated account, see Enabling Snow Device Management on Snowcone.

Topics

Unlocking a device locally

To connect and unlock your device locally

1. Open the flap on your device, locate the power cord, and connect it to a power source.
2. Connect the device to your network using an Ethernet cable (typically an RJ45 cable), then open the front panel and power on the device.
3. Open the AWS OpsHub application. If you are a first-time user, you are prompted to choose a language. Then choose **Next**.
4. On the **Get started with OpsHub** page, choose **Sign in to local devices**, and then choose **Sign in**.
5. On the **Sign in to local devices** page, choose **Snowcone**, and then choose **Sign in**.

    If you don't have a device, you can order one. For information about how to order a device, see Getting Started (p. 20).
6. On the **Sign in to your Snowcone** page, enter the **Device IP address** and **Unlock code**. To select the device manifest, choose **Choose file**, and then choose **Sign in**.
7. (Optional) Save your device's credentials as a *profile*. Name the profile and choose **Save profile name**. For more information about profiles, see Managing Profiles (p. 40).
8. On the **Local devices** tab, choose a device to see its details, such as the network interfaces and AWS services that are running on the device. You can also see details for clusters from this tab, or manage your devices just as you do with the AWS Command Line Interface (AWS CLI). For more information, see Managing AWS services on your device (p. 30).

    For devices that have AWS Snow Device Management installed, you can choose **Enable remote management** to turn on the feature. For more information, see Using AWS Snow Device Management to Manage Devices (p. 58).

Unlocking a device remotely

To connect and unlock your device remotely

1. Open the flap on your device, locate the power cord, and connect it to a power source.
2. Connect the device to your network using an Ethernet cable (typically an RJ45 cable), then open the front panel and power on the device.

    Note
    To be unlocked remotely, your device must be able to connect to `device-order-region`.amazonaws.com.
3. Open the AWS OpsHub application. If you are a first-time user, you are prompted to choose a language. Then choose **Next**.
4. On the **Get started with OpsHub** page, choose **Sign in to remote devices**, and then choose **Sign in**.

5. On the **Sign in to remote devices** page, enter the AWS Identity and Access Management (IAM) credentials (access key and secret key) for the AWS account that is linked to your device, and then choose **Sign in**.

6. On the **Remote devices** tab, choose your device to see its details, such as its state and network interfaces. Then choose **Unlock** to unlock the device.

    From the remote device's details page, you can also reboot your devices and manage them just as you do with the AWS Command Line Interface (AWS CLI). To view remote devices in different AWS Regions, choose the current Region on the navigation bar, and then choose the Region that you want to view. For more information, see Managing AWS services on your device (p. 30).

Verifying the signature of AWS OpsHub (optional)

You can install the AWS OpsHub for Snow Family application on a Linux client machine. The AWS OpsHub application installer packages for Linux are cryptographically signed. You can use a public key to verify that the installer package is original and unmodified. If the files are damaged or altered, the verification fails. You can verify the signature of the installer package using GNU Privacy Guard (GPG). This verification is optional. If you choose to verify the signature of the application, you can do it at any time.

You can download the SIGNATURE file for the Linux installer from AWS Snowcone Resources or Snowball Edge Resources.

To verify the AWS OpsHub package on a Linux client machine

1. Copy the following public key, save it to a file, and name the file—for example, `opshub-public-key.pgp`.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
xsFNBF/hGf8BEAC9HCDV8uljDX02Jxspi6kmPu4xqf4ZZLQsSqJcHU61oL/c
/zAN+mUqJT9aJ1rr0QFGVD1bMogecUPflTWlDkEEpG8ZbX5P8vR+EElO/rW/
WtqizSudy6qy59ZRK+YVSDx7DZyuJmIO7j00UADCL+95ZQN9vqwHNjBHsgfQ
l/1Tqhy81ozTZXcI/+u+99YLaugJIP6ZYIeDfpxnghqyVtaappBFTAyfG67Y
N/5mea1VqJzd8liFpIFQnl+X7U2x6emDbM01yJWV3aMmPwhtQ7iBdt5a4x82
EF5bZJ8HSRMvANDILD/9VTN8VfUQGKFjFY2GdX9ERwvfTb47bbv9Z28Vl284
4lw2w1Bl007FoO2v/Y0ukrN3VHCpmJQS1IiqZbYRa0DVK6UR5QNvUlj5fwWs
4qW9UDPhT/HDuaMrMFCejEn/7wvRUrGVtzCT9F56Al/dwRSxBejQQEb1AC8j
uuyi7gJaPdyNntROEFTD7iO2L6X2jB4YLfvGxP7Xeq1Y37t8NKF8CYTpOry/
Wvw0iKZFbo4AkiI0aLyBCk9HBXhUKa9x06gOnhh1UFQrPGrk60RPQKqL76HA
E2ewzGDa90wlRBUAt2nRQpyNYjoASBvz/cAr3e0nuWsIzopZIenrxI5ffcjY
f6UWA/OK3ITHtYHewVhseDyEqTQ4MUIWQS4NAwARAQABzTlBV1MgT3BzSHVi
IGZvciBTbm93IEZhbWlseSA8YXdzLW9wc2h1Yi1zaWduZXXZJAYW1hem9uLmNv
bT7CwY0EEAEIACAFAl/hGf8GCwkHCAMCBBUICgIEFgIBAAIZAQIbAwIeAQAh
CRAhgc9adPNF8RYhBDcvpelIaY930bOvqiGBz1p080XxGbcP+gPZX7LzKc1Y
w9CT3UHgkAIawOSXYktujzoYVxAz8/j3jEkCY0dKnfyqvWZDiJAXnzmxWWbg
cxg1g0GXNXCM4lAd68CmbAOLoLTaWSQX30ZbswzhbtX2ADAlopV8RLBik7fm
bS9FyuubDRhfYRQq0fpjUGXFiEgwg6aMFxsrGLlv4QD7t+6ftFIe/mxLbjR4
iMgtr8FIPXbgn05YYY/LeF4NIgX4iLEqRbAnfWjPzqQ1spFWAotIzDmZqby+
WdWThrH4K1rwtYM8sDhqRnMnqJrGFZzk7aDhVPwF+FOVMmPeEN5JRazEeUrl
VZaSw6mu0n4FMGSXuwGgdvmkqnMe6I5/xLdU4IOPNhp0UmakDWOq/a1dREDE
ZLMQDMINphmeQno4inGmwbRo63gitD4ZNR5sWwfuwty25lo8Ekv7jkkp3mSv
pdxn5tptttnPaSPcSIX/4EDl19Tu0i7aup+v30t7eikYDSZG6g9+jHB3Va9e
/VWShFSgy8Jm2+qq/ujUQDAGTCfSuY9jg1ITsog6ayEZa/2upDJ1m+4OHK4p
8DrEzP/3jTahT8q5ofFWSRDL17d3lTSU+JBmPE3mz311FNXgiO8w+taY320z
+irHtb3iSiiukbjS8s0maVgzszRqS9mhaEn4LL0zoqrUicmXgTyFB7n2LuYv
O7vxMO5xxhGQwsF2BBABCAAJBQJf4RoCAhsDACEJEBFZvzT/tDi5FiEEi+O9
V+UAYN9Gnw36EVm/NP+0OLnnEQ/+J4C0Mn8j0AebXrwBiFs83sQo2q+WHL1S
MRc1g5gRFDXs6h1Gv+TGXRen7j1oeaddWvgOtUBxqmCOjr+8AKH0OtiBWSuO
lsS8JU5rindEsKUrKTwcG2wyZFoe1zlE8xPkLRSRN5ZbbgKsTz16l1HgCCId
```

```
Do+WJdDkWGWxmtDvzjM32EI/PVBd108ga9aPwXdhLwOdKAjZ4JrJXLUQJjRI
IVDSyMObEHOUM6a/+mWNZazNfo0LsGWqGVa6Xn5WJWlwR1S78vPNfO3BQYuO
YRjaVQR+kPtB9aSAZNi5sWfk6NrRNd1Q78d067uhhejsjRt7Mja2fEL4Kb1X
nK4U/ps7XlO3o/VjblneZOhJK6kAKU172tnPJTJ31JbOxX73wsMWDYZRZVcK
9X9+GFrpwhKHWKKPjpMOt/FRxNepvqRl72TkgBPqGH2TMOFdB1f/uQprvqge
PBbS0JrmBIH9/anIqgtMdtcNQB/0erLdCDqI5afOuD1OLcLwdJwG9/bSrfwT
TVEE3WbXmJ8pZgMzlHUiZE6V2DSadV/YItk50IOjjrOVHOHvlFMwGCEAIFzf
9P/pNi8hpEmlRphRiOVVcdQ30bH0M0gPHu5V9flIhyCL1zU3LjYTHkq0yJD5
YDA1xO1MYq3DcSM513OVBbLmuVS2GpcsTCYqlgQA6h/zzMwz+/7OwU0EX+EZ
/wEQAOAY8ULmcJIQWIr14V0jylpJeD3qwj7wd+QsBzJ+mOpOB/3ZFAhQiNOl
9yCDlHeiZeAmWYX9OIXrNiIdcHy+WTAp4G+NaMpqE52qhbDjz+IbvLpl1yDH
bYEHPjnTHXEy2lbvKAJOKkw/2RcQOi4dodGnq5icyYj+9gcuHvnVwbrQ96Ia
0D7c+b5T+bzFqk90nIcztrMRuhDLJnJpi7OjpvQwfq/TkkZA+mzupxfSkq/Y
N9qXNEToT/VI2gn/LS0X4Ar1l2KxBjzNEsQkwGSiWSYtMA5J+Tj5ED0uZ/qe
omNblAlD4bm7Na8NAoLxCtAiDq/f3To9Xb18lHsndOmfLCb/BVgP4edQKTIi
C/OZHy9QJlfmN0aq7JVLQAuvQNEL88RKW6YZBqkPd3P6zdc7sWDLTMXMOd3I
e6NUvU7pW0E9NyRfUF+oT4s9wAJhAodinAi8Zi9rEfhK1VCJ76j7bcQqYZe0
jXD3IJ7T+X2XA8M/BmypwMW0Soljzhwh044RAasr/fAzpKNPB318JwcQunIz
u2N3CeJ+zrsomjcPxzehwsSVq1lzaL2ureJBLOKkBgYxUJYXpbS01ax1TsFG
09ldANOs9Ej8CND37GsNnuygjOgWXbX6MNgbvPs3H3zi/AbMunQ1VBlw07JX
zdM1hBQZh6w+NeiEsK1T6wHi7IhxABEBAAHCwXYEGAEIAAkFAl/hGf8CGwwA
IQkQIYHPWnTzRfEWIQQ3L6XpSGmPd9Gzr6ohgc9adPNF8TMBD/9TbU/+PVbF
ywKvwi3GLOlpY7BXn8lQaHyunMGuavmO8OfaRROynkH0ZqLHCp6bIajFOfvF
b7c0Jamzx8Hg+SIdl6yRpRY+fA4RQ6PNnnmT93ZgWW3EbjPyJGlm0/rt03SR
+0yn4/ldlg2KfBX4pqMoPCMKUdWxGrmDETXsGihwZ0gmCZqXe8lK122PYkSN
JQQ+LlfjKvCaxfPKEjXYTbIbfyyhCR6NzAOVZxCrzSz2xDrYWp/V002Klxda
0ix6r2aEHf+xYEUhOaBt8OHY5nXTuRReCVU789MUVtCMqD2u6amdo4BR0kWA
QNg4yavKwV+LVtyYh2Iju9VSyv4xL1Q4xKHvcAUrSH73bHG7b7jkUJckD0f4
twhjJk/Lfwe6RdnVo2WoeTvE93w+NAq2FXmvbiG7eltl0XfQecvQU3QNbRvH
U8B96W0w8UXJdvTKg4f0NbjSw7iJ3x5naixQ+rA8hLV8xOgn2LX6wvxT/SEu
mn20KX+fPtJELK7v/NheFLX1jsKLXYo4jHrkfIXNsNUhg/x2E71kAjbeT3s+
t9kCtxt2iXDDZvpIbmGO4QkvLFvoROaSmN6+8fupe3e+e2yN0e6xGTuE60gX
I2+X1p1g9IduDYTpoI2OXleHyyMqGEeIb4gOiiSloTp5oi3EuAYRGflXuqAT
VA19bKnpkBsJ0A==
=tD2T
-----END PGP PUBLIC KEY BLOCK-----
```

2. Import the public key into your keyring, and note the returned key value.

   GPG

   ```
   gpg --import opshub-public-key.pgp
   ```

   Example output

   ```
   gpg: key 1655BBDE2B770256: public key "AWS OpsHub for Snow Family <aws-opshub-
   signer@amazon.com>" imported
   gpg: Total number processed: 1
   gpg:               imported: 1
   ```

3. Verify the fingerprint. Be sure to replace *key-value* with the value from the preceding step. We recommend that you use GPG to verify the fingerprint.

   ```
   gpg --fingerprint key-value
   ```

   This command returns output similar to the following.

   ```
   pub   rsa4096 2020-12-21 [SC]
         372F A5E9 4869 8F77 D1B3  AFAA 2181 CF5A 74F3 45F1
   uid           [ unknown] AWS OpsHub for Snow Family <aws-opshub-signer@amazon.com>
   sub   rsa4096 2020-12-21 [E]
   ```

The fingerprint should match the following:

```
372F A5E9 4869 8F77 D1B3 AFAA 2181 CF5A 74F3 45F1
```

If the fingerprint doesn't match, don't install the AWS OpsHub application. Contact AWS Support.

4. Verify the installer package, and download the SIGNATURE file according to your instance's architecture and operating system if you haven't already done so.

5. Verify the installer package signature. Be sure to replace *signature-filename* and *OpsHub-download-filename* with the values that you specified when downloading the SIGNATURE file and AWS OpsHub application.

GPG

```
gpg --verify signature-filename OpsHub-download-filename
```

This command returns output similar to the following.

GPG

```
gpg: Signature made Mon Dec 21 13:44:47 2020 PST
gpg:                  using RSA key 1655BBDE2B770256
gpg: Good signature from "AWS OpsHub for Snow Family <aws-opshub-
signer@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9C93 4C3B 61F8 C434 9F94  5CA0 1655 BBDE 2B77 0256
```

When using GPG, if the output includes the phrase `BAD signature`, check whether you performed the procedure correctly. If you continue to get this response, contact AWS Support and don't install the agent. The warning message about the trust doesn't mean that the signature is not valid, only that you have not verified the public key. A key is trusted only if you or someone who you trust has signed it.

Managing AWS services on your device

With AWS OpsHub, you can use and manage AWS services on your Snow Family Devices. Currently, AWS OpsHub supports the following resources:

- Amazon Elastic Compute Cloud (Amazon EC2) instances – Use Amazon EC2 instances to run software installed on a virtual server without sending it to the AWS Cloud for processing.
- AWS DataSync—Transfer a large number of files between your on-premises storage and other AWS Cloud locations, such as file systems or Amazon S3.
- Network File System (NFS) – Use file shares to move data to your device. You can ship the device to AWS to transfer your data to the AWS Cloud, or use DataSync to transfer to other AWS Cloud locations.
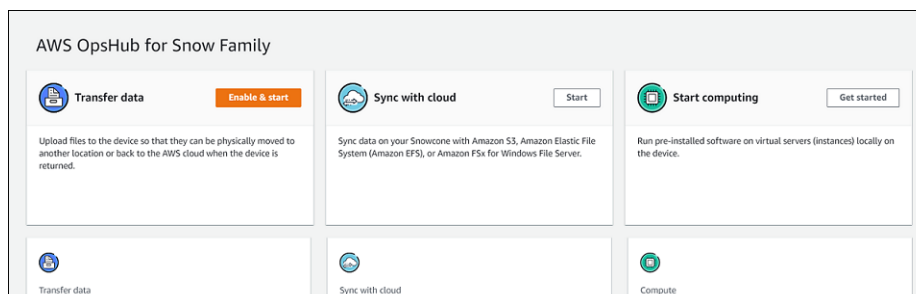
Topics

Using Amazon EC2 compute instances locally

You can use AWS OpsHub to run pre-installed software on virtual servers (instances) locally on your device, and also to manage Amazon EC2 instances on your device.

Topics

Launching an Amazon EC2 instance

Follow these steps to launch an Amazon EC2 instance using AWS OpsHub.

To launch an Amazon EC2 instance

1. Open the AWS OpsHub application.
2. In the **Start computing** section on the dashboard, choose **Get started**. Or, choose the **Services** menu at the top, and then choose **Compute (EC2)** to open the **Compute** page. All your compute resources appear in the **Resources** section.
3. If you have Amazon EC2 instances running on your device, they appear in the **Instance name** column under **Instances**. You can see details of each instance on this page.
4. Choose **Launch instance**. The launch instance wizard opens.
5. For **Device**, choose the Snow device that you want to launch the Amazon EC2 instance on.
6. For **Image (AMI)**, choose an Amazon Machine Image (AMI) from the list. This AMI is used to launch your instance.
7. For **Instance type**, choose one from the list.
8. Choose how you want to attach an IP address to the instance. You have the following options:

   - **Create public IP address (VNI)** – Choose this option to create a new IP address using a physical network interface. Choose a physical network interface and IP address assignment.

- **Use existing IP address (VNI)** – Choose this option to use an existing IP address and then use existing virtual network interfaces. Choose a physical network interface and a virtual network interface.

- **Do not attach IP address** – Choose this option if you don't want to attach an IP address.

9. Choose how you want to attach a key pair to the instance. You have the following options:

   Create key pair – Choose this option to create a new key pair and launch the new instance with this key pair.

   Use existing key pair – Choose this option to use an existing key pair to launch the instance.

   Do not attach IP address – Choose this option if you don't want to attach a key pair. You must acknowledge that you won't able to connect to this instance unless you already know the password that is built into this AMI.

   For more information, see Working with key pairs (p. 33).

10. Choose **Launch**. You should see your instance launching in the **Compute instances** section. The **State** is **Pending** and then changes to **Running** when done.

Stopping an Amazon EC2 instance

Use the following steps to use AWS OpsHub to stop an Amazon EC2 instance.

To stop an Amazon EC2 instance

1. Open the AWS OpsHub application.

2. In the **Start computing** section of the dashboard, choose **Get started**. Or, choose the **Services** menu at the top, and then choose **Compute (EC2)** to open the **Compute** page.

   All your compute resources appear in the **Resources** section.

3. If you have Amazon EC2 instances running on your device, they appear in the **Instance name** column under **Instances**.

4. Choose the instance that you want to stop, and choose **Stop**. The **State** changes to **Stopping**, and then to **Stopped** when done.

Starting an Amazon EC2 instance

Use these steps to start an Amazon EC2 instance using AWS OpsHub.

To start an Amazon EC2 instance

1. Open the AWS OpsHub application.

2. In the **Start computing** section of the dashboard, choose **Get started**. Or, choose the **Services** menu at the top, and then choose **Compute (EC2)** to open the **Compute** page.

   Your compute resources appear in the **Resources** section.

3. In the **Instance name** column, under **Instances**, find the instance that you want to start.

4. Choose the instance, and then choose **Start**. The **State** changes to **Pending**, and then changes to **Running** when done.

Working with key pairs

When you launch an Amazon EC2 instance and intend to connect to it using SSH, you have to provide a key pair. You can use Amazon EC2 to create a new key pair, or you can import an existing key pair or manage your key pairs.

To create, import, or manage key pairs

1. Open **Compute** on the AWS OpsHub dashboard.

2. In the navigation pane, choose the **Compute (EC2)** page, and then choose the **Key Pairs** tab. You are redirected to the Amazon EC2 console where you can create, import, or manage your key pairs.

3. For instructions on how to create and import key pairs, see Amazon EC2 key pairs and Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Terminating an Amazon EC2 instance

After you terminate an Amazon EC2 instance, you can't restart the instance.

To terminate an Amazon EC2 instance

1. Open the AWS OpsHub application.

2. In the **Start computing** section on the dashboard, choose **Get started**. Or, choose the **Services** menu at the top, and then choose **Compute (EC2)** to open the **Compute** page. You can see all your compute resources in the **Resources** section.

3. In the **Instance name** column, under **Instances**, find the instance that you want to terminate.

4. Choose the instance, and choose **Terminate**. The **State** changes to **Terminating**, and then to **Terminated** when done.

    After the instance is terminated, you can't restart it.

Using storage volumes locally

Amazon EC2 instances use Amazon EBS volumes for storage. In this procedure, you create a storage volume and attach it to your instance using AWS OpsHub.

To create a storage volume

1. Open the AWS OpsHub application.

2. In the **Start computing** section on the dashboard, choose **Get started**. Or, choose the **Services** menu at the top, and then choose **Compute (EC2)** to open the **Compute** page.

3. Choose the **Storage volumes** tab. If you have storage volumes on your device, the details about the volumes appear under **Storage volumes**.

4. Choose **Create volume** to open the **Create volume** page.

5. Choose the device that you want to create the volume on, enter the size (in GiBs) that you want to create, and choose the type of volume.

6. Choose **Submit**. The **State** is **Creating**, and changes to **Available** when done. You can see your volume and details about it in the **Volumes** tab.

To attach a storage volume to your instance

1. Choose the volume that you created, and then choose **Attach volume**.

2. For **Compute instance Id**, choose the instance you want to attach the volume to.

3. For **Volume Device Name**, enter the device name of your volume (for example, `/dev/sdh` or `xvdh`).

4. Choose **Attach**.

If you no longer need the volume, you can detach it from the instance and then delete it.

Using NFS file shares to manage file storage

You can use AWS OpsHub to upload files to your device and move them to other locations or the AWS Cloud when you return the device, or use AWS DataSync to transfer files.

Topics

You can configure your Snow Family device as an NFS file system and use your native file system to manage files on your device. You can upload files from an on-premises location to your device and then transfer the files to AWS or move them to other locations. You can use the AWS OpsHub defaults to configure NFS automatically or you can configure it manually yourself.

> **Note**
> You can provide CIDR blocks of IP ranges that are allowed to mount the NFS shares exposed by the device. For example, `10.0.0.0/16`. If you don't provide allowed CIDR blocks, all mount requests will be denied.
> Be aware that data transferred through NFS is not encrypted in transit.
> Other than the allowed hosts by CIDR blocks, your Snow Family device doesn't provide any authentication or authorization mechanism for the NFS shares.

> **Note**
> File names are object keys. The name for a key is a sequence of Unicode characters whose UTF-8 encoding is at most 1,024 bytes long. We recommend using NFSv4.1 where possible and encode file names with Unicode UTF-8 to ensure a successful data import. File names that are not encoded with UTF-8 might not be uploaded to S3 or might be uploaded to S3 with a different file name, depending on the NFS encoding that you use.
> Ensure that the maximum length of your file path is less than 1024 characters. Snow Family devices do not support file paths that are greater that 1024 characters. Exceeding this file path length will result in file import errors.
> For more information, see Working with object metadata in the *Amazon Simple Storage Service User Guide*.

Mounting NFS on a Windows client

If your client computer is using Windows 10 Enterprise or Windows 7 Enterprise, you first must start NFS service on Windows before you configure NFS in the AWS OpsHub application.

To mount NFS on a Windows client

1. On your client computer, open **Start**, choose **Control Panel** and choose **Programs**.

2. Choose **Turn Windows features on or off**.

3. Under **Services for NFS**, choose **Client for NFS** and choose **OK**.

Configuring NFS automatically (quick setup)

The NFS service is not running on the device by default so you need to start it to enable data transfer on the device. With a few clicks, your Snow Family device can automatically figure NFS for you, or you can configure it manually yourself.

> **Note**
> In Linux, mounting and unmount NFS endpoints requires root permissions.

To start and enable NFS on your Snow Family device automatically

1. In the **Transfer data** section on the dashboard, choose **Enable & start**. This could take a minute or two to complete.

2. When the NFS service is started, the IP address of the NFS server is shown on the dashboard and the **Transfer data** section indicates that the service is active.

3. Choose **Open in Explorer** (in Windows and Linux) to open the file share in your client's file browser and start transferring files from your client to your Snow Family device. You can copy and paste, or drag and drop files from your client computer into the file share. In Windows, your file share looks like the following `buckets(\\12.123.45.679)(Z:)`.

Configuring NFS manually

You can manually configure NFS by providing IP address (VNI) and restrict access to your file share.

To configure NFS manually

1. At the bottom of **Transfer data** section, on the dashboard, choose **Configure manually**.

2. Choose **Enable & start** to open the **Start NFS** wizard. The **Physical network interface** field is populated.

3. Choose **Create IP address (VNI)** or choose **Use existing IP address**.

4. If you choose **Create IP address (VNI)**, then choose **DHCP** or **Static IP** in the **IP Address assignment** list box.

   > **Important**
   > If you use a DHCP network, it is possible that the NFS client's IP address could be reassigned by the DCHP server. This can happen after the device has been disconnected and the IP addresses are recycled. If you set an allowed host range and the address of the client changes, another client can pick up that address. In this case, the new client will have access to the share. To prevent this, use DHCP reservations or static IP addresses.

   If you choose **Use existing IP address**, then choose a virtual interface from the **Virtual network interface** list box.

5. **Restrict NFS to allowed hosts** is selected by default. This restricts access to the NFS service to hosts you allow but you can choose **Allow all hosts**. We recommend restricting access. For more information about using NFS, see Using NFS for Offline Data Transfer .

6. In the **Allowed hosts** text box, provide the CIDR blocks of hosts that you want to allow to connect to the NFS service. For example, `10.0.0.0/16`.

7. Choose **Add allowed host** to add more hosts to allow.

8. Choose **Start NFS**. It could take about a minute or two to start. NFS uses 1 GB of ram and one of your CPUs. This limits the number of instances available.

   > **Important**
   > Don't turn off your device while the service is starting.

9. From the **Network File System (NFS) resource** section, the **State** of the NFS service shows as **Active**. Use the copy icon to copy the IP address of the NFS service. You will need this IP address for connect your NFS service when are ready to transfer files.

10. In the **Mount paths** box, you can filter and look for your endpoints.

11. For **Endpoint name**, choose an endpoint from the list, and choose **Mount NFS endpoint**. In Linux, mounting and unmount NFS endpoints requires root permissions. This endpoint is configured with the S3 bucket you specified when you ordered the device. The endpoint is shown under **NFS endpoints**. The endpoint is configured as an NFS file and shares. It appears as a drive letter and you can use your native operating system to drag and drop files onto and out of your device.

    The following are the default mount options:

    - Windows: `mount -o nolock rsize=128 wsize=128 mtype=hard endpoint:path *`

    - Linux: `mount -t nfs endpoint:path mount_point`

    - macOS: `mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 endpoint:path mount_point`

12. Choose the icon next to the drive letter to open the file share in your client's file browser. Then start transferring files from your client to your Snow Family device. You can copy and paste, or drag and drop files from your client computer into the file share. In Windows, your file share looks like the following: `buckets(\\12.123.45.679)(Z:)`

Stopping data transfer

To stop data transfer

1. From the dashboard, choose **Services** and then choose **File Storage**.
2. On the **File Storage** page, choose **Disable data transfer**. It usually takes up to 2 minutes for the NFS endpoints to disappear from the dashboard.

Using DataSync to transfer files to AWS

You can use AWS OpsHub to create an AWS DataSync agent on your Snowcone device. You can use it to transfer files between your device and Amazon S3, Amazon Elastic File System (Amazon EFS), or FSx for Windows File Server in the AWS Cloud.

AWS DataSync is an online data transfer service designed to simplify, automate, and accelerate copying large amounts of data to and from AWS storage services. DataSync copies data over the internet or AWS Direct Connect. As a fully managed service, DataSync removes much of the need to modify applications, develop scripts, or manage infrastructure.

DataSync supports data transfer between Network File System (NFS) and Amazon EFS, Amazon S3, or Amazon FSx for Windows File Server.

For information about the source and destination location combination supported by AWS DataSync, see Working with locations in the *AWS DataSync User Guide*.

Snowcone ships with the DataSync agent, which is a virtual machine (VM) that is used to read or write data from an on-premises storage system. To use DataSync, you first start the agent and then go the DataSync console and activate it. For information about DataSync, see Getting started with AWS DataSync.

To start the DataSync agent

1. On the AWS OpsHub dashboard, choose **Start** in the **Sync with cloud** section to open the **Start DataSync agent wizard**. The **Start DataSync agent** form is populated with the **Device IP address**, and **Physical network interface** fields.

2. Choose **Create IP address (VNI)** to create a virtual IP address or choose **Use existing IP address**.

3. If you choose **Create IP address (VNI)**, then choose **DHCP** or **Static IP** in the **IP Address assignment** list box.

   If you choose **Use existing IP address**, then choose a virtual interface from the **Virtual network interface** list box.

4. Choose **Start agent**. You are redirected to the **DataSync resource** page. It could take up to five minutes for the IP address of the agent to appear.

5. Use the copy icon to copy the IP address value of the agent from the **Agent IP address** file, and choose **Open DataSync console**.

   This opens the DataSync console, where you activate the agent and transfer your files. The rest of the setup is done in the AWS DataSync console.

Transferring files with DataSync

AWS Snowcone has already created the agent, so you only need to activate it, configure your source and destination location, create a task, and start the task.

To activate the DataSync agent and use the DataSync service

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.

2. In the **Activation** section, on the **Create agent page**, paste the IP address you copied into the **Agent address** box, and choose **Get key**. Your browser connects to the IP address and gets a unique activation key from your agent.

3. After the agent is activated, you will configure the NFS running on your device as a source location. For instructions, see Configure a source location in the *AWS DataSync User Guide*.

   > **Note**
   > The DataSync agent running on your Snowcone device can transfer files to and from a location that's reachable on your network.

4. On the **Configure a destination** page, choose and configure the destination you want to transfer files to. For instructions, see Configure a destination location in the *AWS DataSync User Guide*.

5. Configure task setting. For instructions, see Configure task settings in the *AWS DataSync User Guide*.

6. Review your settings and create your task. For instructions, see Review your settings and create your task in the *AWS DataSync User Guide*.

7. Start your task and wait for your files to be transferred. For instructions, see Start your task in the *AWS DataSync User Guide*.

Using AWS IoT Greengrass to run pre-installed software on Amazon EC2 instances

AWS IoT Greengrass is an open source Internet of Things (IoT) edge runtime and cloud service that helps you build, deploy, and manage IoT applications on your devices. You can use AWS IoT Greengrass to build software that enables your devices to act locally on the data that they generate, run predictions based

on machine learning models, and filter and aggregate device data. For detailed information about AWS IoT Greengrass, see What is AWS IoT Greengrass? in the *AWS IoT Greengrass Version 2 Developer Guide*.

By using AWS IoT Greengrass on your Snow Family device, you enable the device to collect and analyze data closer to where it is generated, react autonomously to local events, and communicate securely with other devices on the local network.

Setting up your Amazon EC2 instance

> **Note**
> To install AWS IoT Greengrass Version 2 on a Snow Family device, make sure that your device is connected to the internet. After installation, the internet is not required for a Snow Family device to work with AWS IoT Greengrass.

To set up an EC2 instance for AWS IoT Greengrass V2

1. On the AWS OpsHub dashboard, in the **Start Green Grass** section, choose **Get Started**.
2. Choose **Launch instance**.
3. Configure the instance with the settings that you want. The instance should have a public IP address and an SSH key.
4. Choose **Launch** in the launch instance window to launch the instance.
5. Open the Amazon EC2 console, and choose the **Instance** tab. Choose the instance and verify that it's running.

   Take note of the public IP address and SSH key name that are associated with the instance.
6. Connect to the EC2 instance using SSH. To do so, run the following command on the computer that is connected to your device. Replace *ssh-key* with the key you used to launch the EC2 instance. Replace *public-ip-address* with the public IP address of the EC2 instance.

   ```
   ssh -i ssh-key ec2-user@ public-ip-address
   ```

   > **Important**
   > If your computer uses an earlier version of Microsoft Windows, you might not have the SSH command, or you might have SSH but can't connect to your EC2 instance. To connect to your EC2 instance, you can install and configure PuTTY, which is a no-cost, open source SSH client. You must convert the SSH key from `.pem` format to PuTTY format and connect to your EC2 instance. For instructions on how to convert from `.pem` to PuTTY format, see the PuTTY documentation.

Installing AWS IoT Greengrass

Next, you set up your EC2 instance as an AWS IoT Greengrass Core device that you can use for local development.

To install AWS IoT Greengrass

1. Use the following command to install the prerequisite software for AWS IoT Greengrass. This command installs the AWS Command Line Interface (AWS CLI) v2, Python 3, and Java 8.

   ```
   "curl \"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip\" -o \"awscliv2.zip\"
    && unzip awscliv2.zip &&sudo ./aws/install && sudo yum -y install python3 java-1.8.0-
   openjdk"
   ```

2. Grant the root user permission to run the AWS IoT Greengrass software and modify the root permission from `root ALL=(ALL) ALL` to `root ALL=(ALL:ALL) ALL` in the sudoers config file.

```
sudo sed -in 's/root\tALL=(ALL)/root\tALL=(ALL:ALL)/' /etc/sudoers
```

3. Use the following command to download the AWS IoT Greengrass Core software.

```
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip >
 greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d GreengrassCore
  && rm greengrass-nucleus-latest.zip
```

4. Install and configure the AWS IoT Greengrass Core software. For instructions, see Getting started with AWS IoT Greengrass V2 in the *AWS IoT Greengrass Developer Guide*.

   Skip steps 1–3 and start with step 4. Steps 1–3 are not needed.

When you are finished, you will have an AWS IoT Greengrass core running on your Snow Family device for your local use.

Managing Your Devices

You use the AWS OpsHub to manage your Snow Family Devices. On the **Device details** page, you can perform the same the tasks that you do using the AWS CLI, including changing the alias of your device, rebooting the device, and checking for updates.

Topics

Rebooting Your Device

Follow these steps to use AWS OpsHub to reboot your Snow device.

> **Important**
> We highly recommend that you suspend all activities on the device before you reboot the device. Rebooting a device stops running instances, interrupts any writing to Amazon S3 buckets on the device, and stops any write operations from the file interface without clearing the cache.

To reboot a device

1. On the AWS OpsHub dashboard, find your device under **Devices**. Then choose the device to open the device details page.
2. Choose the **Reboot device** tab, and in the dialog box that appears, choose **Reboot device**. Your device starts to reboot.

Editing Your Device Alias

Use these steps to edit your device alias using AWS OpsHub.

To edit your device's alias

1. On the AWS OpsHub dashboard, find your device under **Devices**. Choose the device to open the device details page.

2. Choose the **Edit device alias** tab.

3. For **Device alias**, enter a new name, and choose **Save alias**.

Getting Updates for Your Device and the AWS OpsHub Application

You can check for updates for your device and install them. You can also configure AWS OpsHub to automatically update the application to the latest version.

Updating your device

Follow these steps to use AWS OpsHub to update your Snow device.

To update your device

1. On the AWS OpsHub dashboard, find your device under **Devices**. Choose the device to open the device details page.

2. Choose the **Check for updates** tab.

    The **Check for updates** page displays the current software version on your device and the latest software version, if there is one.

3. If there is an update, choose **Update**. Otherwise, choose **Close**.

Updating your AWS OpsHub application

AWS OpsHub automatically updates the application to the latest version. Follow these steps to verify that automatic update is enabled.

To verify that automatic updates are enabled for AWS OpsHub

1. On the AWS OpsHub dashboard, choose**Preferences**.

2. Open the **Updates** tab.

3. Verify that **Automatic updates enabled** is selected. Automatic update is enabled by default.

    If **Automatic updates enabled** is not selected, you will not get the latest version of the AWS OpsHub application.

Managing Profiles

A *profile* is a persistent storage of your credentials on your local file system. You can create a profile when you first unlock your device, or you can create one after your device is running. You can create new profiles, edit existing profiles, or delete them.

To create a profile

1. In the upper-right corner of the application, choose your name, and then choose **Manage profile**.

2. On the **Manage profiles** page, choose **Create profile**. You create a profile for each device.

    If your device is locked, see Unlocking a device (p. 26) for instructions.

3. Provide the name for the profile, the IP address of your device, and the unlock code.

4. Choose **Upload**, upload the manifest, and then choose **Create device**.

You now have a new profile for your device. You use this profile to log in to the device.

To edit a profile

1. In the upper-right corner of the application, choose your name, and then choose **Manage profile**.

    All your profiles appear on the page.
2. On the **Manage profiles** page, choose the profile that you want to edit.
3. On the next page, choose **Edit**. Make the changes that you want for your device, and choose **Save device**.

To delete a profile

1. In the upper-right corner of the application, choose your name, and then choose **Manage profile**.

    All your profiles appear on the page.
2. Choose the profile that you want to delete, and choose **Delete profile**.

Using the Snowball Edge Client

The Snowball Edge client is a standalone terminal application that you run on your local server to unlock your AWS Snowcone device and get credentials, logs, and status information. While using the Snowball Edge client, you can get additional support information by running the `snowballEdge help` command.

When you read and write data to the AWS Snowcone device, you use the file interface. You can also use the AWS OpsHub for Snow Family application to manage Snow Family Devices, including Snowcone devices. For more information, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

Downloading and Installing the Snowball Edge Client

You can download and install the Snowball Edge client from AWS Snowball Resources. On that page, find the installation package for your operating system and follow the instructions to install the Snowball Edge client. Running the Snowball Edge 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 Edge client must be run from the `~/snowball-client-linux-`*`build_number`*`/bin/` directory. Note that the Snowball Edge client is only supported on 64-bit Linux distributions.
- **macOS** – The `install.sh` script copies folders from the Snowball Edge client .tar file to the `/usr/local/bin/snowball` directory. If you run this script, you can then run the Snowball Edge 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 Edge Client

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

> **Note**
> The AWS Snowcone device uses the same Snowball Edge CLI commands, but it doesn't support commands that apply to clustering.

Topics

- Getting Your Certificate for Transferring Data (p. 48)
- AWS Snowcone Logs (p. 49)
- Getting Device Status (p. 50)
- Getting Service Status (p. 51)
- Launching the AWS DataSync AMI (p. 52)
- Starting NFS and Restricting Access (p. 54)
- Restricting Access to NFS Shares When NFS is Running (p. 54)
- Getting the Export Path for an Amazon S3 Bucket (p. 55)
- Enabling Local AWS Operator Debugging (p. 55)
- Disabling Local AWS Operator Debugging (p. 55)
- Creating a Direct Network Interface (p. 55)
- Getting Information About a Direct Network Interface (p. 56)
- Updating a Direct Network Interface (p. 56)
- Deleting a Direct Network Interface (p. 56)
- Checking feature status (p. 56)
- Changing feature status (p. 57)

Configuring a Profile for the Snowball Edge Client

Every time you run a command for the Snowball Edge client, you provide your manifest file, unlock code, and an IP address. You can get the first two of these from the AWS Snow Family Management Console or the job management API. For more information about getting your manifest and unlock code, see Getting Credentials (p. 47).

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 Edge client commands without having to manually enter these values for a particular job. After you configure the Snowball Edge client, the information is saved in a plaintext JSON format to *home directory*`/.aws/snowball/config/snowball-.config`.

The endpoint is the IP address, with `https://` added to it. You can locate the IP address for the AWS Snowcone device on the AWS Snowcone device LCD display. When the AWS Snowcone device 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 AWS Services on AWS Snowcone (p. 74).

> **Important**
> Anyone who can access the configuration file can access the data on your Snowcone device. 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 Example Output

```
Configuration will be stored at home directory\.aws\snowball\config\snowball-.config
Snowcone  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. For more information about multiple AWS CLI profiles, see Named Profiles in the *AWS Command Line Interface User Guide*.

Getting Your QR Code for NFC Validation

You can use this command to generate a device-specific QR code for use with the AWS Snowcone Verification App. You can download this app from the Apple App Store or Google Play store. For more information about NFC validation, see Validating NFC Tags (p. 103).

Usage

```
snowballEdge get-app-qr-code --output-file ~/downloads/snowball-qr-code.png
```

Example Example Output

```
QR code is saved to ~/downloads/snowball-qr-code.png
```

Unlocking an AWS Snowcone Device

To unlock a standalone AWS Snowcone device, run the `snowballEdge unlock-device` command. These commands authenticate your access to the AWS Snowcone device.

When you run one of these unlock commands, you can manually enter the path to the manifest file, the 29-character unlock code, and the IP address for your standalone device. This process can get tedious, so we recommend that you configure your Snowball Edge client instead. If you've already configured the Snowball Edge client, then you only need to enter the command itself without the path to the manifest, the unlock code, or the IP address.

> **Note**
> To unlock the device associated with your job, the device must be onsite, plugged into power and the network, and turned on. In addition, the LCD display on the front of the AWS Snowcone device must indicate that the device is ready for use.

Usage (configured Snowball Edge client)

```
snowballEdge unlock-device
```

Example Single Device Unlock Input

```
snowballEdge unlock-device
```

Example Single Device Unlock Output

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

Updating a Snowcone

Use the following commands to download and install updates for your Snowcone device. For procedures that use these commands, see Updating a Snowcone (p. 44).

`snowballEdge check-for-updates` – Returns version information about the Snowball software available in the cloud, and the current version installed on the device.

Usage (configured Snowball Edge client)

```
snowballEdge check-for-updates
```

Example Example Output

```
Latest version: 102
Installed version: 101
```

`snowballEdge describe-device-software` – Returns the current software version for the device. Additionally, if the update is being downloaded, the download state is also displayed. If a software update is in progress, the version manifest of update, and state of installation is also displayed. Following is a list of possible outputs:

- `NA` – No software updates are currently in progress.
- `Downloading` – New software is being downloaded.
- `Installing` – New software is being installed.
- `Requires Reboot` – New software has been installed, and the device must be restarted.

    Warning
    We highly recommend that you suspend all activity on the device before you restart the device. Restarting a device stops running instances, interrupts any writing to Amazon S3 buckets on the device, and stops any write operations from the file interface without clearing the cache. All of these processes can result in lost data.

Usage (configured Snowball Edge client)

```
snowballEdge describe-device-software
```

Example Example Output

```
Installed version: 101
Installing version: 102
Install State: Downloading
```

`snowballEdge download-updates` – Starts downloading the latest software updates for your Snowcone device.

Usage (configured Snowball Edge client)

```
snowballEdge download-updates
```

Example Example Output

```
Download started. Run describe-device-software API for additional information.
```

`snowballEdge install-updates` – Starts installing the latest software updates for your Snowcone device that were already downloaded.

Usage (configured Snowball Edge client)

```
snowballEdge install-updates
```

Example Example Output

```
Installation started.
```

`snowballEdge reboot-device` – Reboots the device.

> **Warning**
> We highly recommend that you suspend all activity on the device before you restart the device.
> Restarting a device stops running instances, interrupts any writing to Amazon S3 buckets on the
> device, and stops any write operations from the file interface without clearing the cache. All of
> these processes can result in lost data.

Usage (configured Snowball Edge client)

```
snowballEdge reboot-device
```

Example Example Output

```
Rebooting device now.
```

`snowballEdge configure-auto-update-strategies` – Configures an automatic update strategy.

Usage (configured Snowball Edge client)

```
snowballEdge configure-auto-update-strategy --auto-check autoCheck [--auto-check-frequency
autoCheckFreq] --auto-download autoDownload
[--auto-download-frequency autoDownloadFreq]
--auto-install autoInstall
[--auto-install-frequency autoInstallFreq]
--auto-reboot autoReboot [--endpoint
endpoint]
```

Example Example Output

```
Successfully configured auto update strategy. Run describe-auto-update-strategies for
 additional information.
```

`snowballEdge describe-auto-update-strategies` – Returns any currently configured automatic
update strategy.

Usage (configured Snowball Edge client)

```
snowballEdge describe-auto-update-strategies
```

Example Example Output

```
auto-update-strategy {[
auto-check:true,
```

```
auto-check-frequency: "0 0 * * FRI", // CRON Expression String, Every Friday at midnight
auto-download:true,
auto-download-frequency: "0 0 * * SAT", // CRON Expression String, Every Saturday at
midnight
auto-install:true,
auto-install-frequency: "0 13 * * Sun", // CRON Expression String, Every Saturday at
midnight
auto-reboot: false;
]}
```

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 Snowcone, and you can use them only on the device. The device doesn't have any AWS Identity and Access Management (IAM) permissions in the AWS Cloud.

> **Note**
> If you're using the AWS CLI with Snowball, 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 (configured Snowball Edge client)

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

Example Example Output

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

Usage (configured Snowball Edge client)

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

Example Example Output

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

Starting a Service on Your Snowcone Device

Snowcone supports multiple services, including compute instances, the NFS file interface, Amazon EC2, and AWS DataSync. You can start these services with the `snowballEdge start-service` command. To get the service ID for each service, you can use the `snowballEdge list-services` command.

Before you run this command, create a single virtual network interface to bind to the service that you're starting. For more information, see Creating a Virtual Network Interface (p. 82).

Usage (configured Snowball Edge client)

```
snowballEdge start-service --service-id service_id --virtual-network-interface-
arns virtual-network-interface-arn
```

Example Example Output

```
Starting the AWS service on your Snowball Edge . You can determine the status of the AWS
 service using the describe-service command.
```

Stopping a Service on Your Snowcone Device

To stop a service running on your Snowcone device, you can use the `snowballEdge stop-service` command. The Amazon EC2 services cannot be stopped.

> **Warning**
> Data loss can occur if the file interface is stopped before remaining buffered data is written to the device.

Usage (configured Snowball Edge client)

```
snowballEdge stop-service --service-id service_id
```

Example Example Output

```
Stopping the AWS service on your Snowball . You can determine the status of the AWS service
 using the describe-service command.
```

Getting Your Certificate for Transferring Data

To transfer data to a Snowcone device, use the NFS file interface or AWS DataSync. If you unlock your Snowcone 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 can get the new, updated certificate from the Snowcone device again using the `get-certificate` command.

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

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

  Usage (configured Snowball Edge client)

  ```
  snowballEdge list-certificates
  ```

  Example 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 (configured Snowball Edge client)

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

  Example Example Output

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

AWS Snowcone Logs

When you transfer data between your on-premises data center and a Snowcone device, 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 (configured Snowball Edge client)

  ```
  snowballEdge list-logs
  ```

  Example Example Output

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

- `get-log` – Downloads a copy of a specific log from the Snowcone device 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 (configured Snowball Edge client)

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

Example Example Output

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

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

  Usage (configured Snowball Edge client)

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

  Example Example Output

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

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

Getting Device Status

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

- `describe-device`

  Usage (configured Snowball Edge client)

  ```
  snowballEdge describe-device
  ```

  Example Example Output

  ```
  {
    "DeviceId" : "JID-EXAMPLE12345-123-456-7-890",
    "UnlockStatus" : {
      "State" : "UNLOCKED"
    },
    "ActiveNetworkInterface" : {
      "IpAddress" : "192.0.2.0"
    },
    "PhysicalNetworkInterfaces" : [ {
      "PhysicalNetworkInterfaceId" : "s.ni-EXAMPLEd9ecbf03e3",
      "PhysicalConnectorType" : "RJ45",
      "IpAddressAssignment" : "STATIC",
      "IpAddress" : "0.0.0.0",
      "Netmask" : "0.0.0.0",
      "DefaultGateway" : "192.0.2.1",
      "MacAddress" : "EX:AM:PL:E0:12:34"
    }, {
      "PhysicalNetworkInterfaceId" : "s.ni-EXAMPLE4c3840068f",
      "PhysicalConnectorType" : "RJ45",
      "IpAddressAssignment" : "STATIC",
      "IpAddress" : "0.0.0.0",
      "Netmask" : "0.0.0.0",
      "DefaultGateway" : "192.0.2.2",
  ```

```
      "MacAddress" : "EX:AM:PL:E0:56:78"
  }, {
      "PhysicalNetworkInterfaceId" : "s.ni-EXAMPLE0a3a6499fd",
      "PhysicalConnectorType" : "SFP_PLUS",
      "IpAddressAssignment" : "DHCP",
      "IpAddress" : "192.168.1.231",
      "Netmask" : "255.255.255.0",
      "DefaultGateway" : "192.0.2.3",
      "MacAddress" : "EX:AM:PL:E0:90:12"
  } ]
}
```

Getting Service Status

You can determine the status and general health of the services running on a Snowcone device by using the `describe-service` command. You can first run the `list-services` command to see what services are running.

- `list-services`

  Usage (configured Snowball Edge client)

  ```
  snowballEdge list-services
  ```

  Example Example Output

  ```
  {
    "ServiceIds" : [ "nfs", "datasync", "ec2" ]
  }
  ```

- `describe-service`

  This command returns a status value for a service. It also includes state information that might be helpful in resolving issues you encounter with the service. Those states are as follows.

  - `ACTIVE` – The service is running and available for use.

  - `ACTIVATING` – The service is starting up, but it is not yet available for use.

  - `DEACTIVATING` – The service is in the process of shutting down.

  - `INACTIVE` – The service is not running and is not available for use.

  Usage (configured Snowball Edge client)

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

  Example Example Output

  ```
  {
  "ServiceId" : "ec2",
    "Status" : {
      "State" : "ACTIVE"
    },
  "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"
}
} ]
}
```

Launching the AWS DataSync AMI

Launch the AWS DataSync AMI on Snowcone.

Usage (configured Snowball Edge client)

AWS DataSync must be launched with the `snc1.medium` instance type. Launching DataSync with a different instance type can result in an unstable operation and potential data loss. Use the `describe-images` command to find the image to launch an instance from. The output looks like the following.

```
{
  "ImageId": "s.ami-0c046f119de4f752f",
  "Public": false,
  "State": "AVAILABLE",
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda",
      "Ebs": {
        "DeleteOnTermination": true,
        "Iops": 0,
        "SnapshotId": "s.snap-0d7558ce444ab09bf",
        "VolumeSize": 20,
        "VolumeType": "sbp1"
      }
    }
  ],
  "Description": "AWS DataSync AMI for online data transfer",
  "EnaSupport": false,
  "Name": "scn-datasync-ami",
  "RootDeviceName": "/dev/sda"
}
```

```
aws ec2 describe-instances --endpoint \ http://${snowcone_ip}:8008
```

Example Example Output

```
{
    "Reservations": [
        {
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "s.image id",
                    "InstanceId": "s.instance id",
                    "InstanceType": "snc1.medium",
                    "LaunchTime": "2020-03-06T18:58:36.609Z",
```

```
            "PrivateIpAddress": "ip address",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda",
                    "Ebs": {
                        "AttachTime": "2020-03-06T19:14:21.336Z",
                        "DeleteOnTermination": true,
                        "Status": "attached",
                        "VolumeId": "s.volume id"
                    }
                }
            ],
            "EbsOptimized": false,
            "EnaSupport": false,
            "RootDeviceName": "/dev/sda",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "s.security group id"
                }
            ],
            "SourceDestCheck": false,
            "CpuOptions": {
                "CoreCount": 2,
                "ThreadsPerCore": 1
            }
        }
    ],
    "ReservationId": "s.r-80c8ee6b041b29eb4"
    },
    ]
}
```

Run the instance.

```
aws ec2 run-instances --image-id s.ami id \--instance-type snc1.medium --endpoint http://
${snowcone_ip}:8008
```

Example Example Output

```
{
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "s.ami-0623310b494365cc5",
            "InstanceId": "s.i-80c8ee6b041b29eb4",
            "InstanceType": "snc1.medium",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "EnaSupport": false,
            "RootDeviceName": "/dev/sda",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "s.sg-80c8ee6b041b29eb4"
                }
```

```
            ],
            "SourceDestCheck": false,
            "CpuOptions": {
                "CoreCount": 2,
                "ThreadsPerCore": 1
            }
        }
    ],
    "ReservationId": "s.r-80c8ee6b041b29eb4"
}
```

Starting NFS and Restricting Access

Important
Don't start the NFS service if you intend to use Amazon Elastic Block Store (Amazon EBS). The
first time NFS is started, all storage is allocated to NFS. It is not possible to reallocate NFS
storage to Amazon EBS, even if the NFS service is stopped.

> **Note**
> You can provide CIDR blocks for IP address ranges that are allowed to mount the NFS
> shares exposed by the device. For example, `10.0.0.0/16`. If you don't provide allowed
> CIDR blocks, all mount requests will be denied.
> Data transferred through NFS is not encrypted in transit.
> Other than the allowed hosts by CIDR blocks, Snowcone doesn't provide an
> authentication or authorization mechanism for the NFS shares.

Start NFS with the `snowballEdge start-service` command. To get the service ID for the NFS
service, you can use the `snowballEdge list-services` command.

Before you run this command, create a single virtual network interface to bind to the service that you're
starting. For more information, see Creating a Virtual Network Interface (p. 82). You can restrict access
to your file shares and data in your Amazon S3 buckets and see what restrictions are currently in place.
You do this by allocating CIDR blocks for allowed hosts that can access your file share and S3 buckets
when you start the NFS service.

Usage (configured Snowball Edge client)

```
snowballEdge start-service --service-id nfs --virtual-network-interface-arns
 arn:aws:snowball-device:::interface/s.ni-12345fgh45678j --service-configuration
 AllowedHosts=ip address-1/32,ip address-2/24
```

Example Example Output

```
Starting the service on your Snowball Edge. You can determine the status of the service
 using the describe-service command.
```

Restricting Access to NFS Shares When NFS is Running

You can restrict access your file shares and data in your Amazon S3 buckets after you have started NFS.
You can see what restrictions are currently in place, and give each bucket different access restrictions.
You do this by allocating CIDR blocks for hosts that can access your file share and S3 buckets when you
start the NFS service. The following is an example command.

Usage (configured Snowball Edge client)

```
snowballEdge start-service \
```

```
    --service-id nfs \
    --virtual-network-interface-arns virtual-network-interface-arn --service-configuration
 AllowedHosts=ip-address-1/32,ip-address-1/24
```

To see the current restrictions, use the `describe-service` command.

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

Getting the Export Path for an Amazon S3 Bucket

There is no specific Snowcone command for getting the export path of an Amazon S3 bucket. The format of the export path looks like the following.

```
/buckets/bucket-name.
```

Enabling Local AWS Operator Debugging

- `enable-local-aws-operator-debugging` – Enables the device for local AWS operator debugging by opening SSH port 22.

Usage (configured Snowball Edge client)

```
snowballEdge enable-local-aws-operator-debugging
```

Disabling Local AWS Operator Debugging

- `disable-local-aws-operator-debugging` – Disables the device for local AWS operator debugging by closing SSH port 22. By default, SSH port 22 is closed. When the Snowcone device is turned off or is power cycled, local AWS operator debugging is disabled.

Usage (configured Snowball Edge client)

```
snowballEdge disable-local-aws-operator-debugging
```

Creating a Direct Network Interface

- `create-direct-network-interface` – Creates a direct network interface (DNI). Creates a direct network interface to use with Amazon EC2 compute instances on your device. You can find the direct network interfaces available on your device by using the `describe-direct-network-interfaces` command.

Usage (configured Snowball Edge client)

```
create-direct-network-interface [--endpoint endpoint] [--instance-id instanceId] [--
mac macAddress]
                                [--manifest-file manifestFile] [--physical-network-
interface-id physicalNetworkInterfaceId]
                                [--profile profile] [--unlock-code unlockCode] [--
vlan vlanId]
```

Getting Information About a Direct Network Interface

- `describe-direct-network-interface` – Gets the direct network interfaces on your device. A direct network interface can be used to configure networking for Amazon EC2 compute instances and services on your device. You can create a new direct network interface by using the `create-direct-network-interface` command.

Usage (configured Snowball Edge client)

```
describe-direct-network-interfaces [--endpoint endpoint] [--manifest-file manifestFile] [--profile profile] [--unlock-code unlockCode]
```

Updating a Direct Network Interface

- `update-direct-network-interface` – Updates a direct network interface. Use this command to update a direct network interface that will be used with Amazon EC2 compute instances on your device. You can find the direct network interfaces that are available on your device by using the `describe-direct-network-interfaces` command. When you are modifying a network interface that is attached to an Amazon EC2 instance, the interface will first be detached.

Usage (configured Snowball Edge client)

```
update-direct-network-interface [--direct-network-interface-arn directNetworkInterfaceArn]
 [--endpoint endpoint]
                                [--mac macAddress]
                                [--manifest-file manifestFile] [--profile profile] [--
unlock-code unlockCode]
                                [--vlan vlanId] [--attach-instance-id instanceId | --
detach]
```

Deleting a Direct Network Interface

- `delete-direct-network-interface` – Deletes a direct network interface that is no longer in use. To delete a direct network interface associated with your Amazon EC2 compute instance, you must first disassociate the direct network interface from your instance.

Usage (configured Snowball Edge client)

```
delete-direct-network-interface [--direct-network-interface-arn directNetworkInterfaceArn]
 [--endpoint endpoint]
                                [--manifest-file manifestFile] [--profile profile] [--
unlock-code unlockCode]
```

Checking feature status

To list the status of features available on your device, including AWS Snow Device Management, which allows you to manage your Snowcone device and local AWS services remotely, use the `describe-features` command.

`RemoteManagementState` indicates the status of Snow Device Management and returns one of the following states:

- `INSTALLED_ONLY` – The feature is installed but not enabled.
- `INSTALLED_AUTOSTART` – The feature is enabled and the device will attempt to connect to its AWS Region when it is powered on.
- `NOT_INSTALLED` – The device does not support the feature or was already in the field before its launch.

Usage (configured Snowball Edge client)

```
snowballEdge describe-features
--manifest-file manifest.bin path
--unlock-code unlock-code
--endpoint https://device-local-ip:9091
```

Example Output

```
{
   "RemoteManagementState" : String
}
```

Changing feature status

To change the status of the features available on your AWS Snowcone device, use the `set-features` command. To enable or disable AWS Snow Device Management, which allows you to manage your Snowcone device and local AWS services remotely, use the `--remote-management-state` parameter. The device must be unlocked before you run this command.

You can set Snow Device Management to the following states:

- `INSTALLED_ONLY` – The feature is installed but not enabled.
- `INSTALLED_AUTOSTART` – The feature is enabled and the device attempts to connect to its AWS Region when it is powered on.

    Note
    The `NOT_INSTALLED` state exists only to identify devices that don't support Snow Device Management or were already in the field before its launch. It is not possible to install or uninstall the feature on devices that are already deployed. To use Snow Device Management, you must order a new device with the feature preinstalled.

Usage (configured Snowball Edge client)

```
snowballEdge set-features
--remote-management-state INSTALLED_AUTOSTART
--manifest-file ./JID2bf11d5a-fict-414a-b5b1-3bf7e6a6e83d_manifest.bin
--unlock-code 73bb0-f8ke1-69a4a-f4288-4f88d
--endpoint https://10.0.0.25
```

Example Output

```
{
   "RemoteManagementState" : "INSTALLED_AUTOSTART"
}
```

Using AWS Snow Device Management to Manage Devices

AWS Snow Device Management allows you to manage your AWS Snowcone device and local AWS services remotely. All Snowcone devices support Snow Device Management, and it comes preinstalled on new devices in most AWS Regions where Snowcone is available.

You can order a new device installed with Snow Device Management in the following ways:

- When you order a new Snowcone device from the AWS Management Console, you specify which state Snow Device Management is in when the device arrives. Snow Device Management can be installed in the following states:
  - `INSTALLED_ONLY` – Snow Device Management is installed but not activated.
  - `INSTALLED_AUTOSTART` – Snow Device Management is installed, and the device attempts to connect to its AWS Region when it is powered on.
- When you order a new Snowcone device through the AWS Command Line Interface (AWS CLI) or an AWS SDK, you use the `--remote-management` parameter to specify the `INSTALLED_ONLY` or `INSTALLED_AUTOSTART` states when running the `create-job` command. If you don't specify a value for this parameter, Snow Device Management defaults to `INSTALLED_ONLY` for supported devices.

  > **Note**
  > It is not possible to order a new Snowcone device without preinstalled Snow Device Management feature artifacts. The `NOT_INSTALLED` state exists only to identify devices that don't support the feature or that were already in the field before its launch. If you don't want to use Snow Device Management, set it to the INSTALLED_ONLY state.
  > Snow Device Management can't be added to Snowcone devices that are already deployed in the field. To use Snow Device Management, you must order a new device with the feature preinstalled.

The following example shows the syntax for the `--remote-management` parameter, in addition to other parameters that you might include for a typical `create-job` command. For more information, see Job Management API Reference in the "AWS Snow Family API Reference" guide.

Command

```
aws snowball create-job \
        --job-type IMPORT \
        --remote-management INSTALLED_AUTOSTART
        --device-configuration '{"SnowconeDeviceConfiguration": {"WirelessConnection":
 {"IsWifiEnabled": false} } }' \
        --resources '{"S3Resources":[{"BucketArn":"arn:aws:s3:::bucket-name"}]}' \
        --description "Description here" \
        --address-id ADID00000000-0000-0000-0000-000000000000 \
        --kms-key-arn arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
        --role-arn arn:aws:iam::000000000000:role/SnowconeImportGamma \
        --snowball-capacity-preference T8 \
        --shipping-option NEXT_DAY \
        --snowball-type SNC1_HDD \
        --region us-west-2 \
```

Topics

Managing devices remotely

If you specified `INSTALLED_AUTOSTART` for Snow Device Management during the job order, the feature is ready to use immediately when your Snowcone device arrives and is powered on for the first time.

If you specified `INSTALLED_ONLY` when ordering your Snowcone device, you must change the feature state to `INSTALLED_AUTOSTART` before the device can call back to its AWS Region to enable remote management. You can enable Snow Device Management at any time after you receive and unlock your device.

Enabling Snow Device Management on AWS Snowcone

Follow this procedure to enable Snow Device Management using the Snowball Edge CLI.

> **Note**
> This procedure requires the Snowball Edge client. Make sure you've installed the latest Snowball Edge client before you proceed. For more information, see Downloading and Installing the Snowball Client.

To enable Snow Device Management on your device

1. To download the manifest file for the job from AWS, use the following command. Replace *placeholder values* with your information.

    Command

    ```
    aws snowball get-job-manifest
    --job-id JID970A5018-F8KE-4D06-9F7B-335C1C7221E4
    ```

    Output

    ```
    {
        "ManifestURI": "https://awsie-frosty-manifests-prod.s3.us-east-1.amazonaws.com/
    JID970A5018-F8KE-4D06-9F7B-335C1C7221E4_manifest.bin"
    }
    ```

2. To download the unlock code for the job from AWS, use the following command. Replace *placeholder values* with your information.

    Command

```
aws snowball get-job-unlock-code
--job-id JID970A5018-F8KE-4D06-9F7B-335C1C7221E4
```

Output

```
{
    "UnlockCode": "7c0e1-bab84-f7675-0a2b6-f8k33"
}
```

3. Use a compatible power adapter to power your device and turn it on. Then connect the device to your network using an Ethernet cable or a Wi-Fi connection. For more information, see AWS Snowcone Power Supply and Accessories.

4. Make note of the local IP address shown on the device's display. You'll need this IP address for the next steps. This IP address is either obtained automatically through DHCP or statically configured.

5. To unlock the device, use the following command. Replace *placeholder values* with your information. For the `--endpoint` parameter, specify the device local IP address you noted previously.

Command

```
snowballEdge unlock-device
--manifest-file JID1717d8cc-2dc9-4e68-aa46-63a3ad7927d2_manifest.bin
--unlock-code 7c0e1-bab84-f7675-0a2b6-f8k33
--endpoint https://10.186.0.56:9091
```

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.
```

6. (Optional) To describe the features of the device, use the following command. Replace *placeholder values* with your information. For the `--endpoint` parameter, specify the device local IP address you noted previously.

Command

```
snowballEdge describe-features
--manifest-file JID1717d8cc-2dc9-4e68-aa46-63a3ad7927d2_manifest.bin
--unlock-code 7c0e1-bab84-f7675-0a2b6-f8k33
--endpoint https://10.186.0.56:9091
```

Output

```
{
   "RemoteManagementState" : "INSTALLED_ONLY"
}
```

7. To enable Snow Device Management, use the following command. Replace *placeholder values* with your information. For the `--endpoint` parameter, specify the device local IP address you noted previously.

    Command

    ```
    snowballEdge set-features
    --remote-management-state INSTALLED_AUTOSTART
    --manifest-file JID1717d8cc-2dc9-4e68-aa46-63a3ad7927d2_manifest.bin
    --unlock-code 7c0e1-bab84-f7675-0a2b6-f8k33
    --endpoint https://10.186.0.56:9091
    ```

    Output

    ```
    {
      "RemoteManagementState" : "INSTALLED_AUTOSTART"
    }
    ```

8. On the AWS account from which the device was ordered, create an AWS Identity and Access Management (IAM) role, and add the following policy to the role. Then, assign the role to the IAM user who will log in to remotely manage your device with Snow Device Management. For more information, see Creating IAM roles and Creating an IAM user in your AWS account.

    Policy

    ```
    {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    'snow-device-management:ListDevices',
                    'snow-device-management:DescribeDevice',
                    'snow-device-management:DescribeDeviceEc2Instances',
                    'snow-device-management:ListDeviceResources',
                    'snow-device-management:CreateTask',
                    'snow-device-management:ListTasks',
                    'snow-device-management:DescribeTask',
                    'snow-device-management:CancelTask',
                    'snow-device-management:DescribeExecution',
                    'snow-device-management:ListExecutions',
                    'snow-device-management:ListTagsForResource',
                    'snow-device-management:TagResource',
                    'snow-device-management:UntagResource'
                ],
                "Resource": "*"
            }
        ]
    }
    ```

Snow Device Management CLI commands

This section describes the AWS CLI commands that you can use to manage your Snow Family Devices remotely with Snow Device Management. You can also perform some remote management tasks using AWS OpsHub for Snow Family. For more information, see Unlocking a device remotely (p. 27).

Note
Before managing your device, make sure it is powered on, connected to your network, and can connect to the AWS Region where it was provisioned.

Topics

Create a task

To instruct one or more target devices to perform a task, such as unlocking or rebooting, use `create-task`. You specify target devices by providing a list of managed device IDs with the `--targets` parameter, and specify the tasks to perform with the `--command` parameter. Only a single command can be run on a device at a time.

Supported commands:

- `unlock` (no arguments)
- `reboot` (no arguments)

To create a task to be run by the target devices, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management create-task
--targets smd-fictbgr3rbcjeqa5
--command reboot={}
```

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
ServiceQuotaExceededException
```

Output

```
{
    "taskId": "st-ficthmqoc2pht111",
    "taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-
cjkwhmqoc2pht111"
}
```

Check task status

To check the status of a remote task running on one or more target devices, use the `describe-execution` command.

A task can have one of the following states:

- `QUEUED`
- `IN_PROGRESS`
- `CANCELED`
- `FAILED`
- `COMPLETED`
- `REJECTED`
- `TIMED_OUT`

To check the status of a task, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management describe-execution \
--taskId st-ficthmqoc2phtlef \
--managed-device-id smd-fictqic6gcldf111
```

Output

```
{
    "executionId": "1",
    "lastUpdatedAt": "2021-07-22T15:29:44.110000+00:00",
    "managedDeviceId": "smd-fictqic6gcldf111",
    "startedAt": "2021-07-22T15:28:53.947000+00:00",
    "state": "SUCCEEDED",
    "taskId": "st-ficthmqoc2pht111"
}
```

Check device info

To check device-specific information, such as the device type, software version, IP addresses, and lock status, use the `describe-device` command. The output also includes the following:

- `lastReachedOutAt` – When the device last contacted the AWS Cloud. Indicates that the device is online.
- `lastUpdatedAt` – When data was last updated on the device. Indicates when the device cache was refreshed.

To check device info, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management describe-device \
--managed-device-id smd-fictqic6gcldf111
```

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "associatedWithJob": "JID2bf11d5a-ea1e-414a-b5b1-3bf7e6a6e111",
    "deviceCapacities": [
        {
            "available": 158892032000,
            "name": "HDD Storage",
            "total": 158892032000,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 0,
            "name": "SSD Storage",
            "total": 0,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 3,
            "name": "vCPU",
            "total": 3,
            "unit": "Number",
            "used": 0
        },
        {
            "available": 5368709120,
```

```
            "name": "Memory",
            "total": 5368709120,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 0,
            "name": "GPU",
            "total": 0,
            "unit": "Number",
            "used": 0
        }
    ],
    "deviceState": "UNLOCKED",
    "deviceType": "SNC1_HDD",
    "lastReachedOutAt": "2021-07-23T21:21:56.120000+00:00",
    "lastUpdatedAt": "2021-07-23T21:21:56.120000+00:00",
    "managedDeviceId": "smd-fictqic6gcldf111",
    "managedDeviceArn": "arn:aws:snow-device-management:us-west-2:000000000000:managed-
device/smd-fictqic6gcldf111"
    "physicalNetworkInterfaces": [
        {
            "defaultGateway": "10.0.0.1",
            "ipAddress": "10.0.0.2",
            "ipAddressAssignment": "DHCP",
            "macAddress": "ab:cd:ef:12:34:56",
            "netmask": "255.255.252.0",
            "physicalConnectorType": "RJ45",
            "physicalNetworkInterfaceId": "s.ni-530f866d526d4b111"
        },
        {
            "defaultGateway": "10.0.0.1",
            "ipAddress": "0.0.0.0",
            "ipAddressAssignment": "STATIC",
            "macAddress": "ab:cd:ef:12:34:57",
            "netmask": "0.0.0.0",
            "physicalConnectorType": "RJ45",
            "physicalNetworkInterfaceId": "s.ni-8abc787f0a6750111"
        }
    ],
    "software": {
        "installState": "NA",
        "installedVersion": "122",
        "installingVersion": "NA"
    },
    "tags": {
        "Project": "PrototypeA"
    }
}
```

Check Amazon EC2 instance state

To check the current state of the Amazon EC2 instance, use the `describe-ec2-instances` command. The output is similar to that of the `describe-device` command, but the results are sourced from the device cache in the AWS Cloud and include a subset of the available fields.

To check the state of the Amazon EC2 instance, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management describe-device-ec2-instances \
--managed-device-id smd-fictbgr3rbcje111 \
--instance-ids s.i-84fa8a27d3e15e111
```

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "instances": [
        {
            "instance": {
                "amiLaunchIndex": 0,
                "blockDeviceMappings": [
                    {
                        "deviceName": "/dev/sda",
                        "ebs": {
                            "attachTime": "2021-07-23T15:25:38.719000-07:00",
                            "deleteOnTermination": true,
                            "status": "ATTACHED",
                            "volumeId": "s.vol-84fa8a27d3e15e111"
                        }
                    }
                ],
                "cpuOptions": {
                    "coreCount": 1,
                    "threadsPerCore": 1
                },
                "createdAt": "2021-07-23T15:23:22.858000-07:00",
                "imageId": "s.ami-03f976c3cadaa6111",
                "instanceId": "s.i-84fa8a27d3e15e111",
                "state": {
                    "name": "RUNNING"
                },
                "instanceType": "snc1.micro",
                "privateIpAddress": "34.223.14.193",
                "publicIpAddress": "10.111.60.160",
                "rootDeviceName": "/dev/sda",
                "securityGroups": [
                    {
                        "groupId": "s.sg-890b6b4008bdb3111",
                        "groupName": "default"
                    }
                ],
                "updatedAt": "2021-07-23T15:29:42.163000-07:00"
            },
            "lastUpdatedAt": "2021-07-23T15:29:58.
071000-07:00"
        }
    ]
}
```

Check task metadata

To check the metadata for a given task on a device, use the `describe-task` command. The metadata for a task includes the following items:

- The target devices
- The status of the task
- When the task was created
- When data was last updated on the device
- When the task was completed
- The description (if any) that was provided when the task was created

To check a task's metadata, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management describe-task \
--task-id st-ficthmqoc2pht111
```

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "completedAt": "2021-07-22T15:29:46.758000+00:00",
    "createdAt": "2021-07-22T15:28:42.613000+00:00",
    "lastUpdatedAt": "2021-07-22T15:29:46.758000+00:00",
    "state": "COMPLETED",
    "tags": {},
    "targets": [
        "smd-fictbgr3rbcje111"
    ],
    "taskId": "st-ficthmqoc2pht111",
    "taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-
ficthmqoc2pht111"
}
```

Cancel a task

To send a cancel request for a specific task, use the `cancel-task` command. You can cancel only tasks in the `QUEUED` state that have not yet run. Tasks that are already running can't be canceled.

Note
A task that you're attempting to cancel might still run if it is processed from the queue before the `cancel-task` command changes the task's state.

To cancel a task, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management cancel-task \
--task-id st-ficthmqoc2pht111
```

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "taskId": "st-ficthmqoc2pht111"
}
```

List commands and syntax

To return a list of all supported commands for the Snow Device Management API, use the `help` command. You can also use the `help` command to return detailed information about and syntax for a given command.

To list all the supported commands, use the following command.

Command

```
aws snow-device-management help
```

To return detailed information and syntax for a command, use the following command. Replace *command* with the name of the command that you're interested in.

Command

```
aws snow-device-management command help
```

List remote-manageable devices

To return a list of all devices on your account that have Snow Device Management enabled in the AWS Region where the command is run, use the `list-devices` command. `--max-results` and `--next-`

`token` are optional. For more information, see Using AWS CLI pagination options in the "AWS Command Line Interface User Guide".

To list remote-manageable devices, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management list-devices \
--max-results 10
```

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "devices": [
        {
            "associatedWithJob": "ID2bf11d5a-ea1e-414a-b5b1-3bf7e6a6e111",
            "managedDeviceId": "smd-fictbgr3rbcjeqa5",
            "managedDeviceArn": "arn:aws:snow-device-management:us-
west-2:000000000000:managed-device/smd-fictbgr3rbcje111"
            "tags": {}
        }
    ]
}
```

List task status across devices

To return the status of tasks for one or more target devices, use the `list-executions` command. To filter the return list to show tasks that are currently in a single specific state, use the `--state` parameter. `--max-results` and `--next-token` are optional. For more information, see Using AWS CLI pagination options in the "AWS Command Line Interface User Guide".

A task can have one of the following states:

- `QUEUED`
- `IN_PROGRESS`
- `CANCELED`
- `FAILED`
- `COMPLETED`
- `REJECTED`
- `TIMED_OUT`

To list task status across devices, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management list-executions \
--taskId st-ficthmqoc2phtlef \
--state SUCCEEDED \
--max-results 10
```

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "executions": [
        {
            "executionId": "1",
            "managedDeviceId": "smd-fictbgr3rbcje111",
            "state": "SUCCEEDED",
            "taskId": "st-ficthmqoc2pht111"
        }
    ]
}
```

List available resources

To return a list of the AWS resources available for a device, use the `list-device-resources` command. To filter the list by a specific type of resource, use the `--type` parameter. Currently, Amazon EC2 instances are the only supported resource type. `--max-results` and `--next-token` are optional. For more information, see Using AWS CLI pagination options in the "AWS Command Line Interface User Guide".

To list the available resources for a device, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management list-device-resources \
--managed-device-id smd-fictbgr3rbcje111 \
--type AWS::EC2::Instance
--next-token YAQGPwAT9l3wVKaGYjt4yS34MiQLWvzcShe9oIeDJr05AT4rXSprqcqQhhBEYRfcerAp0YYbJmRT=
--max-results 10
```

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
```

```
AccessDeniedException
```

Output

```
{
    "resources": [
        {
            "id": "s.i-84fa8a27d3e15e111",
            "resourceType": "AWS::EC2::Instance"
        }
    ]
}
```

List device or task tags

To return a list of tags for a managed device or task, use the `list-tags-for-resource` command.

To list the tags for a device, use the following command. Replace the example Amazon Resource Name (ARN) with the ARN for your device.

Command

```
aws snow-device-management list-tags-for-resource
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/smd-
fictbgr3rbcjeqa5
```

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
ThrottlingException
```

Output

```
{
    "tags": {
        "Project": "PrototypeA"
    }
}
```

List tasks by status

Use the `list-tasks` command to return a list of tasks from the devices in the AWS Region where the command is run. To filter the results by `IN_PROGRESS`, `COMPLETED`, or `CANCELED` status, use the `--state` parameter. `--max-results` and `--next-token` are optional. For more information, see Using AWS CLI pagination options in the "AWS Command Line Interface User Guide".

To list tasks by status, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management list-tasks \
--state IN_PROGRESS \
--next-token K8VAMqKiP2Cf4xGkmH8GMyZrgOF8FUb+d1OKTP9+P4pUb+8PhW+6MiXh4= \
--max-results 10
```

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "tasks": [
        {
            "state": "IN_PROGRESS",
            "tags": {},
            "taskId": "st-ficthmqoc2phtlef",
            "taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-
ficthmqoc2phtlef"

        }
    ]
}
```

Apply tags

To add or replace a tag for a device, or for a task on a device, use the `tag-resource` command. The `--tags` parameter accepts a comma-separated list of `Key=Value` pairs.

To apply tags to a device, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management tag-resource \
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/smd-
fictbgr3rbcjeqa5 \
--tags Project=PrototypeA
```

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
ThrottlingException
```

Remove tags

To remove a tag from a device, or from a task on a device, use the `untag-resources` command.

To remove tags from a device, use the following command. Replace each *user input placeholder* with your own information.

Command

```
aws snow-device-management untag-resources \
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/smd-
fictbgr3rbcjeqa5 \
--tag-keys Project
```

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
ThrottlingException
```

Using AWS Services on AWS Snowcone

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

For information about unlocking an AWS Snowcone device, see Using the Snowball Edge Client (p. 42).

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

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

The device has three doors, a front, a back, and a top, which all can be opened by latches. The top door contains the power cable for the device. The other two doors can be opened and slid inside the device so that they're out of the way while you're using it. By opening the doors, you get access to the LCD E Ink display embedded in the front side of the device, and the power and network ports in the back.

After your device arrives and is powered on, you're ready to use it.

Topics

Using Amazon EC2 Compute Instances

In this topic, you can find an overview of using Amazon Elastic Compute Cloud (Amazon EC2) compute instances on an AWS Snowcone device. The topic includes conceptual information, procedures, and examples.

> **Note**
> These features are not supported in the Asia Pacific (Mumbai) AWS Region.

Overview

You can run Amazon EC2 compute instances hosted on a Snowcone using the supported EC2 instance types. Like their cloud-based counterparts, these instances require Amazon Machine Images (AMIs) to launch. You choose the AMI to be that base image for an instance in the cloud before you create your Snowcone job. For information about supported instance types, see Using Amazon EC2 on Snowcone (p. 16).

If the job type is local compute, you might create a total of 8 TiB local EBS volumes and attach them to Amazon EC2 instances. This allows local EC2 instances to access more local capacity than the root

volume alone. This is local storage only, so data written to the EBS volumes is lost when the device is returned to AWS because it can't be imported into Amazon S3.

> **Note**
> The NFS server is not available for compute jobs. If you need to import or export data to or from the AWS Cloud, don't choose the local compute job type when you place your order.

To use a compute instance on a Snowcone, create a job and specify your AMIs. You can do this from the AWS Snow Family Management Console, with the AWS CLI, or with one of the AWS SDKs. Typically, you must perform some housekeeping prerequisites before creating your job to use your instances.

After your device arrives, you can start managing your AMIs and instances. You can manage your compute instances on a Snowcone through an Amazon EC2–compatible endpoint. This type of endpoint supports many of the Amazon EC2 CLI commands and actions for the AWS SDKs. You use the AWS OpsHub for Snow Family tool to manage your AMIs, compute instances, and AWS services. For more information, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

When you're done with your device, return it to AWS. If the device was used in an import job, the data transferred through the NFS file interface is imported into Amazon S3. Otherwise, we perform a complete erasure of the device when it is returned to AWS. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

> **Important**
>
> - Using encrypted AMIs on Snowcone devices is not supported.
> - Data in compute instances running on a Snowcone isn't imported into AWS.

Pricing for Compute Instances on Snowcone

There are additional costs associated with using compute instances. For more information, see AWS Snowcone pricing.

Prerequisites

Before creating your job, keep the following information in mind:

- Before you can add any AMIs to your job, you must have an AMI in your AWS account, and it must be a supported image type. Currently, supported AMIs are based on the Amazon Linux 2, CentOS 7 (x86_64) - with Updates HVM, or Ubuntu 16.04 LTS - Xenial (HVM) images. You can get these images from the AWS Marketplace.
- All AMIs must be based on Amazon Elastic Block Store (Amazon EBS), with a single volume.
- If you are planning connecting to a compute instance running on a Snowcone, you must use Secure Shell (SSH). To do so, you first add the key pair.

Creating a Job with Compute Instances

In this section, you create your first compute instance job.

> **Important**
> Be aware of the following points before you create your job:
>
> - Make sure that the vCPU, memory, and storage values associated with your AMI match the type of instance that you want to create.
> - If you're going to use SSH to connect to the instance *after* you launch the instance on your Snowcone, you must first perform the following procedure. You can't update the AMIs on your Snowcone after the fact. You must do this step before creating the job.

- Using encrypted AMIs or encrypted Amazon EBS volumes on AWS Snowcone devices is not supported.

Configure an AMI to Use SSH to Connect to Compute Instances Launched on the Device

To use Secure Shell (SSH) to connect to your compute instances on Snowcone devices, you must perform the following procedure. This procedure adds the SSH key to the AMI before creating your job. We also recommend that you use this procedure to set up your applications on the instance that you plan to use as the AMI for your job.

> **Important**
> If you don't follow this procedure, you can't connect to your instances with SSH when you receive your Snowcone device.

To put an SSH key into an AMI

1. Launch a new instance in the AWS Cloud based on the Amazon Linux 2 for Snow Family, CentOS 7 (x86_64) - with Updates HVM, or Ubuntu 16.04 LTS - Xenial (HVM) image.

   When you launch your instance, make sure that the storage size that you assign to the instance is appropriate for your later use on the Snowcone. In the Amazon EC2 console, you do this in **Step 4: Add Storage**. For a list of the supported sizes for compute instance storage volumes on a Snowcone, see AWS Snowcone Quotas (p. 117).

2. Install and configure the applications that you want to run on the Snowcone, and test that they work as expected.

3. Make a copy of the PEM/PPK file that you used for the SSH key pair to create this instance. Save this file to the server that you plan to use to communicate with the Snowcone. This file is required for using SSH to connect to the launched instance on your device, so make a note of the path to this file.

4. Save the instance as an AMI. For more information, see Creating an Amazon EBS-Backed Linux AMI in the *Amazon EC2 User Guide for Linux Instances.*

5. Repeat this procedure for each of the instances that you want to connect to using SSH. Make sure that you make copies of your different SSH key pairs and take note of the AMIs they're associated with.

Creating Your Job in the Console

Your next step is to create a job. Your job can be of any job type, including a cluster. To use the AWS Snow Family Management Console, follow the instructions in Getting Started (p. 20).

Creating Your Job in the AWS CLI

You can also create your job using the AWS CLI. To do this, open a terminal and run the following command, replacing the red text with your actual values.

```
aws snowballEdge create-job --job-type IMPORT --resources '{"S3Resources":
[{"BucketArn":"arn:aws:s3:::bucket-name"}],"Ec2AmiResources":[{"AmiId":"ami-12345678"}]}'
 --description Example --address-id ADIEXAMPLE60-1234-1234-5678-41fEXAMPLE57  --kms-key-
arn arn:aws:kms:us-west-2:012345678901:key/eEXAMPLE-1234-1234-5678-5b4EXAMPLE8e --role-
arn arn:aws:iam::012345678901:role/snowball-local-s3-lambda-us-west-2-role --snowball-
capacity-preference T100 --shipping-option SECOND_DAY --snowball-type SNOWCONE
```

After your device arrives and you unlock it, use the Snowball Edge client to get your local credentials. For more information, see Getting Credentials (p. 47).

Network Configuration for Compute Instances

After you launch your compute instances on a Snow Family device, you must provide it with an IP address by creating a network interface. Snow Family devices support two kinds of network interfaces, a virtual network interface and a direct network interface.

Virtual network interface (VNI)

A virtual network interface is the standard network interface for connecting to an EC2 instance on your Snow Family device. You must create a VNI for each of your EC2 instances regardless of whether you also use a direct network interface or not. The traffic passing through a VNI is protected by the security groups that you set up. You can only associate VNIs with the physical network port you use to control your Snow Family device.

Direct network interface (DNI)

A direct network interface (DNI) is an advanced network feature that enables use cases like multicast streams, routing, and load balancing. By providing instances with layer 2 network access without any intermediary translation or filtering, you can gain increased flexibility over the network configuration of your Snow Family device and improved network performance. DNIs can be associated with any physical network port, allowing you to use any or all of the physical network ports on your Snow Family device. Multiple DNIs can be associated with an EC2 instance. DNIs support VLAN tags and customizing the MAC address. Traffic on DNIs is not protected by security groups.

Topics

- Prerequisites (p. 77)
- Setting Up a Virtual Network Interface (VNI) (p. 78)
- Setting Up a Direct Network Interface (DNI) (p. 78)

Prerequisites

Before you configure a VNI or a DNI, be sure that you've done the following prerequisites.

1. Make sure there's power to your device and that one of your physical network interfaces, like the RJ45 port, is connected with an IP address.
2. Get the IP address associated with the physical network interface that you're using on the Snow Family device.
3. Configure your Snowball Edge client. For more information, see Configuring a Profile for the Snowball Edge Client.
4. Unlock the device. We recommend using AWS OpsHub for Snow Family to unlock your device. For instructions, see Unlocking a Device.

   If you want to use the CLI command, run the following command, and provide the information that appears in the dialog box.

   ```
   snowballEdge configure
   ```

   Snowball Edge Manifest Path: manifest.bin

   Unlock Code: *unlock code*

   Default Endpoint: https://*device ip*
5. Run the following command.

```
snowballEdge unlock-device
```

The device display update indicates that it is unlocked.

6. Launch an EC2 instance on the device. You will associate the VNI with this instance.

7. Run the `snowballEdge describe-device` command to get the list of physical network interface IDs.

8. Identify the ID for the physical network interface that you want to use, and make a note of it.

Setting Up a Virtual Network Interface (VNI)

After you have identified the ID for your physical network interface, you can set up a virtual network interface (VNI). Use the following procedure set up a VNI. Make sure that you perform the prerequisite tasks before you create a VNI.

Create a VNI and associate IP address

1. Run the `snowballEdge create-virtual-network-interface` command. The following examples show running this command with the two different IP address assignment methods, either `DHCP` or `STATIC`. The `DHCP` method uses Dynamic Host Configuration Protocol (DHCP).

```
snowballEdge create-virtual-network-interface \
--physical-network-interface-id s.ni-abcd1234 \
--ip-address-assignment DHCP


        //OR//

snowballEdge create-virtual-network-interface \
--physical-network-interface-id s.ni-abcd1234 \
--ip-address-assignment STATIC \
--static-ip-address-configuration IpAddress=192.0.2.0,Netmask=255.255.255.0
```

The command returns a JSON structure that includes the IP address. Make a note of that IP address for the `ec2 associate-address` AWS CLI command later in the process.

Anytime you need this IP address, you can use the `snowballEdge describe-virtual-network-interfaces` Snowball Edge client command, or the `aws ec2 describe-addresses` AWS CLI command to get it.

2. To associate your newly created IP address with your instance, use the following command, replacing the red text with your values:

```
aws ec2 associate-address --public-ip 192.0.2.0 --instance-id s.i-01234567890123456 --
endpoint Snow Family device physical IP address:8008
```

Setting Up a Direct Network Interface (DNI)

Note

The direct network interface feature is available on or after January 12, 2021 and is available in all AWS Regions where Snow Family devices are available.

Prerequisites

Before you set up a direct network interface (DNI), you must perform the tasks in the prerequisites section.

1. Perform the prerequisite tasks before setting up the DNI. For instructions, see Prerequisites (p. 77).

2. Additionally, you must launch an instance on your device, create a VNI, and associate it with the instance. For instructions, see Setting Up a Virtual Network Interface (VNI) (p. 78).

    Note
    If you added direct networking to your existing device by performing an in-the-field software update, you must restart the device twice to fully enable the feature.

Create a DNI and associate IP address

1. Create a direct network interface and attach it to the Amazon EC2 instance by running the following command. You will need the MAC address of the device for the next step.

```
create-direct-network-interface [--endpoint endpoint] [--instance-id instanceId] [--
mac macAddress]
                                [--physical-network-interface-
id physicalNetworkInterfaceId]
                                [--unlock-code unlockCode] [--vlan vlanId]
```

    OPTIONS

    --endpoint <endpoint> The endpoint to send this request to. The endpoint for your devices will be a URL using the `https` scheme followed by an IP address. For example, if the IP address for your device is 123.0.1.2, the endpoint for your device would be https://123.0.1.2.

    --instance-id <instanceId> The EC2 instance ID to attach the interface to (optional).

    --mac <macAddress> Sets the MAC address of the network interface (optional).

    --physical-network-interface-id <physicalNetworkInterfaceId> The ID for the physical network interface on which to create a new virtual network interface. You can determine the physical network interfaces available on your Snowball Edge using the `describe-device` command.

    --vlan <vlanId> Set the assigned VLAN for the interface (optional). When specified, all traffic sent from the interface is tagged with the specified VLAN ID. Incoming traffic is filtered for the specified VLAN ID, and has all VLAN tags stripped before being passed to the instance.

2. If you didn't associate your DNI with an instance in step 1, you can associate it by running the Updating a Direct Network Interface (p. 56) command.

3. After you create a DNI and associate it with your EC2 instance, you must make two configuration changes inside your Amazon EC2 instance.

    - The first is to change ensure that packets meant for the VNI associated with the EC2 instance are sent through eth0.
    - The second change configures your direct network interface to use either DCHP or static IP when booting.

    To set these configurations, you must first SSH into your EC2 instance and run a script to route packets that are meant for the VNI to eth0 and set your DNI to either DHCP or static IP.

    The following is an example of a shell script for CentOS that makes these configuration changes.

```
# Mac address of the direct network interface. You got this when you created the direct
 network interface.
DNI_MAC=[MAC ADDRESS FROM CREATED DNI]
# The name to use for the direct network interface. You can pick any name that isn't
 already in use.
```

```
DNI=eth1

# Configure routing so that packets meant for the VNIC always are sent through eth0
PRIVATE_IP=$(curl -s http://ip address/latest/meta-data/local-ipv4)
PRIVATE_GATEWAY=$(ip route show to match 0/0 dev eth0 | awk '{print $3}')
ROUTE_TABLE=10001
echo from $PRIVATE_IP table $ROUTE_TABLE > /etc/sysconfig/network-scripts/rule-eth0
echo default via $PRIVATE_GATEWAY dev eth0 table $ROUTE_TABLE > /etc/sysconfig/network-
scripts/route-eth0

# Configure your direct network interface to use DHCP on boot.
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-$DNI
DEVICE="$DNI"
NAME="$DNI"
HWADDR="$DNI_MAC"
ONBOOT=yes
NOZEROCONF=yes
BOOTPROTO=dhcp
TYPE=Ethernet
EOF

# Rename DNI device if needed.
CURRENT_DEVICE_NAME=$(LANG=C ip -o link | awk -F ': ' -vIGNORECASE=1 '!/link\/
ieee802\.11/ && /'"$DNI_MAC"'/ { print $2 }')
ip link set $CURRENT_DEVICE_NAME name $DNI

# Make all changes live.
systemctl restart network
```

Additional commands

For all available commands for AWS Snowcone devices, see Using the Snowball Edge Client.

Connecting to Your Compute Instance on a Snowcone Using SSH

To use SSH to connect to your compute instances on Snowcone devices, you must first provide the SSH key to the AMI before creating your job. For more information on that procedure, see Configure an AMI to Use SSH to Connect to Compute Instances Launched on the Device (p. 76). If you haven't followed that procedure, you can't use SSH to connect to your instances.

To connect to your instance with SSH

1. Make sure that your device is powered on, connected to the network, and unlocked.

2. Make sure that you have your network settings configured for your compute instances. For more information, see Network Configuration for Compute Instances (p. 77).

3. Check your notes to find the PEM or PPK key pair that you used for this specific instance. Make a copy of those files somewhere on your computer. Make a note of the path to the PEM file.

4. Connect to your instance through SSH as in the following example command. The IP address is the IP address of the virtual network interface (VNIC) that you set up in Network Configuration for Compute Instances (p. 77).

```
ssh -i path/to/PEM/key/file instance-user-name@192.0.2.0
```

For more information, see Connecting to Your Linux Instance Using SSH in the *Amazon EC2 User Guide for Linux Instances.*

Snowcone Client Commands for Compute Instances

The Snowball Edge client is a standalone terminal application that you can run on your local server. It enables you to perform some administrative tasks on your Snowcone device or cluster of devices. For more information about how to use the Snowball Edge client, including how to start and stop services with it, see Using the Snowball Edge Client (p. 42).

Following, you can find information on the Snowball Edge client commands that are specific to compute instances, including examples of use. For a list of Amazon EC2 commands you can use on your AWS Snowcone device, see Supported AWS CLI Commands for Amazon EC2 on a Snowcone (p. 85).

> **Note**
> Commands related to clusters are not supported and will return an error.

Creating a Launch Configuration to Autostart Amazon EC2 Instances

To automatically start Amazon EC2 compute instances on your AWS Snowcone device after it is unlocked, you can create a launch configuration. To do so, use the `snowballEdge create-autostart-configuration` command, whose usage is shown following.

Usage

```
                snowballEdge create-autostart-configuration --physical-connector-type
 [SFP_PLUS or RJ45]
                                        --ip-address-assignment [DHCP or STATIC]
                                        [--static-ip-address-configuration
 IpAddress=[IP address],NetMask=[Netmask]]

                                        --launch-template-id
                                        [--launch-template-version]
```

Updating a Launch Configuration to Autostart EC2 Instances

To update an existing launch configuration on your Snowcone, use the `snowballEdge update-autostart-configuration` command. You can find its usage following. To enable or disable a launch configuration, specify the `--enabled` parameter.

Usage

```
                snowballEdge update-autostart-configuration --autostart-configuration-arn
                                        [--physical-connector-type [SFP_PLUS or RJ45]]
                                        [--ip-address-assignment [DHCP or STATIC]]
                                        [--static-ip-address-configuration
 IpAddress=[IP address],NetMask=[Netmask]]
                                        [--launch-template-id]
                                        [--launch-template-version]
                                        [--enabled]
```

Deleting a Launch Configuration to Autostart EC2 Instances

To delete a launch configuration that's no longer in use, use the `snowballEdge delete-autostart-configuration` command. You can find its usage following.

Usage

```
                snowballEdge delete-autostart-configuration --autostart-configuration-arn
```

Listing Launch Configurations to Autostart EC2 Instances

To list the launch configurations that you have created on your Snowcone, use the `describe-autostart-configurations` command. You can find its usage following.

Usage

```
                snowballEdge describe-autostart-configurations
```

Creating a Virtual Network Interface

To run a compute instance on your Snowcone or start the file interface on your Snowcone, you first create a virtual network interface (VNIC). Each Snowcone has three network interfaces (NICs), the physical network interface controllers for the device. These are the RJ45 ports on the back of the device.

Each VNIC is based on a physical one, and you can have any number of VNICs associated with each NIC. To create a virtual network interface, use the `snowballEdge create-virtual-network-interface` command.

> **Note**
> The `--static-ip-address-configuration` parameter is valid only when using the `STATIC` option for the `--ip-address-assignment` parameter.

Usage

You can use this command in two ways: with the Snowball Edge client configured, or without the Snowball Edge client configured. The following usage example shows the method with the Snowball Edge client configured.

```
snowballEdge create-virtual-network-interface --ip-address-assignment [DHCP or STATIC]
 --physical-network-interface-id [physical network interface id] --static-ip-address-
configuration IpAddress=[IP address],NetMask=[Netmask]
```

The following usage example shows the method without the Snowball Edge client configured.

```
snowballEdge create-virtual-network-interface --endpoint https://[ip address] --manifest-
file /path/to/manifest --unlock-code [unlock code] --ip-address-assignment [DHCP or STATIC]
 --physical-network-interface-id [physical network interface id] --static-ip-address-
configuration IpAddress=[IP address],NetMask=[Netmask]
```

Example Example: Creating VNICs (Using DHCP)

```
snowballEdge create-virtual-network-interface --ip-address-assignment dhcp --physical-
network-interface-id s.ni-8EXAMPLEaEXAMPLEd
{
  "VirtualNetworkInterface" : {
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-8EXAMPLE8EXAMPLEf",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.0",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
```

```
      "MacAddress" : "EX:AM:PL:E1:23:45"
   }
}
```

Describing Your Virtual Network Interfaces

To describe the VNICs that you previously created on your device, use the `snowballEdge describe-virtual-network-interfaces` command. You can find its usage following.

Usage

You can use this command in two ways: with the Snowball Edge client configured, or without the Snowball Edge client configured. The following usage example shows the method with the Snowball Edge client configured.

```
snowballEdge describe-virtual-network-interfaces
```

The following usage example shows the method without the Snowball Edge client configured.

```
snowballEdge describe-virtual-network-interfaces --endpoint https://[ip address] --
manifest-file /path/to/manifest --unlock-code [unlock code]
```

Example Example: Describing VNICs

```
snowballEdge describe-virtual-network-interfaces
[
   {
     "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-8EXAMPLE8EXAMPLE8",
     "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
     "IpAddressAssignment" : "DHCP",
     "IpAddress" : "192.0.2.0",
     "Netmask" : "255.255.255.0",
     "DefaultGateway" : "192.0.2.1",
     "MacAddress" : "EX:AM:PL:E1:23:45"
   },{
     "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-1EXAMPLE1EXAMPLE1",
     "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
     "IpAddressAssignment" : "DHCP",
     "IpAddress" : "192.0.2.2",
     "Netmask" : "255.255.255.0",
     "DefaultGateway" : "192.0.2.1",
     "MacAddress" : "12:34:5E:XA:MP:LE"
   }
]
```

Updating a Virtual Network Interface

After creating a virtual network interface (VNIC), you can update its configuration using the `snowballEdge update-virtual-network-interface` command. After providing the Amazon Resource Name (ARN) for a particular VNIC, you provide values only for whatever elements you are updating.

Usage

You can use this command in two ways: with the Snowball Edge client configured, or without the Snowball Edge client configured. The following usage example shows the method with the Snowball Edge client configured.

```
snowballEdge update-virtual-network-interface --virtual-network-interface-arn [virtual
 network-interface-arn] --ip-address-assignment [DHCP or STATIC] --physical-network-
interface-id [physical network interface id] --static-ip-address-configuration
 IpAddress=[IP address],NetMask=[Netmask]
```

The following usage example shows the method without the Snowball Edge client configured.

```
snowballEdge update-virtual-network-interface --endpoint https://[ip address] --
manifest-file /path/to/manifest --unlock-code [unlock code] --virtual-network-interface-
arn [virtual network-interface-arn] --ip-address-assignment [DHCP or STATIC] --physical-
network-interface-id [physical network interface id] --static-ip-address-configuration
 IpAddress=[IP address],NetMask=[Netmask]
```

Example Example: Updating a VNIC (Using DHCP)

```
snowballEdge update-virtual-network-interface --virtual-network-interface-arn
 arn:aws:snowball-device:::interface/s.ni-8EXAMPLEbEXAMPLEd --ip-address-assignment dhcp
```

Deleting a Virtual Network Interface

To delete a virtual network interface, you can use the `snowballEdge delete-virtual-network-interface` command.

Usage

You can use this command in two ways: with the Snowball Edge client configured, or without the Snowball Edge client configured. The following usage example shows the method with the Snowball Edge client configured.

```
snowballEdge delete-virtual-network-interface --virtual-network-interface-arn [virtual
 network-interface-arn]
```

The following usage example shows the method without the Snowball Edge client configured.

```
snowballEdge delete-virtual-network-interface --endpoint https://[ip address] --manifest-
file /path/to/manifest --unlock-code [unlock code] --virtual-network-interface-arn [virtual
 network-interface-arn]
```

Example Example: Deleting a VNIC

```
snowballEdge delete-virtual-network-interface --virtual-network-interface-arn
 arn:aws:snowball-device:::interface/s.ni-8EXAMPLEbEXAMPLEd
```

Using the Amazon EC2 Endpoint

Following, you can find an overview of the Amazon Elastic Compute Cloud (Amazon EC2) endpoint. Using this endpoint, you can manage your Amazon Machine Images (AMIs) and compute instances programmatically using Amazon EC2 API operations.

Topics

Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint

When you use the AWS CLI to issue a command to the AWS Snowcone device, you can specify that the endpoint is the Amazon EC2 endpoint. You have the choice of using the HTTPS endpoint, or an unsecured HTTP endpoint, as shown following.

HTTPS secured endpoint

```
aws ec2 describe-instances --endpoint https://192.0.2.0:8243 --ca-bundle path/to/
certificate
```

HTTP unsecured endpoint

```
aws ec2 describe-instances --endpoint http://192.0.2.0:8008
```

If you use the HTTPS endpoint of `8243`, your data in transit is encrypted. This encryption is ensured with a certificate that's generated by the Snowcone whenever it is unlocked. 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 Amazon EC2 endpoint

1. Connect the Snowcone to power and the network, and turn it on.
2. After the device has finished unlocking, make a note of its IP address on your local network.
3. From a terminal on your network, make sure that you can ping the Snowcone device.
4. Run the `snowballEdge get-certificate` command in your terminal. For more information about this command, see Getting Your Certificate for Transferring Data (p. 48).
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 profile.snowball.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.

Unsupported Amazon EC2 Features for Snowcone

Using the Amazon EC2 endpoint, you can programmatically manage your AMIs and compute instances on a Snowcone with Amazon EC2 API operations. However, not all features and API operations are supported for use with a Snowcone device.

Any features or actions not explicitly listed as supported in this guide are not supported. For example, the following Amazon EC2 actions are not supported for use with Snowcone:

- create-nat-gateway
- create-key-pair

Supported AWS CLI Commands for Amazon EC2 on a Snowcone

Following, you can find information about how to specify the Amazon EC2 endpoint for applicable AWS CLI commands. For information about installing and setting up the AWS CLI, including specifying what AWS Regions you want to make AWS CLI calls against, see the AWS Command Line Interface User Guide.

List of Supported Amazon EC2 AWS CLI Commands on a Snowcone

Following, you can find a description of the subset of AWS CLI commands and options for Amazon EC2 that are supported on Snowcone devices. If a command or option isn't listed following, it's not supported. You can declare some unsupported options along with a command. However, these are ignored.

- associate-address – Associates a virtual IP address with an instance for use on one of the three physical network interfaces on the device:
  - --instance-id – The ID of a single `sbe` instance.
  - --public-ip – The virtual IP address that you want to use to access your instance.
- attach-volume – Attaches an Amazon EBS volume to a stopped or running instance on your AWS Snowcone device and exposes it to the instance with the specified device name.
  - --device `value` – The device name.
  - --instance-id – The ID of a target Amazon EC2 instance.
  - --volume-id `value` – The ID of the EBS volume.
- authorize-security-group-egress – Adds one or more egress rules to a security group for use with a Snowcone device. Specifically, this action permits instances to send traffic to one or more destination IPv4 CIDR address ranges. For more information, see Security Groups in Snow Devices (p. 96).
  - --group-id `value` – The ID of the security group
  - [--ip-permissions `value`] – One or more sets of IP permissions.
- authorize-security-group-ingress – Adds one or more ingress rules to a security group. When calling `authorize-security-group-ingress`, you must specify a value either for `group-name` or `group-id`.
  - [--group-name `value`] – The name of the security group.
  - [--group-id `value`] – The ID of the security group
  - [--ip-permissions `value`] – One or more sets of IP permissions.
  - [--protocol `value`] The IP protocol. Possible values are `tcp`, `udp`, and `icmp`. The `--port` argument is required unless the "all protocols" value is specified (-1).
  - [--port `value`] – For TCP or UDP, the range of ports to allow. This value can be a single integer or a range (minimum–maximum).

    For ICMP, a single integer or a range (`type-code`) in which `type` represents the ICMP type number and `code` represents the ICMP code number. A value of -1 indicates all ICMP codes for all ICMP types. A value of -1 just for `type` indicates all ICMP codes for the specified ICMP type.
  - [--cidr `value`] – The CIDR IP range.
- create-launch-template – Creates a launch template. A *launch template* contains the parameters to launch an instance. When you launch an instance using `RunInstances`, you can specify a launch template instead of providing the launch parameters in the request. You can create up to 100 templates per AWS Snowcone device.
  - --launch-template-name `string` – A name for the launch template.
  - --launch-template-data `structure` – The information for the launch template. The following attributes are supported:
    - `ImageId`
    - `InstanceType`
    - `SecurityGroupIds`
    - `TagSpecifications`
    - `UserData`

    JSON syntax:

```
'{
    "ImageId":"string",
    "InstanceType":"sbe-c.large",
    "SecurityGroupIds":["string", ...],
    "TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"Name","Value":"Test"},
    {"Key":"Stack","Value":"Gamma"}]}],
create-ec2-job    "UserData":"this is my user data"
}'
```

- [--version-description `string`] – A description for the first version of the launch template.

- --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).

- create-launch-template-version – Creates a new version for a launch template. You can specify an existing version of a launch template from which to base the new version. Launch template versions are numbered in the order in which they are created. You can't specify, change, or replace the numbering of launch template versions. You can create up to 100 versions of each launch template.

  Specify either the launch template ID or launch template name in the request.

  - --launch-template-id `string` – The ID of the launch template.

  - --launch-template-name `string` – A name for the launch template.

  - --launch-template-data `structure` – The information for the launch template. The following attributes are supported:

    - `ImageId`

    - `InstanceType`

    - `SecurityGroupIds`

    - `TagSpecifications`

    - `UserData`

    JSON syntax:

```
'{
    "ImageId":"string",
    "InstanceType":"sbe-c.large",
    "SecurityGroupIds":["string", ...],
    "TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"Name","Value":"Test"},
     {"Key":"Stack","Value":"Gamma"}]}],
    "UserData":"this is my user data"
}'
```

  - [--source-version `string`] – The version number of the launch template on which to base the new version. The new version inherits the same launch parameters as the source version, except for parameters that you specify in `launch-template-data`.

  - [--version-description `string`] – A description for the first version of the launch template.

  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).

- create-tags – Adds or overwrites one or more tags for the specified resource. Each resource can have a maximum of 50 tags. Each tag consists of a key and optional value. Tag keys must be unique for a resource. The following resources are supported:

  - AMI

  - Instance

- Launch template
- Security group
- create-security-group – Creates a security group on your Snowcone device. You can create up to 50 security groups. When you create a security group, you specify a friendly name of your choice:
  - --group-name `value` – The name of the security group.
  - --description `value` – A description of the security group. This is informational only. This value can be up to 255 characters in length.
- create-volume – Creates an Amazon EBS volume that can be attached to an instance on your AWS Snowcone device.
  - [--size `value`] – The size of the volume in GiBs, which can be from 1 GiB to 1 TB (1000 GiBs).
  - [--snapshot-id `value`] – The snapshot from which to create the volume.
  - [--volume-type `value`] – The volume type. If no value is specified, the default is `sbg1`. Possible values include the following:
    - `sbg1` for magnetic volumes
    - `sbp1` for SSD volumes
  - [--tag-specification `value` – A list of tags to apply to the volume during creation.
- delete-launch-template – Deletes a launch template. Deleting a launch template deletes all of its versions.

  Specify either the launch template ID or launch template name in the request.
  - --launch-template-id `string` – The ID of the launch template.
  - --launch-template-name `string` – A name for the launch template.
  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see .
- delete-launch-template-version – Deletes one or more versions of a launch template. You can't delete the default version of a launch template; you must first assign a different version as the default. If the default version is the only version for the launch template, delete the entire launch template by using the `delete-launch-template` command.

  Specify either the launch template ID or launch template name in the request.
  - --launch-template-id `string` – The ID of the launch template.
  - --launch-template-name `string` – A name for the launch template.
  - --versions (list) `"string" "string"` – The version numbers of one or more launch template versions to delete.
  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations.
- delete-security-group – Deletes a security group.

  If you attempt to delete a security group that is associated with an instance, or is referenced by another security group, the operation fails with `DependencyViolation`.
  - --group-name `value` – The name of the security group.
  - --description `value` – A description of the security group. This is informational only. This value can be up to 255 characters in length.
- delete-tags – Deletes the specified set of tags from the specified resource (AMI, compute instance, launch template, or security group).
- delete-volume – Deletes the specified Amazon EBS volume. The volume must be in the `available` state (not attached to an instance).
  - --volume-id `value` – The ID of the volume.

- **describe-addresses** – Describes one or more of your virtual IP addresses associated with the same number of `sbe` instances on your device.
  - --public-ips – One or more of the virtual IP addresses associated with your instances.
- **describe-images** – Describes one or more of the images (AMIs) available to you. Images available to you are added to the Snowcone device during job creation.
  - --image-id – The Snowcone AMI ID of the AMI.
- **describe-instance-attribute** – Describes the specified attribute of the specified instance. You can specify only one attribute at a time. The following attributes are supported:
  - `instanceInitiatedShutdownBehavior`
  - `instanceType`
  - `userData`
- **describe-instances** – Describes one or more of your instances. The response returns any security groups that are assigned to the instances.
  - --instance-ids – The IDs of one or more `sbe` instances that were stopped on the device.
  - --page-size – The size of each page to get in the call. This value doesn't affect the number of items returned in the command's output. Setting a smaller page size results in more calls to the device, retrieving fewer items in each call. Doing this can help prevent the calls from timing out.
  - --max-items – The total number of items to return in the command's output. If the total number of items available is more than the value specified, `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the `starting-token` argument of a subsequent command.
  - --starting-token – A token to specify where to start paginating. This token is the `NextToken` value from a previously truncated response.
- **describe-launch-templates** – Describes one or more launch templates. The `describe-launch-templates` command is a paginated operation. You can make multiple calls to retrieve the entire dataset of results.

  Specify either the launch template IDs or launch template names in the request.
  - --launch-template-ids (list) `"string"` `"string"` – A list of IDs of the launch templates.
  - --launch-template-names (list) `"string"` `"string"` – A list of names for the launch templates.
  - --page-size – The size of each page to get in the call. This value doesn't affect the number of items returned in the command's output. Setting a smaller page size results in more calls to the device, retrieving fewer items in each call. Doing this can help prevent the calls from timing out.
  - --max-items – The total number of items to return in the command's output. If the total number of items available is more than the value specified, `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the `starting-token` argument of a subsequent command.
  - --starting-token – A token to specify where to start paginating. This token is the `NextToken` value from a previously truncated response.
  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).
- **describe-launch-template-versions** – Describes one or more versions of a specified launch template. You can describe all versions, individual versions, or a range of versions. The `describe-launch-template-versions` command is a paginated operation. You can make multiple calls to retrieve the entire dataset of results.

  Specify either the launch template IDs or launch template names in the request.
  - --launch-template-id `string` – The ID of the launch template.
  - --launch-template-name `string` – A name for the launch template.

- [--versions (list) `"string"` `"string"`] – The version numbers of one or more launch template versions to delete.

- [--min-version `string`] – The version number after which to describe launch template versions.

- [--max-version `string`] – The version number up to which to describe launch template versions.

- --page-size – The size of each page to get in the call. This value doesn't affect the number of items returned in the command's output. Setting a smaller page size results in more calls to the device, retrieving fewer items in each call. Doing this can help prevent the calls from timing out.

- --max-items – The total number of items to return in the command's output. If the total number of items available is more than the value specified, `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the `starting-token` argument of a subsequent command.

- --starting-token – A token to specify where to start paginating. This token is the `NextToken` value from a previously truncated response.

- --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).

- describe-security-groups – Describes one or more of your security groups.

  The `describe-security-groups` command is a paginated operation. You can issue multiple API calls to retrieve the entire dataset of results.

  - [--group-name `value`] – The name of the security group.

  - [--group-id `value`] – The ID of the security group.

  - [--page-size `value`] – The size of each page to get in the AWS service call. This size doesn't affect the number of items returned in the command's output. Setting a smaller page size results in more calls to the AWS service, retrieving fewer items in each call. This approach can help prevent the AWS service calls from timing out. For usage examples, see Pagination in the *AWS Command Line Interface User Guide*.

  - [--max-items `value`] – The total number of items to return in the command's output. If the total number of items available is more than the value specified, `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the `starting-token` argument of a subsequent command. Don't use the `NextToken` response element directly outside of the AWS CLI. For usage examples, see Pagination in the *AWS Command Line Interface User Guide*.

  - [--starting-token `value`] – A token to specify where to start paginating. This token is the `NextToken` value from a previously truncated response. For usage examples, see Pagination in the *AWS Command Line Interface User Guide*.

- describe-tags – Describes one or more of the tags for specified resource (`image`, `instance`, or security group). With this command, the following filters are supported:

  - launch-template

  - resource-id

  - resource-type – `image` or `instance`

  - key

  - value

- describe-volumes – Describes the specified Amazon EBS volumes.

  - [--max-items `value`] – The total number of items to return in the command's output. If the total number of items available is more than the value specified, `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the `starting-token` argument of a subsequent command.

  - [--starting-token `value`] – A token to specify where to start paginating. This token is the `NextToken` value from a previously truncated response.

  - [--volume-ids `value`] – One or more volume IDs.

- detach-volume – Detaches an Amazon EBS volume from a stopped or running instance.

- [--device `value`] – The device name.
- [--instance-id] – The ID of a target Amazon EC2 instance.
- --volume-id `value` – The ID of the volume.
- disassociate-address – Disassociates a virtual IP address from the instance it's associated with.
  - --public-ip – The virtual IP address that you want to disassociate from your instance.
- get-launch-template-data – Retrieves the configuration data of the specified instance. You can use this data to create a launch template.
  - --instance-id – The ID of a single `sbe` instance.
  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).
- modify-launch-template – Modifies a launch template. You can specify which version of the launch template to set as the default version. When you launch an instance without specifying a launch template version, the default version of the launch template applies.

  Specify either the launch template ID or launch template name in the request.
  - --launch-template-id `string` – The ID of the launch template.
  - --launch-template-name `string` – A name for the launch template.
  - --default-version `string` – The version number of the launch template to set as the default version.
  - --endpoint `snowballEndpoint` – A value that enables you to manage your compute instances programmatically using Amazon EC2 API operations. For more information, see Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint (p. 85).
- modify-instance-attribute – Modifies an attribute of the specified instance. The following attributes are supported:
  - `instanceInitiatedShutdownBehavior`
  - `userData`
- revoke-security-group-egress – Removes one or more egress rules from a security group:
  - [--group-id `value`] – The ID of the security group
  - [--ip-permissions `value`] – One or more sets of IP permissions.
- revoke-security-group-ingress – Revokes one or more ingress rules to a security group. When calling `revoke-security-group-ingress`, you must specify a value for either `group-name` or `group-id`.
  - [--group-name `value`] – The name of the security group.
  - [--group-id `value`] – The ID of the security group.
  - [--ip-permissions `value`] – One or more sets of IP permissions.
  - [--protocol `value`] The IP protocol. Possible values are `tcp`, `udp`, and `icmp`. The `--port` argument is required unless the "all protocols" value is specified (-1).
  - [--port `value`] – For TCP or UDP, the range of ports to allow. A single integer or a range (minimum– maximum).

    For ICMP, a single integer or a range (`type-code`) in which `type` represents the ICMP type number and `code` represents the ICMP code number. A value of -1 indicates all ICMP codes for all ICMP types. A value of -1 just for `type` indicates all ICMP codes for the specified ICMP type.
  - [--cidr `value`] – The CIDR IP range.
- run-instances – Launches a number of compute instances by using a Snowcone AMI ID for an AMI.

    Note
    It can take up to an hour and a half to launch a compute instance on a Snowcone device, depending on the size and type of the instance.
  - [--block-device-mappings (`list`)] – The block device mapping entries. The parameters `DeleteOnTermination`, `VolumeSize`, and `VolumeType` are supported. Boot volumes must be type `sbg1`.

The JSON syntax for this command is as follows.

```
{
    "DeviceName": "/dev/sdh",
    "Ebs":
    {
        "DeleteOnTermination": true|false,
        "VolumeSize": 100,
        "VolumeType": "sbp1"|"sbg1"
    }
}
```

- --count – Number of instances to launch. If a single number is provided, it is assumed to be the minimum to launch (defaults to 1). If a range is provided in the form `min:max`, then the first number is interpreted as the minimum number of instances to launch and the second is interpreted as the maximum number of instances to launch.

- --image-id – The Snowcone AMI ID of the AMI, which you can get by calling `describe-images`. An AMI is required to launch an instance.

- --InstanceInitiatedShutdownBehavior – By default, when you initiate a shutdown from your instance (using a command such as shutdown or poweroff), the instance stops. You can change this behavior so that it terminates instead. The parameters `stop` and `terminate` are supported. The default is `stop`. For more information, see Changing the instance initiated shutdown behavior in the *Amazon EC2 User Guide for Linux Instances*.

- --instance-type – The `sbe` instance type.

- --launch-template `structure` – The launch template to use to launch the instances. Any parameters that you specify in the `run-instances` command override the same parameters in the launch template. You can specify either the name or ID of a launch template, but not both.

```
{
    "LaunchTemplateId": "string",
    "LaunchTemplateName": "string",
    "Version": "string"
}
```

- --security-group-ids – One or more security group IDs. You can create a security group using CreateSecurityGroup. If no value is provided, the ID for the default security group is assigned to created instances.

- --tag-specifications – The tags to apply to the resources during launch. You can only tag instances on launch. The specified tags are applied to all instances that are created during launch. To tag a resource after it has been created, use `create-tags`.

- --user-data – The user data to make available to the instance. If you are using the AWS CLI, base64-encoding is performed for you, and you can load the text from a file. Otherwise, you must provide base64-encoded text.

- start-instances – Starts an `sbe` instance that you've previously stopped. All resources attached to the instance persist through starts and stops, but are erased if the instance is terminated.

  - --instance-ids – The IDs of one or more `sbe` instances that were stopped on the device.

- stop-instances – Stops an `sbe` instance that is running. All resources attached to the instance persist through starts and stops, but are erased if the instance is terminated.

  - --instance-ids – The IDs of one or more `sbe` instances to be stopped on the device.

- terminate-instances – Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds. All resources attached to the instance persist through starts and stops, but data is erased if the instance is terminated.

> **Note**
> By default, when you use a command like `shutdown` or `poweroff` to initiate
> a shutdown from your instance, the instance stops. However, you can use the
> `InstanceInitiatedShutdownBehavior` attribute to change this behavior so that these
> commands terminate your instance. For more information, see Changing the instance initiated
> shutdown behavior in the *Amazon EC2 User Guide for Linux Instances*.

- --instance-ids – The IDs of one or more `sbe` instances to be terminated on the device. All associated data stored for those instances will be lost.

Supported Amazon EC2 API Operations

Following, you can find Amazon EC2 API operations that you can use with a Snowcone device, with links to their descriptions in the *Amazon EC2 API Reference.* Amazon EC2 API calls require Signature Version 4 (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.

- AssociateAddress – Associates an Elastic IP address with an instance or a network interface.
- AttachVolume – The following request parameters are supported:
  - `Device`
  - `InstanceId`
  - `VolumeId`
- AuthorizeSecurityGroupEgress – Adds one or more egress rules to a security group for use with a Snowcone device. Specifically, this action permits instances to send traffic to one or more destination IPv4 CIDR address ranges.
- AuthorizeSecurityGroupIngress – Adds one or more ingress rules to a security group. When calling AuthorizeSecurityGroupIngress, you must specify a value either for `GroupName` or `GroupId`.
- CreateVolume – The following request parameters are supported:
  - `SnapshotId`
  - `Size`
  - `VolumeType`
  - `TagSpecification.N`
- CreateLaunchTemplate – The following request parameters are supported:
  - `ImageId`
  - `InstanceType`
  - `SecurityGroupIds`
  - `TagSpecifications`
  - `UserData`
- CreateLaunchTemplateVersion
- CreateTags – The following request parameters are supported:
  - `AMI`
  - `Instance`
  - `Launch template`
  - `Security group`
- CreateSecurityGroup – Creates a security group on your Snowcone. You can create up to 50 security groups. When you create a security group, you specify a friendly name of your choice.
- DeleteLaunchTemplate
- DeleteLaunchTemplateVersions

- DeleteSecurityGroup – Deletes a security group. If you attempt to delete a security group that is associated with an instance, or is referenced by another security group, the operation fails with `DependencyViolation`.

- DeleteTags – Deletes the specified set of tags from the specified set of resources.

- DeleteVolume – The following request parameters are supported:

  - `VolumeId`

- DescribeAddresses

- DescribeImages

- DescribeInstanceAttribute – The following attributes are supported:

  - `instanceType`

  - `userData`

- DescribeLaunchTemplates

- DescribeLaunchTemplateVersions

- DescribeInstances

- DescribeSecurityGroups – Describes one or more of your security groups. `DescribeSecurityGroups` is a paginated operation. You can issue multiple API calls to retrieve the entire dataset of results.

- DescribeTags – With this command, the following filters are supported:

  - `resource-id`

  - `resource-type` – AMI or compute instance only

  - `key`

  - `value`

- DescribeVolume – The following request parameters are supported:

  - `MaxResults`

  - `NextToken`

  - `VolumeId.N`

- DetachVolume – The following request parameters are supported:

  - `Device`

  - `InstanceId`

  - `VolumeId`

- DisassociateAddress

- GetLaunchTemplateData

- ModifyLaunchTemplate

- ModifyInstanceAttribute – Only the `userData` attribute is supported.

- RevokeSecurityGroupEgress – Removes one or more egress rules from a security group.

- RevokeSecurityGroupIngress – Revokes one or more ingress rules to a security group. When calling RevokeSecurityGroupIngress, you must specify a value either for `group-name` or `group-id`.

- RunInstances –

  Note
  It can take up to an hour and a half to launch a compute instance on a Snowcone, depending on the size and type of the instance.

- StartInstances

- StopInstances – Resources associated with a stopped instance persist. You can terminate the instance to free up these resources. However, any associated data is deleted.

- TerminateInstances

Autostarting Amazon EC2 Instances with Launch Templates

You can automatically start your Amazon EC2 instances on your AWS Snowcone device using launch templates and Snowball Edge client launch configuration commands. If an instance exits, autostart will start it but If you delete the instance or update the autostart configuration of the instance, the autostart will start a new instance.

A *launch template* contains the configuration information necessary to create an Amazon EC2 instance on your Snowcone. You can use a launch template to store launch parameters so you don't have to specify them every time that you start an EC2 instance on your Snowcone.

When you use autostart configurations on your Snowcone, you configure the parameters that you want your Amazon EC2 instance to start with. After your Snowcone is configured, when you reboot and unlock it, it uses your autostart configuration to launch an instance with the parameters that you specified. If an instance that you launched using an autostart configuration is stopped, the instance starts running when you unlock your device.

> **Note**
> After you first configure an autostart configuration, restart your device to launch it. All subsequent instance launches (after planned or unplanned reboots) happen automatically after your AWS Snowcone device is unlocked.

A launch template can specify the Amazon Machine Image (AMI) ID, instance type, user data, security groups, and tags for an Amazon EC2 instance when you launch that instance.

To automatically launch EC2 instances on your Snowcone, take the following steps:

1. When you order your AWS Snowcone device, create a job with compute instances. For more information, see Creating a Job with Compute Instances (p. 75).
2. After receiving your Snowcone, unlock it.
3. Use the EC2 API command `aws ec2 create-launch-template` to create a launch template. For more information, see List of Supported Amazon EC2 AWS CLI Commands on a Snowcone (p. 86).
   > **Note**
   > The Amazon EC2 endpoint is the device endpoint.
4. Use the Snowball Edge client command `snowballEdge create-autostart-configuration` to bind your EC2 launch template to your network configuration. For more information, see Creating a Launch Configuration to Autostart Amazon EC2 Instances (p. 81).
5. Reboot, then unlock your AWS Snowcone device. Your EC2 instances are automatically started using the attributes specified in your launch template and your Snowcone client command `create-autostart-configuration`.

To view the status of your running instances, use the EC2 API command `describe-autostart-configurations`.
> **Note**
> There is no console or job management API for AWS Snowball support for launching templates. You use EC2 and Snowball Edge client CLI commands to automatically start EC2 instances on your AWS Snowcone device.

Using Block Storage with Your Amazon EC2 Instances

Block storage on Snowcone enables you to add or remove block storage based on the needs of your applications. Volumes that are attached to an Amazon EC2 instance are exposed as storage volumes that persist independently from the life of the instance. You can manage block storage using the familiar Amazon EBS API.

Certain Amazon EBS commands are supported by using the EC2 endpoint. Supported commands include `attach-volume`, `create-volume`, `delete-volume`, `detach-volume`, and `describe-volumes`. For more information on these commands, see List of Supported Amazon EC2 AWS CLI Commands on a Snowcone (p. 86).

> **Important**
> Be sure to unmount any file systems on the device within your operating system before detaching the volume. Failure to do so can potentially result in data loss.

Following, you can find Amazon EBS volume quotas and differences between Amazon EBS volumes on your AWS Snowcone device and Amazon EBS volumes in the cloud:

- Amazon EBS volumes are only available to EC2 instances running on the AWS Snowcone device hosting the volumes.
- Volume types are limited to either capacity-optimized HDD (`sbg1`) or >performance-optimized SSD (`sbp1`). The default volume type is `sbg1`.
- Amazon EC2 root volumes always use the IDE driver. Additional Amazon EBS volumes preferentially use the Virtio driver if available. If the Virtio driver isn't available, SBE defaults to the IDE driver. The Virtio driver allows for better performance and is recommended.
- When creating Amazon EBS volumes, the `encrypted` parameter isn't supported. However, all data on your device is encrypted by default. For more information, see .
- Volumes can be from 1 GB to 8 TB in size.
- Up to 10 Amazon EBS volumes can be attached to a single EC2 instance.
- There is no formal limit to the number of Amazon EBS volumes you can have on your AWS Snowcone device. However, total Amazon EBS volume capacity is limited by the available space on your AWS Snowcone device.

Security Groups in Snow Devices

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group to allow traffic to or from its associated instances. For more information, see Amazon EC2 security groups for Linux instances in the Amazon EC2 User Guide for Linux Instances.

Security groups in Snowcone devices are similar to security groups in the AWS Cloud. Virtual private clouds (VPCs) aren't supported on Snowcone devices.

Following, you can find the other differences between Snowcone security groups and EC2-VPC security groups:

- Each Snowcone has a limit of 50 security groups.
- The default security group allows all inbound and outbound traffic.
- Traffic between local instances can use either the private instance IP address or a public IP address. For example, suppose that you want to connect using SSH from instance A to instance B. In this case, your target IP address can be either the public IP or private IP address of instance B, if the security group rule allows the traffic.
- Only the parameters listed for AWS CLI actions and API calls are supported. These typically are a subset of those supported in EC2-VPC instances.

For more information about supported AWS CLI actions, see List of Supported Amazon EC2 AWS CLI Commands on a Snowcone (p. 86). For more information about supported API operations, see Supported Amazon EC2 API Operations (p. 93).

Supported Instance Metadata and User Data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Snowcone supports a subset of instance metadata categories for your compute instances. For more information, see Instance metadata and user data in the *Amazon EC2 User Guide for Linux Instances.*

The following categories are supported. Using any other categories returns a `404` error message.

Supported Instance Metadata Categories on a Snowcone

Data	Description
`ami-id`	The AMI ID used to launch the instance.
`hostname`	The private IPv4 DNS hostname of the instance.
`instance-id`	The ID of this instance.
`instance-type`	The type of instance.
`local-hostname`	The private IPv4 DNS hostname of the instance.
`local-ipv4`	The private IPv4 address of the instance.
`mac`	The instance's media access control (MAC) address.
`network/interfaces/macs/`*`mac`*`/local-hostname`	The interface's local hostname.
`network/interfaces/macs/`*`mac`*`/local-ipv4s`	The private IPv4 addresses associated with the interface.
`network/interfaces/macs/`*`mac`*`/mac`	The instance's MAC address.
`network/interfaces/macs/`*`mac`*`/public-ipv4s`	The Elastic IP addresses associated with the interface.
`public-ipv4`	The public IPv4 address.
`public-keys/0/openssh-key`	Public key. Only available if supplied at instance launch time.
`reservation-id`	The ID of the reservation.
userData	Shell scripts to send instructions to an instance at launch.

Supported Instance Dynamic Data Categories on a Snowcone

Data	Description
instance-identity/document	JSON containing instance attributes. Only `instanceId`, `imageId`, `privateIp`, and `instanceType` have values, and the other returned attributes are null. For more information, see Instance identity documents in the *Amazon EC2 User Guide for Linux Instances*.

Changing User Data in Snowcone Compute Instances

User data is supported for use with shell scripts for compute instances on a Snowcone device. Using shell scripts, you can send instructions to an instance at launch. You can change user data with the `modify-instance-attribute` AWS CLI command, or the `ModifyInstanceAttribute` API action.

To change user data

1. Stop your compute instance with the `stop-instances` AWS CLI command.

2. Using the `modify-instance-attribute` AWS CLI command, modify the `userData` attribute.

3. Restart your compute instance with the `start-instances` AWS CLI command.

Only shell scripts are supported with compute instances. There is no support for `cloud-init` package directives on compute instances running on a Snowcone. For more information about working with AWS CLI commands, see the AWS CLI Command Reference.

Troubleshooting Compute Instances on Snowcone Devices

Following, you can find troubleshooting tips for Snowcone jobs with compute instances.

Topics

- Virtual Network Interface Has an IP Address of 0.0.0.0 (p. 98)
- Snowcone Hangs When Launching a Large Compute Instance (p. 98)
- My Instance Has One Root Volume (p. 99)
- Unprotected Private Key File Error (p. 99)

Virtual Network Interface Has an IP Address of 0.0.0.0

This issue can occur if the physical network interface (NIC) you associated with your virtual network interface (VNIC) also has an IP address of 0.0.0.0. This effect can happen if the NIC wasn't configured with an IP address (for instance, if you've just powered on the device). It can also happen if you're using the wrong interface. For example, you might be trying to get the IP address of the SFP+ interface, but it's the RJ45 interface that's connected to your network.

Action to Take

If this occurs, you can do the following:

- Create a new VNIC, associated with a NIC that has an IP address. For more information, see Network Configuration for Compute Instances (p. 77).
- Update an existing VNIC. For more information, see Updating a Virtual Network Interface (p. 83).

Snowcone Hangs When Launching a Large Compute Instance

It can appear that your Snowcone has stopped launching an instance. This is generally not the case. However, it can take an hour or more for the largest compute instances to launch. You can check the status of your instances using the AWS CLI command `aws ec2 describe-instances` run against the HTTP or HTTPS Amazon EC2 endpoint on the Snowcone.

My Instance Has One Root Volume

Instances have one root volume by design. All `sbe` instances have a single root volume.

Unprotected Private Key File Error

This error can occur if your .pem file on your compute instance has insufficient read/write permissions.

Action to Take

You can resolve this by changing the permissions for the file with the following procedure:

1. Open a terminal and navigate to the location that you saved your .pem file to.
2. Enter the following command.

```
chmod 400 filename.pem
```

Using AWS DataSync to Transfer Files

AWS DataSync is an online data transfer service that simplifies, automates, and accelerates copying large amounts of data to and from AWS storage services over the internet or AWS Direct Connect. DataSync agent comes pre-installed on your Snowcone device. It can transfer data between the device and Amazon S3 buckets, Amazon EFS, and Amazon FSx for Windows File Server. AWS DataSync automatically handles moving files and objects, scheduling data transfers, monitoring the progress of transfers, encryption, verification of data transfers, and notifying customers of any issues.

The DataSync agent is pre-installed on your Snowcone device as an AMI during the Snowcone job preparation. To transfer data online to AWS, connect the Snowcone device to the external network and use AWS OpsHub or the CLI to launch the DataSync agent AMI. Activate the DataSync agent the AWS Management Console or use the CLI, and set up your online data transfer task between the Snowcone NFS store, and Amazon S3, Amazon EFS, or Amazon FSx for Windows File Server.

You can use AWS DataSync running on Snowcone for the following:

- Edge computing applications, to collect data, process the data to gain immediate insight, and then transfer the data online to AWS.
- Transfer data that is continuously generated by sensors or machines online to AWS in a factory or at other edge locations.
- Distribute media, scientific, or other content online from AWS storage services to your partners and customers.
- Aggregate content by transferring media, scientific or other content online from your edge locations to AWS.

For one-time edge compute or data transfer workflows or for Snowcone workflows in edge locations without a wide area network (WAN) link or inadequate WAN bandwidth, you should ship the Snowcone device back to AWS to complete the data transfer.

Using NFS for Offline Data Transfer

Your Snow Family device contains a file interface that provides access to the internal device storage. To import your data offline to Amazon S3 with your Snow Family device, you connect the device to your on-

premises network and then use AWS OpsHub to unlock it. You can copy data from on-premises storage devices to the Snow Family device through the NFS file interface.

After you copy the data to the Snow Family device, the E Ink shipping label will be updated to ensure that the device is automatically sent to the correct AWS facility. You can track the Snow Family device by using Amazon SNS generated text messages or emails, and the console. For information about AWS OpsHub, see Using AWS OpsHub for Snow Family to Manage Devices (p. 26).

> **Note**
> File names are object keys in your local S3 bucket on the Snow Family device. The key name is a sequence of Unicode characters whose UTF-8 encoding is at most 1,024 bytes long. We recommend using NFSv4.1 where possible and encode file names with Unicode UTF-8 to ensure a successful data import. File names that are not encoded with UTF-8 might not be uploaded to S3 or might be uploaded to S3 with a different file name depending on the NFS encoding you use.
> Ensure that the maximum length of your file path is less than 1024 characters. Snow Family Devices do not support file paths that are greater that 1024 characters. Exceeding this file path length will result in file import errors.
> For more information, see Object keys in the *Amazon Simple Storage Service User Guide*.

AWS starts exporting the data from your Amazon S3 buckets to your Snow Family device within one business day of receiving your export order. We prepare and ship the device to you, and you receive it in approximately 4-6 business days.

You can transfer up to 8 TB with a single Snowcone device and transfer larger datasets with multiple devices, either in parallel, or sequentially. For example, you can transfer 24 TB of data with three Snowcone devices. For larger data transfers jobs, you can use the Snowball Edge Storage Optimized device. You can transfer up to 80 TB with a single Snowball Edge Storage Optimized device and transfer larger datasets with multiple devices, either in parallel, or sequentially.

> **Note**
>
> - You can provide CIDR blocks for IP ranges that are allowed to mount the NFS shares exposed by the device. For example, `10.0.0.0/16`. If you don't provide allowed CIDR blocks, all mount requests will be denied. For details, see Restricting Access to NFS Shares When NFS is Running.
> - Be aware that data transferred through NFS is not encrypted in transit.
> - Other than the allowed hosts by CIDR blocks, your Snow Family device doesn't provide an authentication or authorization mechanism for the NFS shares.
> - For local compute jobs, the device doesn't ship with NFS configure on it so your data will not be imported into Amazon S3.

Troubleshooting NFS Issues

The following are errors you might encounter when using NFS on Snow Family devices.

I Get a DEACTIVATED error message

You get this message if you turn off your Snow Family device without first stopping the NFS service. The next time you start NFS, it might fail with a DEACTIVATED error message.

For example: Starting the service on your Snowball Edge.

```
snowballEdge start-service --service-id nfs --virtual-network-interface-arns
arn:aws:snowball-device:::interface/s.ni-84991da69040a7xxx
```

You can determine the status of the service by using the `describe-service` command.

```
snowballEdge describe-service --service-id nfs
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "DEACTIVATED"
  }
}
```

How To Correct the Issue

To correct the problem, stop and restart the NFS service using the following steps.

Step 1: Use the `describe-service` command to determine the status of the service:

```
snowballEdge describe-service --service-id nfs
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "DEACTIVATED"
  }
}
```

Step 2: Use the `stop-service` command to stop the NFS service:

```
snowballEdge stop-service --service-id nfs --profile 11
```

Step 3: Use the `start-service` command to start the NFS service normally:

```
snowballEdge start-service  --virtual-network-interface-arns arn:aws:snowball-
device:::interface/s.ni-8712e3a5cb180e65d --service-id nfs  --service-configuration
 AllowedHosts=0.0.0.0/0  --profile 11
```

Step 4: Use the `describe-service` command to make sure the service is ACTIVE:

snowballEdge describe-service --service-id nfs

```
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "ACTIVE"
  },
  "Endpoints" : [ {
    "Protocol" : "nfs",
    "Port" : 2049,
    "Host" : "192.168.1.123"
  } ]
}
```

Returning the Snowcone Device

When you've finished transferring data on to the Snowcone device, prepare it for its return trip to AWS. Before you continue, make sure that all data transfer to the device has stopped.

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

Disconnect the Snowcone Device

Disconnect the Snowcone cables.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 device, it's ready to be returned. To see who your region's carrier is, see Shipping Considerations for AWS Snowcone (p. 111).

Job-Type Specific Consideration

Important
If you are importing data, don't delete your local copies of the transferred data until the import to AWS is successful at the end of the process and you can verify the results of the data transfer.

For information about to ship the device, see Shipping Considerations for AWS Snowcone (p. 111).

Protecting Data on Your Device

Consider the following recommendations to help protect the data on your AWS Snowcone device.

Topics

Securing Your AWS Snowcone

Following are some security points that we recommend you consider when using Snowcone, in addition to some high-level information on other security precautions that we take when a device arrives at AWS for processing.

We recommend the following security approaches:

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

    Note
    The Snowcone device is the property of AWS. Tampering with a device 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 file interface, object metadata is persisted.
- When a device 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). Snowcone 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.
- After the data transfer job has been processed and verified, AWS performs a software erasure of the Snowcone device following the National Institute of Standards and Technology (NIST) guidelines for media sanitization.

Validating NFC Tags

AWS Snowcone devices have NFC tags built into them. You can scan these tags with the Snowcone Verification App, available on Android. Scanning and validating these NFC tags can help you verify that your device has not been tampered with before you use it.

Validating NFC tags includes using the Snowball Edge client to generate a device-specific QR code to verify that the tags you're scanning are for the right device. For information, see Getting Your QR Code for NFC Validation (p. 44).

The following procedure describes how to validate the NFC tags on a Snowcone device. Before you get started, make sure you've performed the following first steps of the getting started exercise:

1. Create your first job. For more information, see Creating an AWS Snowcone Job.
2. Receive the device.
3. Connect to your local network.
4. Get your credentials and tools. For more information, see Getting Credentials (p. 47).
5. Download and install the Snowball Edge client. For more information, see Using the Snowball Edge Client (p. 42).

To validate the NFC tags in an AWS Snowcone device

1. Run the `snowballEdge get-app-qr-code` Snowball Edge client command. For more information on using this command, see Getting Your QR Code for NFC Validation (p. 44).

   The QR code is saved to a location of your choice as a .png file.
2. Navigate to the .png file that you saved, and open it so that you can scan the QR code with the app.
3. To scan the NFC tags with your phone, download and install the Snowcone Verification App. Download the app from the Google Play store if you are using an Android phone.
4. Start the app, and follow the on-screen instructions.

You've now successfully scanned and validated the NFC tags for your device.

If you encounter issues while scanning, try the following:

- Download the app on another phone, and try again.
- Move the device to an isolated area of the room, away from interference from other NFC tags, and try again.
- If issues persist, contact AWS Support.

Understanding AWS Snowcone Job Statuses

When you create an AWS Snowcone job, it transitions through the job statuses and status is shown on the AWS Snow Family Management Console.

To see the status of a job

1. Log into the AWS Snow Family Management Console.
2. On the **Job dashboard**, choose the job.
3. Use the arrow on the left side of the job name to expand the job.
4. The status of the job is shown on the status bar.

AWS Snowcone device job statuses

Job Status	Meaning
Job created	Your job has just been created. This status is the only one during which you can cancel a job or its job parts, if the job is an export job.
Preparing appliance	AWS is preparing a device for your job.
Preparing shipment	AWS is preparing to ship a device to you.
In transit to you	The device has been shipped to the address you provided during job creation.
Delivered to you	The device has arrived at the address you provided during job creation.
In transit to AWS	You have shipped the device back to AWS.
At sorting facility	The device for this job is at our internal sorting facility. Any additional processing for import jobs into Amazon S3 will begin soon, typically within 2 days.
At AWS	Your shipment has arrived at AWS. If you're importing data, your import typically begins within a day of its arrival.
Importing	AWS is importing your data into Amazon S3.
Completed	Your job or a part of your job has completed successfully.
Canceled	Your job has been canceled.

Understanding the AWS Snowcone Ordering Process

There are two options when ordering an AWS Snowcone. You can order through the AWS Snow Family Management Console or you can use the job management API (JMAPI).

Understanding the Shipping Process

In this section you will find information about how shipping is handled for an AWS Snowcone device, and a list AWS Regions that are supported. For information about supported Regions and endpoints, see AWS Snow Family endpoints and quotas in the *AWS General Reference*. The shipping rate you choose for a job applies to sending and receiving the Snowcone device used for that job. For information about shipping charges, see AWS Snowcone pricing.

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

Shipping a Snowcone Device

The prepaid shipping label contains the correct address to return the device. For information about how to return your Snowcone device, see Shipping Carriers (p. 112). The Snowcone device 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 Snow Family Management Console.

> **Important**
> Unless personally instructed otherwise by AWS, don't affix a separate shipping label to the Snowcone device. Always use the shipping label that is displayed on the device's E Ink display.

Using the AWS Management Console

You can order a Snowcone device using the AWS Snow Family Management Console.

Ordering the Snowcone from the Console

For step-by step instructions on how to order a Snowcone using the AWS Snowball console, see Getting Started (p. 20).

Using the Job Management API

The job management API (JMAPI) provides programmatic access to the same functionality available in the AWS Snow Family Management Console. This enables you to automate job functionality. By using the JMAPI, you can see the job status, create jobs, download the manifest file, unlock code, and view job completion reports. Because the calls are made through the API, you can integrate these calls into a custom application or web front end.

Topics

Common Uses of JMAPI

- Automating ordering of Snowcone devices
- Downloading the manifest file
- Downloading the unlock file
- Listing out the current Snowcone jobs
- Downloading the Snowcone job completion report

JMAPI Required Strings

When placing an order through the job management API, you use the following required parameters, which are shown with examples.

```
--job-type
--resources
--address-id
--region
--role-arn
--kms-key-arn
--shipping-option
--device-type
--description
```

JMAPI Endpoints

API Endpoint

To make calls to each endpoint, the format is `snowballEdge.`*`region`*`.amazonaws.com`. Following are some examples to help you understand the breakdown of the endpoint.

Example

Region	Endpoint
US East (N. Virginia)	snowball.us-east-1.amazonaws.com
US West (Oregon)	snowball.us-west-2.amazonaws.com

JMAPI CLI Commands

Job Management CLI

The following are the CLI calls that you can make against the job management API.

Command	Example
Listing Jobs	`aws snowball list-jobs`
Describe Job	`aws snowball describe-job --job-id` *[JOB ID]*
Describe Address	`aws snowball describe-address --address-id`
Create Address	`aws snowball create-address --cli-input-json file://create-address.json`
Create Job	`aws snowball create-job --cli-input-json file://create-job.json`
Cancel Job	`aws snowball cancel-job --job-id` *[JOB ID]*

Examples

The following are examples of commands using the job management API.

KMS JSON Example

The following JSON example is a properly formatted JSON file for using the AWS KMS policy file.

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333"
    }
}
```

Create Address Example

The following examples show you how you would format the command to create your address and what the response is when it is successful.

```
aws snowball create-address --address "Name=Bob,Company=AWS,Street1=1234 Fake
 St.,City=All,StateOrProvince=Any,Country=US,PostalCode=12345,PhoneNumber=1234567890"
```

Example Output

```
{
```

```
        "AddressId": "ADID3be640c8-1111-1111-1111-917f201ffa42"
}
```

Create Job Example

The following command shows you an example command for running the `create-job` command.

```
aws snowball create-job --job-type EXPORT --resources file://snowball --address-id
 ADID3be640c8-1111-1111-1111-917f201ffa42 --region us-west-2 --role-arn role arn --kms-key-
arn arn:aws:kms:us-west-2:000000000000:key/Example --shipping-option SECOND_DAY --snowball-
type SNOWCONE
```

Exporting an AMI to use with Amazon EC2 Jobs

This section provides an overview of how to export your Amazon Machine Image (AMI) for use with Amazon EC2 compute instances on an AWS Snowcone device.

Topics
- Configuring an AMI to Use SSH to Connect to Compute Instances Launched on the Device (p. 109)
- Creating Your Job Using the Console (p. 110)
- Creating Your Job Using the AWS CLI (p. 110)

Configuring an AMI to Use SSH to Connect to Compute Instances Launched on the Device

To use Secure Shell (SSH) to connect to your compute instances on Snowcone devices, you must perform the following procedure. This procedure adds the SSH key to the AMI before creating your job. We also recommend that you use this procedure to set up your applications on the instance that you plan to use as the AMI for your job.

> **Important**
> If you don't follow this procedure, you can't connect to your instances with SSH when you receive your Snowcone device.

To put an SSH key into an AMI

1. Launch a new instance in the AWS Cloud based on the CentOS 7 (x86_64) - with Updates HVM, or Ubuntu 16.04 LTS - Xenial (HVM) image.

    When you launch your instance, make sure that the storage size that you assign to the instance is appropriate for your later use on the Snowcone device. In the Amazon EC2 console, you do this in **Step 4: Add Storage**. For a list of the supported sizes for compute instance storage volumes on a Snowcone, see "ec2-snowcone-limits".

2. Install and configure the applications that you want to run on the Snowcone, and test that they work as expected.

3. Make a copy of the PEM/PPK file that you used for the SSH key pair to create this instance. Save this file to the server that you plan to use to communicate with the Snowcone. This file is required for using SSH to connect to the launched instance on your device, so make a note of the path to this file.

4. Save the instance as an AMI. For more information, see Amazon Linux 2 for Snow Family Creating an Amazon EBS-Backed Linux AMI in the *Amazon EC2 User Guide for Linux Instances.*

5. Repeat this procedure for each of the instances that you want to connect to using SSH. Make sure that you make copies of your different SSH key pairs and take note of the AMIs they're associated with.

Creating Your Job Using the Console

Your next step is to create a job. Your job can be of any job type, including a cluster. Using the AWS Snow Family Management Console, follow the instructions provided in see Creating an AWS Snowcone Job. When you get to the **Step 3: Give job details** page in the job creation wizard, add the following additional steps.

1. Choose **Enable compute with EC2**.
2. Choose **Add an AMI**.
3. In the dialog box that opens, choose an AMI and choose **Save**.
4. Add up to 10 total AMIs to your job.
5. Continue creating your job as normal.

Creating Your Job Using the AWS CLI

You can also create your job using the AWS Command Line Interface (AWS CLI). To do this, open a terminal and run the following command, replacing the red text with your actual values.

```
aws snowball create-job --job-type IMPORT --resources '{"S3Resources":
[{"BucketArn":"arn:aws:s3:::bucket-name"}],"Ec2AmiResources":[{"AmiId":"ami-12345678"}]}'
 --description Example --address-id ADIEXAMPLE60-1234-1234-5678-41fEXAMPLE57  --kms-key-
arn arn:aws:kms:us-west-2:012345678901:key/eEXAMPLE-1234-1234-5678-5b4EXAMPLE8e --role-
arn  arn:aws:iam::account id:role/snowcone-import-snowcone-us-east-1-role   --shipping-
option SECOND_DAY --snowball-type SNOWCONE
```

After the device arrives and you unlock your device, use the Snowball Edge client to get your local credentials. For more information, see Getting Credentials (p. 47).

Shipping Considerations for AWS Snowcone

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

Topics

When you create a job, you specify a shipping address and shipping speed. This shipping speed doesn't indicate how soon you can expect to receive the AWS Snowcone device from the day you created the job. It only shows the time that the device 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). Also, carriers generally only pick up outgoing AWS Snowcone devices once a day. Thus, processing before shipping can take a day or more.

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

Preparing an AWS Snowcone Device for Shipping

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

To prepare an AWS Snowcone device for shipping

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

   > **Note**
   > If you've powered off and unplugged the AWS Snowcone device, and your shipping label doesn't appear after about a minute on the E Ink display on top of the device, contact AWS Support.

Region-Based Shipping Restrictions

Before you create a job, you should sign in to the console from the AWS Region that your data resides. Some shipping restrictions apply, as follows:

- For data transfers in US Regions, we don't ship AWS Snowcone devices outside of the United States.
- We don't ship AWS Snowcone devices between non-US Regions—for example, from Europe (Ireland) to Europe (Frankfurt), or from Asia Pacific (Mumbai) to Asia Pacific (Sydney).

> **Note**
> AWS doesn't ship AWS Snowcone devices to post office boxes.

Shipping an AWS Snowcone Device

The prepaid shipping label contains the correct address to return the AWS Snowcone device. For information about how to return your AWS Snowcone device, see Shipping Carriers (p. 112). The AWS Snowcone device 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 Snow Family Management Console.

> **Important**
> Unless personally instructed otherwise by AWS, don't affix a separate shipping label to the AWS Snowcone device. Always use the shipping label that is displayed on the AWS Snowcone device E Ink display.

Shipping Carriers

When you create a job, you provide the address that you want the AWS Snowcone device shipped to. The carrier that supports your region handles the shipping of AWS Snowcone devices from AWS to you, and back to AWS. When an AWS Snowcone device is shipped, you get a tracking number. You can find each job's tracking number and a link to the tracking website on the AWS Snow Family Management Console job dashboard, or by using API calls to the job management API.

Following is the list of supported carriers for AWS Snowcone devices by region:

- For all supported regions, UPS is the carrier.

AWS Snowcone Pickups in the US

In the US, keep the following information in mind for UPS to pick up an AWS Snowcone device:

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

AWS Snowcone Pickups in Australia

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

- **Job ID** – The job ID associated with the AWS Snowcone device that you want returned to AWS.
- **AWS account ID** – The ID for the AWS account that created the job.

- **Earliest Pickup Time** (your local time) – The earliest time of day that you want the AWS Snowcone device picked up.
- **Latest Pickup Time** (your local time) – The latest time of day that you want the AWS Snowcone device picked up.
- **Special Instructions** (optional) – Any special instructions for picking up the AWS Snowcone device.

Soon, you will get a follow-up email from the Snowcone team with more information about your pending pickup at the address your AWS Snowcone device was originally delivered to.

> **Note**
> The device takes 6 to 8 days to arrive at AWS.

Shipping Speeds

Each country has different shipping speeds available. These shipping speeds are based on the country in which you're shipping an AWS Snowcone device. Shipping speeds are as follows:

- **United States of America (US) and Canada** – When shipping in the US, you have access to one-day shipping and two-day shipping.
- **European Union (EU)** – When shipping to any of the countries within the EU, you have access to express shipping. Typically, Snow Family Devices 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.
- **Australia** – When shipping to Australia, you have access to the following shipping speeds:
  - Sydney – 5 days
  - Melbourne Brisbane – 5 days
  - Perth – 8 days

Best Practices for the AWS Snowcone Device

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

Topics

Security

- If you notice anything that looks suspicious about the AWS Snowcone device, don't connect it to your internal network. Instead, contact AWS Support, and a new AWS Snowcone device 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 Snowcone device. For example, you can save a copy of the manifest to your local server, and email the code to a user that unlocks the device. This approach limits access to the AWS Snowcone device 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 Edge client command `snowballEdge list-access-keys` followed by `snowballEdge get-secret-access-key` 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 device. 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.
- You can restrict access to NFS shares. For details, see Restricting Access to NFS Shares When NFS is Running (p. 54).
- When you turn off or power cycle a Snowcone device, it goes into a locked state.

Network

- We recommend that you only use one method of reading and writing data to a local bucket on an AWS Snowcone device at a time. Using both the file interface and the DataSync on the same S3 bucket at the same time can result in read/write conflicts.
- To prevent corrupting your data, don't disconnect an AWS Snowcone device or change its network settings while transferring data.
- Files should be in a static state while being written to the device. Files that are modified while they are being written can result in read/write conflicts.
- For more information about improving performance of your AWS Snowcone device, see Snowcone Performance (p. 116).

Resource Management

- The five free days for performing your on-premises data transfer start the day after the AWS Snowcone device arrives at your data center.

Managing EC2 Instances

To avoid accidentally deleting the Amazon EC2 instances that you create on your AWS Snowcone device, don't shut down your instances from the operating system. For example, don't use the `shutdown` or `reboot` commands. Shutting down an instance from within the operating system has the same effect as calling the terminate-instances command.

Instead, use the stop-instances command to suspend Amazon EC2 instances that you want to preserve.

Snowcone Performance

The following table outlines how your network's transfer rate impacts how long it takes to fill an AWS Snowcone with data.

Rate (MB/s)	8 TB Transfer Time	4 TB Transfer Time
100	21.17 hours	10.59 hours
60	36.57 hours	18.29 hours
30	68.57 hours	34.29 hours
10	210.29 hours	105.15 hours

AWS Snowcone Quotas

Following, you can find information about the quotas for using your Snowcone device.

Snowcone Disk Storage

Internally, your Snowcone device contains 8 TB or 14 TB of disk storage that you can use with the internal Network File System (NFS) service or with local Amazon EC2 instances through a local Amazon Elastic Block Store (Amazon EBS) volume presentation. You can use this storage for either NFS or Amazon EBS, but not both. You can allocate your storage depending on your use case. Be aware that the storage is pre-allocated when you place your Snowcone order.

Compute Job Storage

If the job type is local compute, you might create a total of 8 TB or 14 TB of local Amazon EBS volumes and attach them to Amazon EC2 instances. Using Amazon EBS volumes allows the local Amazon EC2 instances to access more local capacity than the root volume alone. Because this is local storage only, data written to the Amazon EBS volumes is lost when the device is returned to AWS because it can't be imported into Amazon S3.

Topics

Compute Resources Quotas

Available Compute Resources for Snowcone Devices Quotas for Storage

The following table outlines the available compute resources for Snowcone devices and their quotas for storage.

Instance type	vCPU cores	Memory (GiB)	Quota
snc1.micro	1	1	2
snc1.small	1	2	2
snc1.medium	2	4	1

Amazon Machine Image (AMI) and Amazon EC2 Capacity Quotas for Snowcone Devices

The following table outlines the AMI and Amazon EC2 instance capacity quotas for a Snowcone device.

Amazon EC2	Size
Amazon EC2 AMIs	125 GB (the combined size of all the AMIs used in a customer's job)
Amazon EC2 instances	150 GB (represents the space available for the backing storage volumes for the instances)

Limitations for Shipping a Snowcone Device

The following are the limitations for shipping a Snowcone device:

- AWS will not ship a Snowcone device to a post office box.
- Moving a Snowcone device to an address outside of the country specified when the job was created is not allowed and is a violation of the AWS Service Terms.

For more information about shipping, see .

Limitations on Processing Your Returned Snowcone Device for Import

To import your data into the AWS Cloud, the Snowcone device must meet the following requirements:

- The Snowcone device 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 Snowcone device for any reason.
- The device must not be physically damaged. To prevent damage, close the three doors on the Snowcone device and press until the latches make an audible clicking sound.
- Unless a shipping label is provided by AWS, the E Ink display on the Snowcone device must be visible, and it must show the return label that was automatically generated when you finished transferring your data to the Snowcone device.

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

Available AWS Regions

For information about the supported AWS Regions and endpoints, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.

Job Management API Reference

- Job Management API Reference
  - Actions
  - Data Types
  - Common Parameters
  - Common Errors

Document History for AWS Snowcone User Guide

The following table describes the documentation for this release of AWS Snowcone.

- **API version: latest**
- **Latest documentation update:** September 29th, 2021

update-history-change	update-history-description	update-history-date
New AWS Region supported (p. 120)	AWS Snowcone is now available in the US West (N. California), US East (Ohio) and South America (São Paulo) Regions. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	September 29, 2021
New AWS Region supported (p. 120)	AWS Snowcone is now available in the Asia Pacific (Singapore) and Asia Pacific (Tokyo) Regions. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	August 26, 2021
Support for offline data export from Amazon S3 using Snowcone devices (p. 120)	You can now request AWS to export your Amazon S3 data by transferring it to Snowcone devices which are then physically shipped to your location. For more information, see How Import and Export Jobs Work.	August 4, 2021
Introducing AWS Snow Device Management (p. 120)	Snow Device Management allows you to manage your AWS Snowcone device and local AWS services remotely. All Snowcone devices support Snow Device Management, and it comes preinstalled on new devices in most AWS Regions where Snowcone is available. For more information, see Using AWS Snow Device Management to Manage Devices.	August 4, 2021
New AWS Region supported (p. 120)	AWS Snowcone is now available in the Canada (Central) Region. For more information, see AWS Snow Family endpoints and	April 28, 2021

	quotas in the *AWS General Reference*.	
New AWS Region supported (p. 120)	AWS Snowcone is now available in the Asia Pacific (Sydney) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	March 24, 2021
Support for direct network interface (p. 120)	AWS Snowcone now adds direct network interface (DNI) configuration, an advanced network feature that enables use cases like multicast streams, routing, and load balancing. For more information, see Network Configuration for Compute Instances.	January 12, 2021
New AWS Region supported (p. 120)	AWS Snowcone is now available in the Europe (Frankfurt) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	November 18, 2020
New AWS Region supported (p. 120)	AWS Snowcone is now available in the Europe (Ireland) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	September 16, 2020
Introducing AWS Snowcone (p. 120)	AWS Snowcone is a portable, rugged, and secure device for edge computing and data transfer. You can use AWS Snowcone to collect, process, and move data to AWS, either offline by shipping the device to AWS, or online using AWS DataSync. For more information, see What Is AWS Snowcone?	June 17, 2020

AWS glossary

Numbers and symbols

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

A

AAD	See additional authenticated data.
Access Analyzer	A feature of AWS Identity and Access Management (IAM) (p. 135) that helps you identify the resources in your organization and accounts, such as Amazon S3 buckets or IAM roles that are shared with an external entity. See Also https://aws.amazon.com/about-aws/whats-new/2019/12/introducing-aws-identity-and-access-management-access-analyzer/.
access control list (ACL)	A document that defines who can access a particular bucket (p. 140) or object. Each bucket (p. 140) and object in Amazon S3 (p. 129) has an ACL. The document defines what each type of user can do, such as write and read permissions.
access identifiers	See credentials.
access key	The combination of an access key ID (p. 122) (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (p. 171) (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). You use access keys to sign API requests that you make to AWS.
access key ID	A unique identifier that's associated with a secret access key (p. 171); the access key ID and secret access key are used together to sign programmatic AWS requests cryptographically.

access key rotation	A method to increase security by changing the AWS access key ID. You can use this method to retire an old key at your discretion.
access policy language	A language for writing documents (specifically, *policies* (p. 164)) that specify who can access a particular AWS resource (p. 168) and under what conditions.
account	A formal relationship with AWS that's associated with all of the following:

- The owner email address and password
- The control of resources created under its umbrella
- Payment for the AWS activity related to those resources

The AWS account has permission to do anything and everything with all the AWS account resources. This is in contrast to a user (p. 178), which is an entity contained within the account.

account activity	A webpage showing your month-to-date AWS usage and costs. The account activity page is located at https://aws.amazon.com/account-activity/.
ACL	See access control list (ACL).
ACM	See AWS Certificate Manager (ACM).
ACM PCA	See AWS Certificate Manager Private Certificate Authority (ACM PCA).
ACM Private CA	See AWS Certificate Manager Private Certificate Authority (ACM PCA).
action	An API function. Also called *operation* or *call*. The activity the principal (p. 165) has permission to perform. The action is B in the statement "A has permission to do B to C where D applies." For example, Jane sends a request to Amazon SQS (p. 129) with Action=ReceiveMessage.

Amazon CloudWatch (p. 124): The response initiated by the change in an alarm's state (for example, from `OK` to `ALARM`). The state change might be caused by a metric reaching the alarm threshold, or by a `SetAlarmState` request. Each alarm can have one or more actions assigned to each state. Actions are performed once each time the alarm changes to a state that has an action assigned, such as an Amazon Simple Notification Service (p. 129) notification, the running of an Amazon EC2 Auto Scaling (p. 125) policy (p. 164), or an Amazon EC2 (p. 125) instance (p. 154) stop/terminate action.

active trusted key groups	A list showing each of the trusted key groups (p. 177), and the IDs of the public keys in each key group, that are active for a distribution in Amazon CloudFront. CloudFront can use the public keys in these key groups to verify the signatures of CloudFront signed URLs and signed cookies.
active trusted signers	See active trusted key groups (p. 123).
additional authenticated data	Information that's checked for integrity but not encrypted, such as headers or other contextual metadata.
administrative suspension	Amazon EC2 Auto Scaling (p. 125) might suspend processes for Auto Scaling group (p. 132) that repeatedly fail to launch instances. Auto Scaling groups that most commonly experience administrative suspension have zero running instances, have been trying to launch instances for more than 24 hours, and have not succeeded in that time.
alarm	An item that watches a single metric over a specified time period and starts an Amazon SNS (p. 129) topic (p. 177) or an Amazon EC2 Auto Scaling (p. 125)

	policy (p. 164) if the value of the metric crosses a threshold value over a predetermined number of time periods.
allow	One of two possible outcomes (the other is deny (p. 146)) when an IAM (p. 135) access policy (p. 164) is evaluated. When a user makes a request to AWS, AWS evaluates the request based on all permissions that apply to the user and then returns either allow or deny.
Amazon API Gateway	A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. See Also https://aws.amazon.com/api-gateway.
Amazon AppStream 2.0	A fully managed, secure service for streaming desktop applications to users without rewriting those applications. See Also https://aws.amazon.com/appstream/.
Amazon Athena	An interactive query service that makes it easy to analyze data in Amazon S3 using ANSI SQL. Athena is serverless, so there's no infrastructure to manage. Athena scales automatically and is simple to use, so you can start analyzing your datasets within seconds. See Also https://aws.amazon.com/athena/.
Amazon Aurora	A fully managed MySQL-compatible relational database engine that combines the speed and availability of commercial databases with the simplicity and cost-effectiveness of open-source databases. See Also https://aws.amazon.com/rds/aurora/.
Amazon Chime	A secure, real-time, unified communications service that transforms meetings by making them more efficient and easier to conduct. See Also https://aws.amazon.com/chime/.
Amazon Cloud Directory (Cloud Directory)	A service that provides a highly scalable directory store for your application's multihierarchical data. See Also https://aws.amazon.com/cloud-directory/.
Amazon CloudFront	An AWS content delivery service that helps you improve the performance, reliability, and availability of your websites and applications. See Also https://aws.amazon.com/cloudfront.
Amazon CloudSearch	A fully managed service in the AWS Cloud that makes it easy to set up, manage, and scale a search solution for your website or application.
Amazon CloudWatch	A web service that you can use to monitor and manage various metrics, and configure alarm actions based on data from those metrics. See Also https://aws.amazon.com/cloudwatch.
Amazon CloudWatch Events	A web service that you can use to deliver a timely stream of system events that describe changes in AWS resource (p. 168)s to AWS Lambda (p. 136) functions, streams in Amazon Kinesis Data Streams (p. 127), Amazon Simple Notification Service (p. 129) topics, or built-in targets. See Also https://aws.amazon.com/cloudwatch.
Amazon CloudWatch Logs	A web service for monitoring and troubleshooting your systems and applications from your existing system, application, and custom log files. You can send your existing log files to CloudWatch Logs and monitor these logs in near-real time. See Also https://aws.amazon.com/cloudwatch.
Amazon Cognito	A web service that makes it easy to save mobile user data, such as app preferences or game state, in the AWS Cloud without writing any backend

	code or managing any infrastructure. Amazon Cognito offers mobile identity management and data synchronization across devices. See Also https://aws.amazon.com/cognito/.
Amazon Comprehend	A natural language processing (NLP) service that uses machine learning to find insights and relationships in text. See Also https://aws.amazon.com/comprehend/.
Amazon Comprehend Medical	A HIPAA-eligible natural language processing (NLP) service that uses machine learning to extract health data from medical text. See Also https://aws.amazon.com/comprehend/medical/.
Amazon Connect	A service solution that offers easy, self-service configuration and provides dynamic, personal, and natural customer engagement at any scale. See Also https://aws.amazon.com/connect/.
Amazon Corretto	A no-cost, multiplatform, production-ready distribution of the Open Java Development Kit (OpenJDK). See Also https://aws.amazon.com/corretto/.
Amazon Detective	A service that collects log data from your AWS resources to analyze and identify the root cause of security findings or suspicious activities. The Detective behavior graph provides visualizations to help you to determine the nature and extent of possible security issues and conduct an efficient investigation. See Also https://aws.amazon.com/detective/.
Amazon DocumentDB (with MongoDB compatibility)	A managed database service that you can use to set up, operate, and scale MongoDB-compatible databases in the cloud. See Also https://aws.amazon.com/documentdb/.
Amazon DynamoDB	A fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. See Also https://aws.amazon.com/dynamodb/.
Amazon DynamoDB Encryption Client	A software library that helps you protect your table data before you send it to Amazon DynamoDB (p. 125).
Amazon DynamoDB Storage Backend for Titan	A storage backend for the Titan graph database implemented on top of Amazon DynamoDB. Titan is a scalable graph database optimized for storing and querying graphs. See Also https://aws.amazon.com/dynamodb/.
Amazon DynamoDB Streams	An AWS service that captures a time-ordered sequence of item-level modifications in any Amazon DynamoDB table, and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near real time. See Also https://aws.amazon.com/dynamodb/.
Amazon EBS-backed AMI	A type of Amazon Machine Image (AMI) (p. 128) whose instance (p. 154)s use an Amazon EBS (p. 126) volume (p. 179) as their root device. Compare this with instances launched from instance store-backed AMI (p. 155)s, which use the instance store (p. 155) as the root device.
Amazon EC2	A web service for launching and managing Linux/UNIX and Windows Server instance (p. 154)s in Amazon's data centers. See Also Amazon Elastic Compute Cloud (Amazon EC2), https://aws.amazon.com/ec2.
Amazon EC2 Auto Scaling	A web service designed to launch or terminate instance (p. 154)s automatically based on user-defined policies (p. 164), schedules, and health check (p. 153)s.

See Also https://aws.amazon.com/ec2/autoscaling.

Amazon Elastic Block Store (Amazon EBS)	A service that provides block level storage volume (p. 179)s for use with EC2 instance (p. 148)s. See Also https://aws.amazon.com/ebs.
Amazon Elastic Compute Cloud (Amazon EC2)	A web service for launching and managing Linux/UNIX and Windows Server instance (p. 154)s in Amazon's data centers. See Also https://aws.amazon.com/ec2.
Amazon Elastic Container Registry (Amazon ECR)	A fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon Elastic Container Service (Amazon ECS) (p. 126) and AWS Identity and Access Management (IAM) (p. 135). See Also https://aws.amazon.com/ecr.
Amazon Elastic Container Service (Amazon ECS)	A highly scalable, fast, container (p. 143) management service that makes it easy to run, stop, and manage Docker containers on a cluster (p. 142) of EC2 instance (p. 148)s. See Also https://aws.amazon.com/ecs.
Amazon Elastic File System (Amazon EFS)	A file storage service for EC2 (p. 125) instance (p. 154)s. Amazon EFS is easy to use and provides a simple interface with which you can create and configure file systems. Amazon EFS storage capacity grows and shrinks automatically as you add and remove files. See Also https://aws.amazon.com/efs/.
Amazon Elastic Kubernetes Service (Amazon EKS)	A managed service that simplifies running Kubernetes on AWS without your needing to stand up or maintain your own Kubernetes control plane. See Also https://aws.amazon.com/eks/.
Amazon Elastic Transcoder	A cloud-based media transcoding service. Elastic Transcoder is a highly scalable tool for converting (or *transcoding*) media files from their source format into versions that play on devices such as smartphones, tablets, and PCs. See Also https://aws.amazon.com/elastictranscoder/.
Amazon ElastiCache	A web service that simplifies deploying, operating, and scaling an in-memory cache in the cloud. The service improves the performance of web applications by providing information retrieval from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases. See Also https://aws.amazon.com/elasticache/.
Amazon OpenSearch Service (OpenSearch Service)	An AWS managed service for deploying, operating, and scaling OpenSearch, an open-source search and analytics engine, in the AWS Cloud. Amazon OpenSearch Service (OpenSearch Service) also offers security options, high availability, data durability, and direct access to the OpenSearch API. See Also https://aws.amazon.com/elasticsearch-service.
Amazon EMR	A web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop (p. 153) processing combined with several AWS products to do such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing. See Also https://aws.amazon.com/elasticmapreduce.
Amazon EventBridge	A serverless event bus service that you can use to connect your applications with data from a variety of sources and routes that data to targets such as AWS Lambda. You can set up routing rules to determine where to send your data to build application architectures that react in real time to all of your data sources. See Also https://aws.amazon.com/eventbridge/.

Amazon Forecast	A fully managed service that uses statistical and machine learning algorithms to produce highly accurate time-series forecasts. See Also https://aws.amazon.com/forecast/.
Amazon GameLift	A managed service for deploying, operating, and scaling session-based multiplayer games. See Also https://aws.amazon.com/gamelift/.
Amazon GuardDuty	A continuous security monitoring service. Amazon GuardDuty can help to identify unexpected and potentially unauthorized or malicious activity in your AWS environment. See Also https://aws.amazon.com/guardduty/.
Amazon Inspector	An automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed report with prioritized steps for remediation. See Also https://aws.amazon.com/inspector.
Amazon Kinesis	A platform for streaming data on AWS. Kinesis offers services that simplify the loading and analysis of streaming data. See Also https://aws.amazon.com/kinesis/.
Amazon Kinesis Data Firehose	A fully managed service for loading streaming data into AWS. Kinesis Data Firehose can capture and automatically load streaming data into Amazon S3 (p. 129) and Amazon Redshift (p. 129), enabling near real-time analytics with existing business intelligence tools and dashboards. Kinesis Data Firehose automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it. See Also https://aws.amazon.com/kinesis/firehose/.
Amazon Kinesis Data Streams	A web service for building custom applications that process or analyze streaming data for specialized needs. Amazon Kinesis Data Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources. See Also https://aws.amazon.com/kinesis/streams/.
Amazon Lightsail	Lightsail is designed to be the easiest way to launch and manage a virtual private server with AWS. Lightsail offers bundled plans that include everything you need to deploy a virtual private server, for a low monthly rate. See Also https://aws.amazon.com/lightsail/.
Amazon Lookout for Equipment	A machine learning service that uses data from sensors mounted on factory equipment to detect abnormal behavior so you can take action before machine failures occur. See Also https://aws.amazon.com/lookout-for-equipment/.
Amazon Lookout for Vision	A machine learning service that uses computer vision (CV) to find defects in industrial products. Amazon Lookout for Vision can identify missing components in an industrial product, damage to vehicles or structures, irregularities in production lines, and even minuscule defects in silicon wafers—or any other physical item where quality is important. See Also https://aws.amazon.com/lookout-for-vision/.
Amazon Lumberyard	A cross-platform, 3D game engine for creating high-quality games. You can connect games to the compute and storage of the AWS Cloud and engage fans on Twitch. See Also https://aws.amazon.com/lumberyard/.

Amazon Machine Image (AMI)	An encrypted machine image stored in Amazon Elastic Block Store (Amazon EBS) (p. 126) or Amazon Simple Storage Service (p. 129). AMIs function similar to a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, middleware, and web servers.
Amazon Machine Learning	A cloud-based service that creates machine learning (ML) models by finding patterns in your data, and uses these models to process new data and generate predictions. See Also http://aws.amazon.com/machine-learning/.
Amazon Macie	A security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. See Also http://aws.amazon.com/macie/.
Amazon Managed Blockchain	A fully managed service for creating and managing scalable blockchain networks using popular open source frameworks. See Also http://aws.amazon.com/managed-blockchain/.
Amazon ML	See Amazon Machine Learning.
Amazon Mobile Analytics (Mobile Analytics)	A service for collecting, visualizing, understanding, and extracting mobile app usage data at scale. See Also https://aws.amazon.com/mobileanalytics.
Amazon Monitron	An end-to-end system that uses machine learning (ML) to detect abnormal behavior in industrial machinery. Use Amazon Monitron to implement predictive maintenance and reduce unplanned downtime. See Also https://aws.amazon.com/monitron/.
Amazon MQ	A managed message broker service for Apache ActiveMQ that makes it easy to set up and operate message brokers in the cloud. See Also https://aws.amazon.com/amazon-mq/.
Amazon Neptune	A managed graph database service that you can use to build and run applications that work with highly connected datasets. Neptune supports the popular graph query languages Apache TinkerPop Gremlin and W3C's SPARQL, enabling you to build queries that efficiently navigate highly connected datasets. See Also https://aws.amazon.com/neptune/.
Amazon Personalize	An artificial intelligence service for creating individualized product and content recommendations. See Also https://aws.amazon.com/personalize/.
Amazon Polly	A text-to-speech (TTS) service that turns text into natural-sounding human speech. Amazon Polly provides dozens of lifelike voices across a broad set of languages so that you can build build speech-enabled applications that work in many different countries. See Also https://aws.amazon.com/polly/.
Amazon QuickSight	A fast, cloud-powered business analytics service that makes it easy to build visualizations, perform analysis, and quickly get business insights from your data. See Also https://aws.amazon.com/quicksight/.
Amazon Rekognition	A machine learning service that identifies objects, people, text, scenes, and activities, including inappropriate content, in either image or video files. With Amazon Rekognition Custom Labels, you can create a customized ML model that detects objects and scenes specific to your business in images. See Also https://aws.amazon.com/rekognition/.

Amazon Redshift	A fully managed, petabyte-scale data warehouse service in the cloud. With Amazon Redshift, you can analyze your data using your existing business intelligence tools. See Also https://aws.amazon.com/redshift/.
Amazon Relational Database Service (Amazon RDS)	A web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. See Also https://aws.amazon.com/rds.
Amazon Resource Name (ARN)	A standardized way to refer to an AWS resource (p. 168) (for example, arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob).
Amazon Route 53	A web service you can use to create a new DNS service or to migrate your existing DNS service to the cloud. See Also https://aws.amazon.com/route53.
Amazon S3	Storage for the internet. You can use it to store and retrieve any amount of data at any time, from anywhere on the web. See Also Amazon Simple Storage Service (Amazon S3), https://aws.amazon.com/s3.
Amazon S3-Backed AMI	See instance store-backed AMI.
Amazon S3 Glacier	A secure, durable, and low-cost storage service for data archiving and long-term backup. You can reliably store large or small amounts of data for significantly less than on-premises solutions. S3 Glacier is optimized for infrequently accessed data, where a retrieval time of several hours is suitable. See Also https://aws.amazon.com/glacier/.
AWS Security Hub	A service that provides a comprehensive view of the security state of your AWS resources. Security Hub collects security data from AWS accounts and services and helps you analyze your security trends to identify and prioritize the security issues across your AWS environment. See Also https://aws.amazon.com/security-hub/.
Amazon Silk	A next-generation web browser available only on Fire OS tablets and phones. Built on a split architecture that divides processing between the client and the AWS Cloud, Amazon Silk is designed to create a faster, more responsive mobile browsing experience.
Amazon Simple Email Service (Amazon SES)	An easy-to-use, cost-effective email solution for applications. See Also https://aws.amazon.com/ses.
Amazon Simple Notification Service (Amazon SNS)	A web service that applications, users, and devices can use to instantly send and receive notifications from the cloud. See Also https://aws.amazon.com/sns.
Amazon Simple Queue Service (Amazon SQS)	Reliable and scalable hosted queues for storing messages as they travel between computers. See Also https://aws.amazon.com/sqs.
Amazon Simple Storage Service (Amazon S3)	Storage for the internet. You can use it to store and retrieve any amount of data at any time, from anywhere on the web. See Also https://aws.amazon.com/s3.
Amazon Simple Workflow Service (Amazon SWF)	A fully managed service that helps developers build, run, and scale background jobs that have parallel or sequential steps. Amazon SWF functions similar to a state tracker and task coordinator in the AWS Cloud.

	See Also https://aws.amazon.com/swf/.
Amazon Sumerian	A set of tools for creating and running high-quality 3D, augmented reality (AR), and virtual reality (VR) applications on the web. See Also https://aws.amazon.com/sumerian/.
Amazon Textract	A service that automatically extracts text and data from scanned documents. Amazon Textract goes beyond simple optical character recognition (OCR) to also identify the contents of fields in forms and information stored in tables. See Also https://aws.amazon.com/textract/.
Amazon Transcribe	A machine learning service that uses automatic speech recognition (ASR) to quickly and accurately convert speech to text. See Also https://aws.amazon.com/transcribe/.
Amazon Transcribe Medical	An automatic speech recognition (ASR) service for adding medical speech-to-text capabilities to voice-enabled clinical documentation applications. See Also https://aws.amazon.com/transcribe/medical/.
Amazon Translate	A neural machine translation service that delivers fast, high-quality, and affordable language translation. See Also https://aws.amazon.com/translate/.
Amazon Virtual Private Cloud (Amazon VPC)	A web service for provisioning a logically isolated section of the AWS Cloud virtual network that you define. You control your virtual networking environment, including selection of your own IP address range, creation of subnet (p. 175)s, and configuration of route table (p. 169)s and network gateways. See Also https://aws.amazon.com/vpc.
Amazon VPC	See Amazon Virtual Private Cloud (Amazon VPC).
Amazon Web Services (AWS)	An infrastructure web services platform in the cloud for companies of all sizes. See Also https://aws.amazon.com/what-is-cloud-computing/.
Amazon WorkDocs	A managed, secure enterprise document storage and sharing service with administrative controls and feedback capabilities. See Also https://aws.amazon.com/workdocs/.
Amazon WorkLink	A cloud-based service that provides secure access to internal websites and web apps from mobile devices. See Also https://aws.amazon.com/worklink/.
Amazon WorkMail	A managed, secure business email and calendar service with support for existing desktop and mobile email clients. See Also https://aws.amazon.com/workmail/.
Amazon WorkSpaces	A managed, secure desktop computing service for provisioning cloud-based desktops and providing users access to documents, applications, and resource (p. 168)s from supported devices. See Also https://aws.amazon.com/workspaces/.
Amazon WorkSpaces Application Manager (Amazon WAM)	A web service for deploying and managing applications for WorkSpaces. Amazon WAM accelerates software deployment, upgrades, patching, and retirement by packaging Windows desktop applications into virtualized application containers. See Also https://aws.amazon.com/workspaces/applicationmanager.
AMI	See Amazon Machine Image (AMI).
analysis scheme	Amazon CloudSearch (p. 124): Language-specific text analysis options that are applied to a text field to control stemming and configure stopwords and synonyms.

application	AWS Elastic Beanstalk (p. 134): A logical collection of components, including environments, versions, and environment configurations. An application is conceptually similar to a folder. AWS CodeDeploy (p. 133): A name that uniquely identifies the application to be deployed. AWS CodeDeploy uses this name to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.
Application Auto Scaling	A web service that you can use to configure automatic scaling for AWS resources beyond Amazon EC2, such as Amazon ECS services, Amazon EMR clusters, and DynamoDB tables. See Also https://aws.amazon.com/autoscaling/.
Application Billing	The location where your customers manage the Amazon DevPay products they've purchased. The web address is http://www.amazon.com/dp-applications.
application revision	AWS CodeDeploy (p. 133): An archive file containing source content—such as source code, webpages, executable files, and deployment scripts—along with an application specification file (p. 131). Revisions are stored in Amazon S3 (p. 129) bucket (p. 140)s or GitHub (p. 152) repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.
application specification file	AWS CodeDeploy (p. 133): A YAML-formatted file used to map the source files in an application revision to destinations on the instance. The file is also used to specify custom permissions for deployed files and specify scripts to be run on each instance at various stages of the deployment process.
application version	AWS Elastic Beanstalk (p. 134): A specific, labeled iteration of an application that represents a functionally consistent set of deployable application code. A version points to an Amazon S3 (p. 129) object (a JAVA WAR file) that contains the application code.
AppSpec file	See application specification file.
ARN	See Amazon Resource Name (ARN).
artifact	AWS CodePipeline (p. 133): A copy of the files or changes that will be worked upon by the pipeline.
asymmetric encryption	Encryption (p. 149) that uses both a public key and a private key.
asynchronous bounce	A type of bounce (p. 140) that occurs when a receiver (p. 167) initially accepts an email message for delivery and then subsequently fails to deliver it.
atomic counter	DynamoDB: A method of incrementing or decrementing the value of an existing attribute without interfering with other write requests.
attribute	A fundamental data element, something that doesn't need to be broken down any further. In DynamoDB, attributes are similar in many ways to fields or columns in other database systems. Amazon Machine Learning: A unique, named property within an observation in a dataset. In tabular data, such as spreadsheets or comma-separated values (.csv) files, the column headings represent the attributes, and the rows contain values for each attribute.
AUC	Area Under a Curve. An industry-standard metric to evaluate the quality of a binary classification machine learning model. AUC measures the ability of the

model to predict a higher score for positive examples, those that are "correct," than for negative examples, those that are "incorrect." The AUC metric returns a decimal value from 0 to 1. AUC values near 1 indicate an ML model that's highly accurate.

Aurora	See Amazon Aurora.
authenticated encryption	Encryption (p. 149) that provides confidentiality, data integrity, and authenticity assurances of the encrypted data.
authentication	The process of proving your identity to a system.
Auto Scaling group	A representation of multiple EC2 instance (p. 148)s that share similar characteristics, and that are treated as a logical grouping for the purposes of instance scaling and management.
Availability Zone	A distinct location within a Region (p. 167) that's insulated from failures in other Availability Zones, and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.
AWS	See Amazon Web Services (AWS).
AWS Application Discovery Service	A web service that helps you plan to migrate to AWS by identifying IT assets in a data center—including servers, virtual machines, applications, application dependencies, and network infrastructure. See Also https://aws.amazon.com/about-aws/whats-new/2016/04/aws-application-discovery-service/.
AWS AppSync	An enterprise level, fully managed GraphQL service with real-time data synchronization and offline programming features. See Also https://aws.amazon.com/appsync/.
AWS Auto Scaling	A fully managed service that you can use to quickly discover the scalable AWS resources that are part of your application and configure dynamic scaling. See Also https://aws.amazon.com/autoscaling/.
AWS Backup	A managed backup service that you can use to centralize and automate the backup of data across AWS services in the cloud and on premises. See Also https://aws.amazon.com/backup/.
AWS Billing and Cost Management	The AWS Cloud computing model where you pay for services on demand and use as much or as little as you need. While resource (p. 168)s are active under your account, you pay for the cost of allocating those resources. You also pay for any incidental usage associated with those resources, such as data transfer or allocated storage. See Also https://aws.amazon.com/billing/new-user-faqs/.
AWS Blockchain Templates	A service for creating and deploying open-source blockchain frameworks on AWS, such as Ethereum and Hyperledger Fabric. See Also https://aws.amazon.com/blockchain/templates/.
AWS Certificate Manager (ACM)	A web service for provisioning, managing, and deploying Secure Sockets Layer/Transport Layer Security (p. 177) (SSL/TLS) certificates for use with AWS services. See Also https://aws.amazon.com/certificate-manager/.
AWS Certificate Manager Private Certificate Authority (ACM PCA)	A hosted private certificate authority service for issuing and revoking private digital certificate (p. 141)s.

	See Also https://aws.amazon.com/certificate-manager/private-certificate-authority/.
AWS Cloud Development Kit (CDK)	An open-source software development framework for defining your cloud infrastructure in code and provisioning it through AWS CloudFormation. See Also https://aws.amazon.com/cdk/.
AWS Cloud Map	A service that you use to create and maintain a map of the backend services and resources that your applications depend on. With AWS Cloud Map, you can name and discover your AWS Cloud resources. See Also https://aws.amazon.com/cloud-map.
AWS Cloud9	A cloud-based integrated development environment (IDE) that you use to write, run, and debug code. See Also https://aws.amazon.com/cloud9/.
AWS CloudFormation	A service for writing or changing templates that create and delete related AWS resource (p. 168)s together as a unit. See Also https://aws.amazon.com/cloudformation.
AWS CloudHSM	A web service that helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated hardware security module (HSM) appliances within the AWS Cloud. See Also https://aws.amazon.com/cloudhsm/.
AWS CloudTrail	A web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. See Also https://aws.amazon.com/cloudtrail/.
AWS CodeBuild	A fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. See Also https://aws.amazon.com/codebuild.
AWS CodeCommit	A fully managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. See Also https://aws.amazon.com/codecommit.
AWS CodeDeploy	A service that automates code deployments to any instance, including EC2 instance (p. 148)s and instance (p. 154)s running on-premises. See Also https://aws.amazon.com/codedeploy.
AWS CodeDeploy agent	A software package that, when installed and configured on an instance, enables that instance to be used in CodeDeploy deployments.
AWS CodePipeline	A continuous delivery service for fast and reliable application updates. See Also https://aws.amazon.com/codepipeline.
AWS Command Line Interface (AWS CLI)	A unified downloadable and configurable tool for managing AWS services. Control multiple AWS services from the command line and automate them through scripts. See Also https://aws.amazon.com/cli/.
AWS Config	A fully managed service that provides an AWS resource (p. 168) inventory, configuration history, and configuration change notifications for better security and governance. You can create rules that automatically check the configuration of AWS resources that AWS Config records. See Also https://aws.amazon.com/config/.

AWS Database Migration Service	A web service that can help you migrate data to and from many widely used commercial and open-source databases. See Also https://aws.amazon.com/dms.
AWS Data Pipeline	A web service for processing and moving data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals. See Also https://aws.amazon.com/datapipeline.
AWS Device Farm (Device Farm)	An app testing service that allows developers to test Android, iOS, and Fire OS devices on real, physical phones and tablets that are hosted by AWS. See Also https://aws.amazon.com/device-farm.
AWS Direct Connect	A web service that simplifies establishing a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment. See Also https://aws.amazon.com/directconnect.
AWS Directory Service	A managed service for connecting your AWS resource (p. 168)s to an existing on-premises Microsoft Active Directory or to set up and operate a new, standalone directory in the AWS Cloud. See Also https://aws.amazon.com/directoryservice.
AWS Elastic Beanstalk	A web service for deploying and managing applications in the AWS Cloud without worrying about the infrastructure that runs those applications. See Also https://aws.amazon.com/elasticbeanstalk.
AWS Elemental MediaConnect	A service that broadcasters and other premium video providers can reliably use to ingest live video into the AWS Cloud and distribute it to multiple destinations inside or outside the AWS Cloud. See Also https://aws.amazon.com/mediaconnect.
AWS Elemental MediaConvert	A file-based video conversion service that transforms media into formats required for traditional broadcast and for internet streaming to multi-screen devices. See Also https://aws.amazon.com/mediaconvert.
AWS Elemental MediaLive	A video service that you can use to create live outputs for broadcast and streaming delivery. See Also https://aws.amazon.com/medialive.
AWS Elemental MediaPackage	A just-in-time packaging and origination service that you can use to format highly secure and reliable live outputs for a variety of devices. See Also https://aws.amazon.com/mediapackage.
AWS Elemental MediaStore	A storage service optimized for media that provides the performance, consistency, and low latency required to deliver live and on-demand video content at scale. See Also https://aws.amazon.com/mediastore.
AWS Elemental MediaTailor	A video service that you can use to serve targeted ads to viewers while maintaining broadcast quality in over-the-top (OTT) video applications. See Also https://aws.amazon.com/mediatailor.
AWS Encryption SDK	A client-side encryption library designed to make it easy for everyone to encrypt and decrypt data using industry standards and best practices. See Also https://aws.amazon.com/blogs/security/tag/aws-encryption-sdk/.
AWS Firewall Manager	A service that you use with AWS WAF to simplify your AWS WAF administration and maintenance tasks across multiple accounts and resources. With AWS Firewall Manager, you set up your firewall rules only once. The service automatically

	applies your rules across your accounts and resources, even as you add new resources. See Also https://aws.amazon.com/firewall-manager.
AWS Global Accelerator	A network layer service that you use to create accelerators that direct traffic to optimal endpoints over the AWS global network. This improves the availability and performance of your internet applications that are used by a global audience. See Also https://aws.amazon.com/global-accelerator.
AWS Glue	A fully managed extract, transform, and load (ETL) (p. 151) service that you can use to catalog data and load it for analytics. With AWS Glue, you can discover your data, develop scripts to transform sources into targets, and schedule and run ETL jobs in a serverless environment. See Also https://aws.amazon.com/glue.
AWS GovCloud (US)	An isolated AWS Region designed to host sensitive workloads in the cloud, ensuring that this work meets the US government's regulatory and compliance requirements. The AWS GovCloud (US) Region adheres to United States International Traffic in Arms Regulations (ITAR), Federal Risk and Authorization Management Program (FedRAMP) requirements, Department of Defense (DOD) Cloud Security Requirements Guide (SRG) Levels 2 and 4, and Criminal Justice Information Services (CJIS) Security Policy requirements. See Also https://aws.amazon.com/govcloud-us/.
AWS Identity and Access Management (IAM)	A web service that Amazon Web Services (AWS) (p. 130) customers can use to manage users and user permissions within AWS. See Also https://aws.amazon.com/iam.
AWS Import/Export	A service for transferring large amounts of data between AWS and portable storage devices. See Also https://aws.amazon.com/importexport.
AWS IoT Core	A managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices. See Also https://aws.amazon.com/iot.
AWS IoT 1-Click	A service that simple devices can use to launch AWS Lambda functions. See Also https://aws.amazon.com/iot-1-click.
AWS IoT Analytics	A fully managed service used to run sophisticated analytics on massive volumes of IoT data. See Also https://aws.amazon.com/iot-analytics.
AWS IoT Device Defender	An AWS IoT security service that you can use to audit the configuration of your devices, monitor your connected devices to detect abnormal behavior, and to mitigate security risks. See Also https://aws.amazon.com/iot-device-defender.
AWS IoT Device Management	A service used to securely onboard, organize, monitor, and remotely manage IoT devices at scale. See Also https://aws.amazon.com/iot-device-management.
AWS IoT Events	A fully managed AWS IoT service that makes it easy to detect and respond to events from IoT sensors and applications. See Also https://aws.amazon.com/iot-events.
AWS IoT Greengrass	Software that you can use to run local compute, messaging, data caching, sync, and ML inference capabilities for connected devices in a secure way. See Also https://aws.amazon.com/greengrass.

AWS IoT SiteWise	A managed service that you can use to collect, organize, and analyze data from industrial equipment at scale. See Also https://aws.amazon.com/iot-sitewise.
AWS IoT Things Graph	A service that makes it easy to visually connect different devices and web services to build IoT applications. See Also https://aws.amazon.com/iot-things-graph.
AWS Key Management Service (AWS KMS)	A managed service that simplifies the creation and control of encryption (p. 149) keys that are used to encrypt data. See Also https://aws.amazon.com/kms.
AWS Lambda	A web service that you can use to run code without provisioning or managing servers. You can run code for virtually any type of application or backend service with zero administration. You can set up your code to automatically start from other AWS services or call it directly from any web or mobile app. See Also https://aws.amazon.com/lambda/.
AWS managed key	One type of customer master key (CMK) (p. 144) in AWS Key Management Service (AWS KMS) (p. 136).
AWS managed policy	An IAM (p. 135) managed policy (p. 158) that's created and managed by AWS.
AWS Management Console	A graphical interface to manage compute, storage, and other cloud resource (p. 168)s. See Also https://aws.amazon.com/console.
AWS Management Portal for vCenter	A web service for managing your AWS resource (p. 168)s using VMware vCenter. You install the portal as a vCenter plugin within your existing vCenter environment. Once installed, you can migrate VMware VMs to Amazon EC2 (p. 125) and manage AWS resources from within vCenter. See Also https://aws.amazon.com/ec2/vcenter-portal/.
AWS Marketplace	A web portal where qualified partners market and sell their software to AWS customers. AWS Marketplace is an online software store that helps customers find, buy, and immediately start using the software and services that run on AWS. See Also https://aws.amazon.com/partners/aws-marketplace/.
AWS Mobile Hub (Mobile Hub)	An integrated console for building, testing, and monitoring mobile apps. See Also https://aws.amazon.com/mobile.
AWS Mobile SDK	A software development kit whose libraries, code examples, and documentation help you build high quality mobile apps for the iOS, Android, Fire OS, Unity, and Xamarin platforms. See Also https://aws.amazon.com/mobile/sdk.
AWS OpsWorks	A configuration management service that helps you use Chef to configure and operate groups of instances and applications. You can define the application's architecture and the specification of each component including package installation, software configuration, and resource (p. 168)s such as storage. You can automate tasks based on time, load, lifecycle events, and more. See Also https://aws.amazon.com/opsworks/.
AWS Organizations	An account management service that you can use to consolidate multiple AWS accounts into an organization that you create and centrally manage. See Also https://aws.amazon.com/organizations/.
AWS Resource Access Manager	A service that you can use to share your resources with any AWS account or organization in AWS Organizations.

See Also https://aws.amazon.com/ram/.

AWS ParallelCluster	An AWS supported open source cluster management tool that helps you to deploy and manage high performance computing (HPC) clusters in the AWS Cloud.
AWS SDK for C++	A software development kit for that provides C++ APIs for many AWS services including Amazon S3 (p. 129), Amazon EC2 (p. 125), Amazon DynamoDB (p. 125), and more. The single, downloadable package includes the AWS C++ library, code examples, and documentation. See Also https://aws.amazon.com/sdk-for-cpp/.
AWS SDK for Go	A software development kit for integrating your Go application with the full suite of AWS services. See Also https://aws.amazon.com/sdk-for-go/.
AWS SDK for Java	A software development kit that provides Java API operations for many AWS services including Amazon S3 (p. 129), Amazon EC2 (p. 125), Amazon DynamoDB (p. 125), and more. The single, downloadable package includes the AWS Java library, code examples, and documentation. See Also https://aws.amazon.com/sdk-for-java/.
AWS SDK for JavaScript in the Browser	A software development kit for accessing AWS services from JavaScript code running in the browser. Authenticate users through Facebook, Google, or Login with Amazon using web identity federation. Store application data in Amazon DynamoDB (p. 125), and save user files to Amazon S3 (p. 129). See Also https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/.
AWS SDK for JavaScript in Node.js	A software development kit for accessing AWS services from JavaScript in Node.js. The SDK provides JavaScript objects for AWS services, including Amazon S3 (p. 129), Amazon EC2 (p. 125), Amazon DynamoDB (p. 125), and Amazon Simple Workflow Service (Amazon SWF) (p. 129) . The single, downloadable package includes the AWS JavaScript library and documentation. See Also https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/.
AWS SDK for .NET	A software development kit that provides .NET API operations for AWS services including Amazon S3 (p. 129), Amazon EC2 (p. 125), IAM (p. 135), and more. You can download the SDK as multiple service-specific packages on NuGet. See Also https://aws.amazon.com/sdk-for-net/.
AWS SDK for PHP	A software development kit and open-source PHP library for integrating your PHP application with AWS services such as Amazon S3 (p. 129), Amazon S3 Glacier (p. 129), and Amazon DynamoDB (p. 125). See Also https://aws.amazon.com/sdk-for-php/.
AWS SDK for Python (Boto)	A software development kit for using Python to access AWS services such as Amazon EC2 (p. 125), Amazon EMR (p. 126), Amazon EC2 Auto Scaling (p. 125), Amazon Kinesis (p. 127), or AWS Lambda (p. 136). See Also http://boto.readthedocs.org/en/latest/.
AWS SDK for Ruby	A software development kit for accessing AWS services from Ruby. The SDK provides Ruby classes for many AWS services including Amazon S3 (p. 129), Amazon EC2 (p. 125), Amazon DynamoDB (p. 125). and more. The single, downloadable package includes the AWS Ruby Library and documentation. See Also https://aws.amazon.com/sdk-for-ruby/.
AWS Secrets Manager	A service for securely encrypting, storing, and rotating credentials for databases and other services. See Also https://aws.amazon.com/secrets-manager/.

AWS Security Token Service (AWS STS)	A web service for requesting temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) (p. 135) users or for users that you authenticate (federated users (p. 151)). See Also https://aws.amazon.com/iam/.
AWS Service Catalog	A web service that helps organizations create and manage catalogs of IT services that are approved for use on AWS. These IT services can include everything from virtual machine images, servers, software, and databases to complete multitier application architectures. See Also https://aws.amazon.com/servicecatalog/.
AWS Shield	A service that helps to protect your resources—such as Amazon EC2 instances, Elastic Load Balancing load balancers, Amazon CloudFront distributions, and Route 53 hosted zones—against DDoS attacks. AWS Shield is automatically included at no extra cost beyond what you already pay for AWS WAF and your other AWS services. For added protection against DDoS attacks, AWS offers AWS Shield Advanced. See Also https://aws.amazon.com/shield.
AWS Single Sign-On	A cloud-based service that simplifies managing SSO access to AWS accounts and business applications. You can control SSO access and user permissions across all your AWS accounts in AWS Organizations. See Also https://aws.amazon.com/single-sign-on/.
AWS Step Functions	A web service that coordinates the components of distributed applications as a series of steps in a visual workflow. See Also https://aws.amazon.com/step-functions/.
AWS Snowball	A petabyte-scale data transport solution that uses devices designed to be secure to transfer large amounts of data into and out of the AWS Cloud. See Also https://aws.amazon.com/snowball.
Storage Gateway	A web service that connects an on-premises software appliance with cloud-based storage. Storage Gateway provides seamless and secure integration between an organization's on-premises IT environment and AWS storage infrastructure. See Also https://aws.amazon.com/storagegateway/.
AWS Toolkit for Eclipse	An open-source plugin for the Eclipse Java integrated development environment (IDE) that makes it easier to develop, debug, and deploy Java applications using Amazon Web Services. See Also https://aws.amazon.com/eclipse/.
AWS Toolkit for JetBrains	An open-source plugin for the integrated development environments (IDEs) from JetBrains that makes it easier to develop, debug, and deploy serverless applications using Amazon Web Services. See Also https://aws.amazon.com/intellij/, https://aws.amazon.com/pycharm/.
AWS Toolkit for Visual Studio	An extension for Visual Studio that helps in developing, debugging, and deploying .NET applications using Amazon Web Services. See Also https://aws.amazon.com/visualstudio/.
AWS Toolkit for Visual Studio Code	An open-source plugin for the Visual Studio Code (VS Code) editor that makes it easier to develop, debug, and deploy applications using Amazon Web Services. See Also https://aws.amazon.com/visualstudiocode/.
AWS Tools for PowerShell	A set of PowerShell cmdlets to help developers and administrators manage their AWS services from the PowerShell scripting environment. See Also https://aws.amazon.com/powershell/.

AWS Toolkit for Microsoft Azure DevOps	Provides tasks you can use in build and release definitions in VSTS to interact with AWS services. See Also https://aws.amazon.com/vsts/.
AWS Trusted Advisor	A web service that inspects your AWS environment and makes recommendations for saving money, improving system availability and performance, and helping to close security gaps. See Also https://aws.amazon.com/premiumsupport/trustedadvisor/.
AWS VPN CloudHub	Enables secure communication between branch offices using a simple hub-and-spoke model, with or without a VPC (p. 179).
AWS WAF	A web application firewall service that controls access to content by allowing or blocking web requests based on criteria that you specify. For example, you can filter access based on the header values or the IP addresses that the requests originate from. AWS WAF helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. See Also https://aws.amazon.com/waf/.
AWS X-Ray	A web service that collects data about requests that your application serves. X-Ray provides tools that you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization. See Also https://aws.amazon.com/xray/.

B

basic monitoring	Monitoring of AWS provided metrics derived at a 5-minute frequency.
batch	See document batch.
BGP ASN	Border Gateway Protocol Autonomous System Number. A unique identifier for a network, for use in BGP routing. Amazon EC2 (p. 125) supports all 2-byte ASN numbers in the range of 1 – 65335, with the exception of 7224, which is reserved.
batch prediction	Amazon Machine Learning: An operation that processes multiple input data observations at one time (asynchronously). Unlike real-time predictions, batch predictions aren't available until all predictions have been processed. See Also real-time predictions.
billing	See AWS Billing and Cost Management.
binary attribute	Amazon Machine Learning: An attribute for which one of two possible values is possible. Valid positive values are 1, y, yes, t, and true answers. Valid negative values are 0, n, no, f, and false. Amazon Machine Learning outputs 1 for positive values and 0 for negative values. See Also attribute.
binary classification model	Amazon Machine Learning: A machine learning model that predicts the answer to questions where the answer can be expressed as a binary variable. For example, questions with answers of "1" or "0", "yes" or "no", "will click" or "will not click" are questions that have binary answers. The result for a binary classification model is always either a "1" (for a "true" or affirmative answers) or a "0" (for a "false" or negative answers).

block	A dataset. Amazon EMR (p. 126) breaks large amounts of data into subsets. Each subset is called a data block. Amazon EMR assigns an ID to each block and uses a hash table to keep track of block processing.
block device	A storage device that supports reading and (optionally) writing data in fixed-size blocks, sectors, or clusters.
block device mapping	A mapping structure for every AMI (p. 128) and instance (p. 154) that specifies the block devices attached to the instance.
blue/green deployment	CodeDeploy: A deployment method where the instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment).
bootstrap action	A user-specified default or custom action that runs a script or an application on all nodes of a job flow before Hadoop (p. 153) starts.
Border Gateway Protocol Autonomous System Number	See BGP ASN.
bounce	A failed email delivery attempt.
breach	Amazon EC2 Auto Scaling (p. 125): The condition where a user-set threshold (upper or lower boundary) is passed. If the duration of the breach is significant, as set by a breach duration parameter, it can possibly start a scaling activity (p. 170).
bucket	Amazon Simple Storage Service (Amazon S3) (p. 129): A container for stored objects. Every object is contained in a bucket. For example, if the object named `photos/puppy.jpg` is stored in the *DOC-EXAMPLE-BUCKET* bucket, then authorized users can access the object with the URL `https://`*s3-bucket-endpoint*`/`*DOC-EXAMPLE-BUCKET*`/photos/puppy.jpg`.
bucket owner	The person or organization that owns a bucket (p. 140) in Amazon S3 (p. 129). In the same way that Amazon is the only owner of the domain name Amazon.com, only one person or organization can own a bucket.
bundling	A commonly used term for creating an Amazon Machine Image (AMI) (p. 128). It specifically refers to creating instance store-backed AMI (p. 155)s.

C

cache cluster	A logical cache distributed over multiple cache node (p. 140)s. A cache cluster can be set up with a specific number of cache nodes.
cache cluster identifier	Customer-supplied identifier for the cache cluster that must be unique for that customer in an AWS Region (p. 167).
cache engine version	The version of the Memcached service that's running on the cache node.
cache node	A fixed-size chunk of secure, network-attached RAM. Each cache node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of cache nodes are supported, each with varying amounts of associated memory.

cache node type	An EC2 instance (p. 148) type used to run the cache node.
cache parameter group	A container for cache engine parameter values that can be applied to one or more cache clusters.
cache security group	A group maintained by ElastiCache that combines inbound authorizations to cache nodes for hosts belonging to Amazon EC2 (p. 125) security group (p. 171)s specified through the console or the API or command line tools.
campaign	Amazon Personalize (p. 128): A deployed solution version (trained model) with provisioned dedicated transaction capacity for creating real-time recommendations for your application users. After you create a campaign, you use the getRecommendations or getPersonalizedRanking personalization operations to get recommendations. See Also recommendations, solution version.
canned access policy	A standard access control policy that you can apply to a bucket (p. 140) or object. Options include: private, public-read, public-read-write, and authenticated-read.
canonicalization	The process of converting data into a standard format that a service such as Amazon S3 (p. 129) can recognize.
capacity	The amount of available compute size at a given time. Each Auto Scaling group (p. 132) is defined with a minimum and maximum compute size. A scaling activity (p. 170) increases or decreases the capacity within the defined minimum and maximum values.
Cartesian product processor	A processor that calculates a Cartesian product. Also known as a *Cartesian data processor*.
Cartesian product	A mathematical operation that returns a product from multiple sets.
CDN	See content delivery network (CDN).
certificate	A credential that some AWS products use to authenticate AWS accounts (p. 123) and users. Also known as an X.509 certificate (p. 180). The certificate is paired with a private key.
chargeable resources	Features or services whose use incurs fees. Although some AWS products are free, others include charges. For example, in an AWS CloudFormation (p. 133) stack (p. 174), AWS resource (p. 168)s that have been created incur charges. The amount charged depends on the usage load. Use the Amazon Web Services Simple Monthly Calculator to estimate your cost prior to creating instances, stacks, or other resources.
CIDR block	Classless Inter-Domain Routing. An internet protocol address allocation and route aggregation methodology. See Also Classless Inter-Domain Routing in Wikipedia.
ciphertext	Information that has been encrypted (p. 149), as opposed to plaintext (p. 164), which is information that has not.
ClassicLink	A feature for linking an EC2-Classic instance (p. 154) to a VPC (p. 179), allowing your EC2-Classic instance to communicate with VPC instances using private IP addresses. See Also link to VPC, unlink from VPC.
classification	In machine learning, a type of problem that seeks to place (classify) a data sample into a single category or "class." Often, classification problems are modeled to choose one category (class) out of two. These are binary classification problems.

	Problems with more than two available categories (classes) are called "multiclass classification" problems. See Also binary classification model, multiclass classification model.
CLI	See AWS Command Line Interface (AWS CLI).
Cloud Directory	See Amazon Cloud Directory (Cloud Directory).
cloud service provider (CSP)	A company that provides subscribers with access to internet-hosted computing, storage, and software services.
CloudHub	See AWS VPN CloudHub.
cluster	A logical grouping of container instance (p. 143)s that you can place task (p. 176)s on. Amazon OpenSearch Service (OpenSearch Service) (p. 126): A logical grouping of one or more data nodes, optional dedicated master nodes, and storage required to run Amazon OpenSearch Service (OpenSearch Service) and operate your OpenSearch Service domain. See Also data node, dedicated master node, node.
cluster compute instance	A type of instance (p. 154) that provides a great amount of CPU power coupled with increased networking performance, making it well suited for High Performance Compute (HPC) applications and other demanding network-bound applications.
cluster placement group	A logical cluster compute instance (p. 142) grouping to provide lower latency and high-bandwidth connectivity between the instance (p. 154)s.
cluster status	Amazon OpenSearch Service (OpenSearch Service) (p. 126): An indicator of the health of a cluster. A status can be green, yellow, or red. At the shard level, green means that all shards are allocated to nodes in a cluster, yellow means that the primary shard is allocated but the replica shards aren't, and red means that the primary and replica shards of at least one index aren't allocated. The shard status determines the index status, and the index status determines the cluster status.
CMK	See customer master key (CMK).
CNAME	Canonical Name Record. A type of resource record (p. 169) in the Domain Name System (DNS) that specifies that the domain name is an alias of another, canonical domain name. Specifically, it's an entry in a DNS table that you can use to alias one fully qualified domain name to another.
Code Signing for AWS IoT	A service for signing code that you create for any IoT device that's supported by Amazon Web Services (AWS).
complaint	The event where a recipient (p. 167) who doesn't want to receive an email message chooses "Mark as Spam" within the email client, and the internet service provider (ISP) (p. 155) sends a notification to Amazon SES (p. 129).
compound query	Amazon CloudSearch (p. 124): A search request that specifies multiple search criteria using the Amazon CloudSearch structured search syntax.
condition	IAM (p. 135): Any restriction or detail about a permission. The condition is *D* in the statement "A has permission to do B to C where D applies." AWS WAF (p. 139): A set of attributes that AWS WAF searches for in web requests to AWS resource (p. 168)s such as Amazon CloudFront (p. 124) distributions. Conditions can include values such as the IP addresses that web requests originate from or values in request headers. Based on the specified

	conditions, you can configure AWS WAF to allow or block web requests to AWS resources.
conditional parameter	See mapping.
configuration API	Amazon CloudSearch (p. 124): The API call that you use to create, configure, and manage search domains.
configuration template	A series of key–value pairs that define parameters for various AWS products so that AWS Elastic Beanstalk (p. 134) can provision them for an environment.
consistency model	The method a service uses to achieve high availability. For example, it could involve replicating data across multiple servers in a data center. See Also eventual consistency.
console	See AWS Management Console.
consolidated billing	A feature of the AWS Organizations service for consolidating payment for multiple AWS accounts. You create an organization that contains your AWS accounts, and you use the management account of your organization to pay for all member accounts. You can see a combined view of AWS costs that are incurred by all accounts in your organization, and you can get detailed cost reports for individual accounts.
container	A Linux container that was created from a Docker image as part of a task (p. 176).
container definition	Specifies which Docker image (p. 147) to use for a container (p. 143), how much CPU and memory the container is allocated, and more options. The container definition is included as part of a task definition (p. 176).
container instance	An EC2 instance (p. 148) that's running the Amazon Elastic Container Service (Amazon ECS) (p. 126) agent and has been registered into a cluster (p. 142). Amazon ECS task (p. 176)s are placed on active container instances.
container registry	Stores, manages, and deploys Docker image (p. 147)s.
content delivery network (CDN)	A web service that speeds up distribution of your static and dynamic web content —such as .html, .css, .js, media files, and image files—to your users by using a worldwide network of data centers. When a user requests your content, the request is routed to the data center that provides the lowest latency (time delay). If the content is already in the location with the lowest latency, the CDN delivers it immediately. If not, the CDN retrieves it from an origin that you specify (for example, a web server or an Amazon S3 bucket). With some CDNs, you can help secure your content by configuring an HTTPS connection between users and data centers, and between data centers and your origin. Amazon CloudFront is an example of a CDN.
contextual metadata	Amazon Personalize (p. 128): Interactions data that you collect about a user's browsing context (such as device used or location) when an event (such as a click) occurs. Contextual metadata can improve recommendation relevance for new and existing users. See Also Interactions dataset, event.
continuous delivery	A software development practice where code changes are automatically built, tested, and prepared for a release to production. See Also https://aws.amazon.com/devops/continuous-delivery/.
continuous integration	A software development practice where developers regularly merge code changes into a central repository, after which automated builds and tests are run. See Also https://aws.amazon.com/devops/continuous-integration/.

cooldown period	Amount of time that Amazon EC2 Auto Scaling (p. 125) doesn't allow the desired size of the Auto Scaling group (p. 132) to be changed by any other notification from an Amazon CloudWatch (p. 124) alarm (p. 123).
core node	An EC2 instance (p. 148) that runs Hadoop (p. 153) map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS). Core nodes are managed by the master node (p. 159), which assigns Hadoop tasks to nodes and monitors their status. The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run. Because core nodes store data, you can't remove them from a job flow. However, you can add more core nodes to a running job flow. Core nodes run both the DataNodes and TaskTracker Hadoop daemons.
corpus	Amazon CloudSearch (p. 124): A collection of data that you want to search.
coverage	Amazon Personalize (p. 128): An evaluation metric that tells you the proportion of unique items that Amazon Personalize might recommend using your model out of the total number of unique items in Interactions and Items datasets. To make sure Amazon Personalize recommends more of your items, use a model with a higher coverage score. Recipes that feature item exploration, such as user-personalization, have higher coverage than those that don't, such as popularity-count. See Also metrics, Items dataset, Interactions dataset, item exploration, user-personalization recipe, popularity-count recipe.
credential helper	AWS CodeCommit (p. 133): A program that stores credentials for repositories and supplies them to Git when making connections to those repositories. The AWS CLI (p. 133) includes a credential helper that you can use with Git when connecting to CodeCommit repositories.
credentials	Also called *access credentials* or *security credentials*. In authentication and authorization, a system uses credentials to identify who is making a call and whether to allow the requested access. In AWS, these credentials are typically the access key ID (p. 122) and the secret access key (p. 171).
cross-account access	The process of permitting limited, controlled use of resource (p. 168)s in one AWS account (p. 123) by a user in another AWS account. For example, in AWS CodeCommit (p. 133) and AWS CodeDeploy (p. 133) you can configure cross-account access so that a user in AWS account A can access an CodeCommit repository created by account B. Or a pipeline in AWS CodePipeline (p. 133) created by account A can use CodeDeploy resources created by account B. In IAM (p. 135) you use a role (p. 169) to delegate (p. 146) temporary access to a user (p. 178) in one account to resources in another.
cross-Region replication	A solution for replicating data across different AWS Regions (p. 167), in near-real time.
customer gateway	A router or software application on your side of a VPN tunnel that's managed by Amazon VPC (p. 130). The internal interfaces of the customer gateway are attached to one or more devices in your home network. The external interface is attached to the virtual private gateway (VGW) (p. 179) across the VPN tunnel.
customer managed policy	An IAM (p. 135) managed policy (p. 158) that you create and manage in your AWS account (p. 123).
customer master key (CMK)	The fundamental resource (p. 168) that AWS Key Management Service (AWS KMS) (p. 136) manages. CMKs can be either customer managed keys or AWS managed keys. Use CMKs inside AWS KMS to encrypt (p. 149) or decrypt up to 4

kilobytes of data directly or to encrypt generated data keys, which are then used to encrypt or decrypt larger amounts of data outside of the service.

D

dashboard	See service health dashboard.
data consistency	A concept that describes when data is written or updated successfully and all copies of the data are updated in all AWS Regions (p. 167). However, it takes time for the data to propagate to all storage locations. To support varied application requirements, Amazon DynamoDB (p. 125) supports both eventually consistent and strongly consistent reads. See Also eventual consistency, eventually consistent read, strongly consistent read.
data node	Amazon OpenSearch Service (OpenSearch Service) (p. 126): An OpenSearch instance that holds data and responds to data upload requests. See Also dedicated master node, node.
data schema	See schema.
data source	The database, file, or repository that provides information required by an application or database. For example, in AWS OpsWorks (p. 136), valid data sources include an instance (p. 154) for a stack's MySQL layer or a stack's Amazon RDS (p. 129) service layer. In Amazon Redshift (p. 129), valid data sources include text files in an Amazon S3 (p. 129) bucket (p. 140), in an Amazon EMR (p. 126) cluster, or on a remote host that a cluster can access through an SSH connection. See Also datasource.
database engine	The database software and version running on the DB instance (p. 146).
database name	The name of a database hosted in a DB instance (p. 146). A DB instance can host multiple databases, but databases hosted by the same DB instance must each have a unique name within that instance.
dataset	Amazon Personalize (p. 128): A container for the data used by Amazon Personalize. There are three types of Amazon Personalize datasets: Users, Items, and Interactions. See Also Interactions dataset, Users dataset, Items dataset.
dataset group	Amazon Personalize (p. 128): A container for Amazon Personalize components, including datasets, event trackers, solutions, filters, campaigns, and batch inference jobs. A dataset group organizes your resources into independent collections, so resources from one dataset group can't influence resources in any other dataset group. See Also dataset, event tracker, solution, campaign.
datasource	Amazon Machine Learning (p. 128): An object that contains metadata about the input data. Amazon ML reads the input data, computes descriptive statistics on its attributes, and stores the statistics—along with a schema and other information—as part of the datasource object. Amazon ML uses datasources to train and evaluate a machine learning model and generate batch predictions. See Also data source.

DB compute class	The size of the database compute platform used to run the instance.
DB instance	An isolated database environment running in the cloud. A DB instance can contain multiple user-created databases.
DB instance identifier	User-supplied identifier for the DB instance. The identifier must be unique for that user in an AWS Region (p. 167).
DB parameter group	A container for database engine parameter values that apply to one or more DB instance (p. 146)s.
DB security group	A method that controls access to the DB instance (p. 146). By default, network access is turned off to DB instances. After inbound traffic is configured for a security group (p. 171), the same rules apply to all DB instances associated with that group.
DB snapshot	A user-initiated point backup of a DB instance (p. 146).
Dedicated Host	A physical server with EC2 instance (p. 148) capacity fully dedicated to a user.
Dedicated Instance	An instance (p. 154) that's physically isolated at the host hardware level and launched within a VPC (p. 179).
dedicated master node	Amazon OpenSearch Service (OpenSearch Service) (p. 126): An OpenSearch instance that performs cluster management tasks, but doesn't hold data or respond to data upload requests. Amazon OpenSearch Service (OpenSearch Service) uses dedicated master nodes to increase cluster stability. See Also data node, node.
Dedicated Reserved Instance	An option that you purchase to guarantee that sufficient capacity will be available to launch Dedicated Instance (p. 146)s into a VPC (p. 179).
delegation	Within a single AWS account (p. 123): Giving AWS user (p. 178)s access to resource (p. 168)s in your AWS account. Between two AWS accounts: Setting up a trust between the account that owns the resource (the trusting account), and the account that contains the users that need to access the resource (the trusted account). See Also trust policy.
delete marker	An object with a key and version ID, but without content. Amazon S3 (p. 129) inserts delete markers automatically into versioned bucket (p. 140)s when an object is deleted.
deliverability	The likelihood that an email message will arrive at its intended destination.
deliveries	The number of email messages, sent through Amazon SES (p. 129), that were accepted by an internet service provider (ISP) (p. 155) for delivery to recipient (p. 167)s over a period of time.
deny	The result of a policy (p. 164) statement that includes deny as the effect, so that a specific action or actions are expressly forbidden for a user, group, or role. Explicit deny take precedence over explicit allow (p. 124).
deployment configuration	AWS CodeDeploy (p. 133): A set of deployment rules and success and failure conditions used by the service during a deployment.
deployment group	AWS CodeDeploy (p. 133): A set of individually tagged instance (p. 154)s, EC2 instance (p. 148)s in Auto Scaling group (p. 132)s, or both.
detailed monitoring	Monitoring of AWS provided metrics derived at a 1-minute frequency.

Description property	A property added to parameters, resource (p. 168)s, resource properties, mappings, and outputs to help you to document AWS CloudFormation (p. 133) template elements.
dimension	A name–value pair (for example, InstanceType=m1.small, or EngineName=mysql), that contains additional information to identify a metric.
discussion forums	A place where AWS users can post technical questions and feedback to help accelerate their development efforts and to engage with the AWS community. For more information, see the Amazon Web Services Discussion Forums.
distribution	A link between an origin server (such as an Amazon S3 (p. 129) bucket (p. 140)) and a domain name, which CloudFront (p. 124) automatically assigns. Through this link, CloudFront identifies the object you have stored in your origin server (p. 162).
DKIM	DomainKeys Identified Mail. A standard that email senders use to sign their messages. ISPs use those signatures to verify that messages are legitimate. For more information, see https://tools.ietf.org/html/rfc6376.
DNS	See Domain Name System.
Docker image	A layered file system template that's the basis of a Docker container (p. 143). Docker images can comprise specific operating systems or applications.
document	Amazon CloudSearch (p. 124): An item that can be returned as a search result. Each document has a collection of fields that contain the data that can be searched or returned. The value of a field can be either a string or a number. Each document must have a unique ID and at least one field.
document batch	Amazon CloudSearch (p. 124): A collection of add and delete document operations. You use the document service API to submit batches to update the data in your search domain.
document service API	Amazon CloudSearch (p. 124): The API call that you use to submit document batches to update the data in a search domain.
document service endpoint	Amazon CloudSearch (p. 124): The URL that you connect to when sending document updates to an Amazon CloudSearch domain. Each search domain has a unique document service endpoint that remains the same for the life of the domain.
domain	Amazon OpenSearch Service (OpenSearch Service) (p. 126): The hardware, software, and data exposed by Amazon OpenSearch Service (OpenSearch Service) endpoints. An OpenSearch Service domain is a service wrapper around an OpenSearch cluster. An OpenSearch Service domain encapsulates the engine instances that process OpenSearch Service requests, the indexed data that you want to search, snapshots of the domain, access policies, and metadata. See Also cluster, Elasticsearch.
Domain Name System	A service that routes internet traffic to websites by translating friendly domain names (for example, www.example.com) into the numeric IP addresses, such as 192.0.2.1 that computers use to connect to each other.
Donation button	An HTML-coded button to provide an easy and secure way for US-based, IRS-certified 501(c)3 nonprofit organizations to solicit donations.
DynamoDB stream	An ordered flow of information about changes to items in anAmazon DynamoDB (p. 125) table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

See Also Amazon DynamoDB Streams.

E

EBS	See Amazon Elastic Block Store (Amazon EBS).
EC2	See Amazon EC2.
EC2 compute unit (ECU)	An AWS standard for compute CPU and memory. You can use this measure to evaluate the CPU capacity of different EC2 instance (p. 148) types.
EC2 instance	A compute instance (p. 154) in the Amazon EC2 (p. 125) service. Other AWS services use the term *EC2 instance* to distinguish these instances from other types of instances they support.
ECR	See Amazon Elastic Container Registry (Amazon ECR).
ECS	See Amazon Elastic Container Service (Amazon ECS).
edge location	A data center that an AWS service uses to perform service-specific operations. For example, CloudFront (p. 124) uses edge locations to cache copies of your content, so the content is closer to your users and can be delivered faster regardless of their location. Route 53 (p. 129) uses edge locations to speed up the response to public DNS queries.
EFS	See Amazon Elastic File System (Amazon EFS).
Elastic	A company that provides open-source solutions—including OpenSearch, Logstash, Kibana, and Beats—that are designed to take data from any source and search, analyze, and visualize it in real time. Amazon OpenSearch Service (OpenSearch Service) is an AWS managed service for deploying, operating, and scaling OpenSearch in the AWS Cloud. See Also Amazon OpenSearch Service (OpenSearch Service), Elasticsearch.
Elastic Block Store	See Amazon Elastic Block Store (Amazon EBS).
Elastic IP address	A fixed (static) IP address that you have allocated in Amazon EC2 (p. 125) or Amazon VPC (p. 130) and then attached to an instance (p. 154). Elastic IP addresses are associated with your account, not a specific instance. They are *elastic* because you can easily allocate, attach, detach, and free them as your needs change. Unlike traditional static IP addresses, Elastic IP addresses allow you to mask instance or Availability Zone (p. 132) failures by rapidly remapping your public IP addresses to another instance.
Elastic Load Balancing	A web service that improves an application's availability by distributing incoming traffic between two or more EC2 instance (p. 148)s. See Also https://aws.amazon.com/elasticloadbalancing.
elastic network interface	An additional network interface that can be attached to an instance (p. 154). Elastic network interfaces include a primary private IP address, one or more secondary private IP addresses, an Elastic IP Address (optional), a MAC address, membership in specified security group (p. 171)s, a description, and a source/destination check flag. You can create an elastic network interface, attach it to an instance, detach it from an instance, and attach it to another instance.

Elasticsearch	An open-source, real-time distributed search and analytics engine used for full-text search, structured search, and analytics. OpenSearch was developed by the Elastic company.
	Amazon OpenSearch Service (OpenSearch Service) is an AWS managed service for deploying, operating, and scaling OpenSearch in the AWS Cloud. See Also Amazon OpenSearch Service (OpenSearch Service), Elastic.
EMR	See Amazon EMR.
encrypt	To use a mathematical algorithm to make data unintelligible to unauthorized user (p. 178)s. Encryption also gives authorized users a method (such as a key or password) to convert the altered data back to its original state.
encryption context	A set of key–value pairs that contains additional information associated with AWS Key Management Service (AWS KMS) (p. 136)–encrypted information.
endpoint	A URL that identifies a host and port as the entry point for a web service. Every web service request contains an endpoint. Most AWS products provide endpoints for a Region to enable faster connectivity.
	Amazon ElastiCache (p. 126): The DNS name of a cache node (p. 140).
	Amazon RDS (p. 129): The DNS name of a DB instance (p. 146).
	AWS CloudFormation (p. 133): The DNS name or IP address of the server that receives an HTTP request.
endpoint port	Amazon ElastiCache (p. 126): The port number used by a cache node (p. 140).
	Amazon RDS (p. 129): The port number used by a DB instance (p. 146).
envelope encryption	The use of a master key and a data key to algorithmically protect data. The master key is used to encrypt and decrypt the data key and the data key is used to encrypt and decrypt the data itself.
environment	AWS Elastic Beanstalk (p. 134): A specific running instance of an application (p. 131). The application has a CNAME and includes an application version and a customizable configuration (which is inherited from the default container type).
	AWS CodeDeploy (p. 133): Instances in a deployment group in a blue/green deployment. At the start of a blue/green deployment, the deployment group is made up of instances in the original environment. At the end of the deployment, the deployment group is made up of instances in the replacement environment.
environment configuration	A collection of parameters and settings that define how an environment and its associated resources behave.
ephemeral store	See instance store.
epoch	The date from which time is measured. For most Unix environments, the epoch is January 1, 1970.
ETL	See extract, transform, and load (ETL).
evaluation	Amazon Machine Learning: The process of measuring the predictive performance of a machine learning (ML) model.
	Also a machine learning object that stores the details and result of an ML model evaluation.

evaluation datasource	The data that Amazon Machine Learning uses to evaluate the predictive accuracy of a machine learning model.
event	Amazon Personalize (p. 128): A user activity—such as a click, a purchase, or a video viewing—that you record and upload to an Amazon Personalize Interactions dataset. You record events individually in real time or record and upload events in bulk. See Also dataset, Interactions dataset.
event tracker	Amazon Personalize (p. 128): Specifies a destination dataset group for event data that you record in real time. When you record events in real time, you provide the ID of the event tracker so that Amazon Personalize knows where to add the data. See Also dataset group, event.
eventual consistency	The method that AWS services use to achieve high availability, which involves replicating data across multiple servers in Amazon's data centers. When data is written or updated and Success is returned, all copies of the data are updated. However, it takes time for the data to propagate to all storage locations. The data will eventually be consistent, but an immediate read might not show the change. Consistency is usually reached within seconds. See Also data consistency, eventually consistent read, strongly consistent read.
eventually consistent read	A read process that returns data from only one Region and might not show the most recent write information. However, if you repeat your read request after a short time, the response should eventually return the latest data. See Also data consistency, eventual consistency, strongly consistent read.
eviction	The deletion by CloudFront (p. 124) of an object from an edge location (p. 148) before its expiration time. If an object in an edge location isn't frequently requested, CloudFront might evict the object (remove the object before its expiration date) to make room for objects that are more popular.
exbibyte (EiB)	A contraction of exa binary byte, an exbibyte is 2^60 or 1,152,921,504,606,846,976 bytes. An exabyte (EB) is 10^18 or 1,000,000,000,000,000,000 bytes. 1,024 EiB is a zebibyte (ZiB) (p. 180).
expiration	For CloudFront (p. 124) caching, the time when CloudFront stops responding to user requests with an object. If you don't use headers or CloudFront distribution (p. 147) settings to specify how long you want objects to stay in an edge location (p. 148), the objects expire after 24 hours. The next time a user requests an object that has expired, CloudFront forwards the request to the origin (p. 162).
explicit impressions	Amazon Personalize (p. 128): A list of items that you manually add to an Amazon Personalize Interactions dataset to influence future recommendations. Unlike *implicit impressions*, where Amazon Personalize automatically derives the impressions data, you choose what to include in explicit impressions. See Also recommendations, Interactions dataset, impressions data, implicit impressions.
explicit launch permission	An Amazon Machine Image (AMI) (p. 128) launch permission granted to a specific AWS account (p. 123).
exponential backoff	A strategy that incrementally increases the wait between retry attempts in order to reduce the load on the system and increase the likelihood that repeated requests will succeed. For example, client applications might wait up to 400 milliseconds before attempting the first retry, up to 1600 milliseconds before the second, and up to 6400 milliseconds (6.4 seconds) before the third.

expression	Amazon CloudSearch (p. 124): A numeric expression that you can use to control how search hits are sorted. You can construct Amazon CloudSearch expressions using numeric fields, other rank expressions, a document's default relevance score, and standard numeric operators and functions. When you use the `sort` option to specify an expression in a search request, the expression is evaluated for each search hit and the hits are listed according to their expression values.
extract, transform, and load (ETL)	A process that's used to integrate data from multiple sources. Data is collected from sources (extract), converted to an appropriate format (transform), and written to a target data store (load) for purposes of analysis and querying. ETL tools combine these three functions to consolidate and move data from one environment to another. AWS Glue (p. 135) is a fully managed ETL service for discovering and organizing data, transforming it, and making it available for search and analytics.

F

facet	Amazon CloudSearch (p. 124): An index field that represents a category that you want to use to refine and filter search results.
facet enabled	Amazon CloudSearch (p. 124): An index field option that enables facet information to be calculated for the field.
FBL	See feedback loop (FBL).
feature transformation	Amazon Machine Learning: The machine learning process of constructing more predictive input representations or "features" from the raw input variables to optimize a machine learning model's ability to learn and generalize. Also known as *data transformation* or *feature engineering*.
federated identity management (FIM)	Allows individuals to sign in to different networks or services, using the same group or personal credentials to access data across all networks. With identity federation in AWS, external identities (federated users) are granted secure access to resource (p. 168)s in an AWS account (p. 123) without having to create IAM user (p. 178)s. These external identities can come from a corporate identity store (such as LDAP or Windows Active Directory) or from a third party (such as Login with Amazon, Facebook, or Google). AWS federation also supports SAML 2.0.
federated user	See federated identity management (FIM).
federation	See federated identity management (FIM).
feedback loop (FBL)	The mechanism by which a mailbox provider (for example, an internet service provider (ISP) (p. 155)) forwards a recipient (p. 167)'s complaint (p. 142) back to the sender (p. 171).
field weight	The relative importance of a text field in a search index. Field weights control how much matches in particular text fields affect a document's relevance score.
filter	A criterion that you specify to limit the results when you list or describe your Amazon EC2 (p. 125) resource (p. 168)s.

filter query	A way to filter search results without affecting how the results are scored and sorted. Specified with the Amazon CloudSearch (p. 124) fq parameter.
FIM	See federated identity management (FIM).
Firehose	See Amazon Kinesis Data Firehose.
format version	See template format version.
forums	See discussion forums.
function	See intrinsic function.
fuzzy search	A simple search query that uses approximate string matching (fuzzy matching) to correct for typographical errors and misspellings.

G

geospatial search	A search query that uses locations specified as a latitude and longitude to determine matches and sort the results.
gibibyte (GiB)	A contraction of giga binary byte, a gibibyte is 2^30 or 1,073,741,824 bytes. A gigabyte (GB) is 10^9 or 1,000,000,000 bytes. 1,024 GiB is a tebibyte (TiB) (p. 176).
GitHub	A web-based repository that uses Git for version control.
global secondary index	An index with a partition key and a sort key that can be different from those on the table. A global secondary index is considered global because queries on the index can span all of the data in a table, across all partitions. See Also local secondary index.
grant	AWS Key Management Service (AWS KMS) (p. 136): A mechanism for giving AWS principal (p. 165)s long-term permissions to use customer master key (CMK) (p. 144)s.
grant token	A type of identifier that allows the permissions in a grant (p. 152) to take effect immediately.
ground truth	The observations used in the machine learning (ML) model training process that include the correct value for the target attribute. To train an ML model to predict house sales prices, the input observations would typically include prices of previous house sales in the area. The sale prices of these houses constitute the ground truth.
group	A collection of IAM (p. 135) user (p. 178)s. You can use IAM groups to simplify specifying and managing permissions for multiple users.

H

Hadoop	Software that enables distributed processing for big data by using clusters and simple programming models. For more information, see http://hadoop.apache.org.
hard bounce	A persistent email delivery failure such as "mailbox does not exist."
hardware VPN	A hardware-based IPsec VPN connection over the internet.
health check	A system call to check on the health status of each instance in an Amazon EC2 Auto Scaling (p. 125) group.
high-quality email	Email that recipients find valuable and want to receive. Value means different things to different recipients and can come in such forms as offers, order confirmations, receipts, or newsletters.
highlights	Amazon CloudSearch (p. 124): Excerpts returned with search results that show where the search terms appear within the text of the matching documents.
highlight enabled	Amazon CloudSearch (p. 124): An index field option that enables matches within the field to be highlighted.
hit	A document that matches the criteria specified in a search request. Also referred to as a *search result*.
HMAC	Hash-based Message Authentication Code. A specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key. You can use it to verify both the data integrity and the authenticity of a message at the same time. AWS calculates the HMAC using a standard, cryptographic hash algorithm, such as SHA-256.
hosted zone	A collection of resource record (p. 169) sets that Amazon Route 53 (p. 129) hosts. Similar to a traditional DNS zone file, a hosted zone represents a collection of records that are managed together under a single domain name.
HRNN	Amazon Personalize (p. 128): A hierarchical recurrent neural network machine learning algorithm that models changes in user behavior and predicts the items that a user might interact with in personal recommendation applications.
HTTP-Query	See Query.
HVM virtualization	Hardware Virtual Machine virtualization. Allows the guest VM to run as though it's on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. See Also PV virtualization.

I

IAM	See AWS Identity and Access Management (IAM).
IAM group	See group.
IAM policy simulator	See policy simulator.
IAM role	See role.

IAM user	See user.
Identity and Access Management	See AWS Identity and Access Management (IAM).
identity provider (IdP)	An IAM (p. 135) entity that holds metadata about external identity providers.
IdP	See identity provider (IdP) .
image	See Amazon Machine Image (AMI).
import/export station	A machine that uploads or downloads your data to or from Amazon S3 (p. 129).
import log	A report that contains details about how AWS Import/Export (p. 135) processed your data.
implicit impressions	Amazon Personalize (p. 128): The recommendations that your application shows a user. Unlike *explicit impressions*, where you manually record each impression, Amazon Personalize automatically derives implicit impressions from your recommendation data. See Also recommendations, impressions data, explicit impressions.
impressions data	Amazon Personalize (p. 128): The list of items that you presented to a user when they interacted with a particular item such as by clicking it, watching it, or purchasing it. Amazon Personalize uses impressions data to calculate the relevance of new items for a user based on how frequently users have selected or ignored the same item. See Also explicit impressions, implicit impressions.
in-place deployment	CodeDeploy: A deployment method where the application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. You can choose to use a load balancer so each instance is deregistered during its deployment and then restored to service after the deployment is complete.
index	See search index.
index field	A name–value pair that's included in an Amazon CloudSearch (p. 124) domain's index. An index field can contain text or numeric data, dates, or a location.
indexing options	Configuration settings that define an Amazon CloudSearch (p. 124) domain's index fields, how document data is mapped to those index fields, and how the index fields can be used.
inline policy	An IAM (p. 135) policy (p. 164) that's embedded in a single IAM user (p. 178), group (p. 152), or role (p. 169).
input data	Amazon Machine Learning: The observations that you provide to Amazon Machine Learning to train and evaluate a machine learning model and generate predictions.
instance	A copy of an Amazon Machine Image (AMI) (p. 128) running as a virtual server in the AWS Cloud.
instance family	A general instance type (p. 155) grouping using either storage or CPU capacity.
instance group	A Hadoop (p. 153) cluster contains one master instance group that contains one master node (p. 159), a core instance group containing one or more core node (p. 144) and an optional task node (p. 176) instance group, which can contain any number of task nodes.

instance profile	A container that passes IAM (p. 135) role (p. 169) information to an EC2 instance (p. 148) at launch.
instance store	Disk storage that's physically attached to the host computer for an EC2 instance (p. 148), and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.
instance store-backed AMI	A type of Amazon Machine Image (AMI) (p. 128) whose instance (p. 154)s use an instance store (p. 155) volume (p. 179) as the root device. Compare this with instances launched from Amazon EBS (p. 126)-backed AMIs, which use an Amazon EBS volume as the root device.
instance type	A specification that defines the memory, CPU, storage capacity, and usage cost for an instance (p. 154). Some instance types are designed for standard applications, whereas others are designed for CPU-intensive, memory-intensive applications, and so on.
Interactions dataset	Amazon Personalize (p. 128): A container for historical and real-time data collected from interactions between users and items (called events). Interactions data can include impressions data and contextual metadata. See Also dataset, event, impressions data, contextual metadata.
internet gateway	Connects a network to the internet. You can route traffic for IP addresses outside your VPC (p. 179) to the internet gateway.
internet service provider (ISP)	A company that provides subscribers with access to the internet. Many ISPs are also mailbox provider (p. 158)s. Mailbox providers are sometimes referred to as ISPs, even if they only provide mailbox services.
intrinsic function	A special action in a AWS CloudFormation (p. 133) template that assigns values to properties not available until runtime. These functions follow the format *Fn::Attribute*, such as `Fn::GetAtt`. Arguments for intrinsic functions can be parameters, pseudo parameters, or the output of other intrinsic functions.
IP address	A numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 148)s are assigned two IP addresses at launch, which are directly mapped to each other through network address translation (NAT (p. 160)): a private IP address (following RFC 1918) and a public IP address. Instances launched in a VPC (p. 130) are assigned only a private IP address. Instances launched in your default VPC are assigned both a private IP address and a public IP address.
IP match condition	AWS WAF (p. 139): An attribute that specifies the IP addresses or IP address ranges that web requests originate from. Based on the specified IP addresses, you can configure AWS WAF to allow or block web requests to AWS resource (p. 168)s such as Amazon CloudFront (p. 124) distributions.
ISP	See internet service provider (ISP).
issuer	The person who writes a policy (p. 164) to grant permissions to a resource (p. 168). The issuer (by definition) is always the resource owner. AWS doesn't permit Amazon SQS (p. 129) users to create policies for resources they don't own. If John is the resource owner, AWS authenticates John's identity when he submits the policy he's written to grant permissions for that resource.
item	A group of attributes that's uniquely identifiable among all of the other items. Items in Amazon DynamoDB (p. 125) are similar in many ways to rows, records, or tuples in other database systems.

item exploration	Amazon Personalize (p. 128): The process that Amazon Personalize uses to test different item recommendations, including recommendations of new items with no or very little interaction data, and learn how users respond. You configure item exploration at the campaign level for solution versions created with the user-personalization recipe. See Also recommendations, campaign, solution version, user-personalization recipe.
item-to-item similarities (SIMS) recipe	Amazon Personalize (p. 128): A RELATED_ITEMS recipe that uses the data from an Interactions dataset to make recommendations for items that are similar to a specified item. The SIMS recipe calculates similarity based on the way users interact with items instead of matching item metadata, such as price or age. See Also recipe, RELATED_ITEMS recipes, Interactions dataset.
Items dataset	Amazon Personalize (p. 128): A container for metadata about items, such as price, genre, or availability. See Also dataset.

J

job flow	Amazon EMR (p. 126): One or more step (p. 174)s that specify all of the functions to be performed on the data.
job ID	A five-character, alphanumeric string that uniquely identifies an AWS Import/Export (p. 135) storage device in your shipment. AWS issues the job ID in response to a CREATE JOB email command.
job prefix	An optional string that you can add to the beginning of an AWS Import/Export (p. 135) log file name to prevent collisions with objects of the same name. See Also key prefix.
JSON	JavaScript Object Notation. A lightweight data interchange format. For information about JSON, see http://www.json.org/.
junk folder	The location where email messages that various filters determine to be of lesser value are collected so that they don't arrive in the recipient (p. 167)'s inbox but are still accessible to the recipient. This is also referred to as a spam (p. 173) or bulk folder.

K

key	A credential that identifies an AWS account (p. 123) or user (p. 178) to AWS (such as the AWS secret access key (p. 171)).
	Amazon Simple Storage Service (Amazon S3) (p. 129), Amazon EMR (p. 126): The unique identifier for an object in a bucket (p. 140). Every object in a bucket

has exactly one key. Because a bucket and key together uniquely identify each object, you can think of Amazon S3 as a basic data map between the *bucket + key*, and the object itself. You can uniquely address every object in Amazon S3 through the combination of the web service endpoint, bucket name, and key, as in this example: `http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl`, where `doc` is the name of the bucket, and `2006-03-01/AmazonS3.wsdl` is the key.

AWS Import/Export (p. 135): The name of an object in Amazon S3. It's a sequence of Unicode characters whose UTF-8 encoding can't exceed 1024 bytes. If a key (for example, logPrefix + import-log-JOBID) is longer than 1024 bytes, AWS Elastic Beanstalk (p. 134) returns an `InvalidManifestField` error.

IAM (p. 135): In a policy (p. 164), a specific characteristic that's the basis for restricting access (such as the current time or the IP address of the requester).

Tagging resources: A general tag (p. 176) label that acts like a category for more specific tag values. For example, you might have EC2 instance (p. 148) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 168) with up to 10 key–value pairs. Not all AWS resources can be tagged.

key pair	A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.
key prefix	A logical grouping of the objects in a bucket (p. 140). The prefix value is similar to a directory name that you can use to store similar data under the same directory in a bucket.
kibibyte (KiB)	A contraction of kilo binary byte, a kibibyte is 2^10 or 1,024 bytes. A kilobyte (KB) is 10^3 or 1,000 bytes. 1,024 KiB is a mebibyte (MiB) (p. 159).
KMS	See AWS Key Management Service (AWS KMS).

L

labeled data	In machine learning, data for which you already know the target or "correct" answer.
launch configuration	A set of descriptive parameters used to create new EC2 instance (p. 148)s in an Amazon EC2 Auto Scaling (p. 125) activity.
	A template that an Auto Scaling group (p. 132) uses to launch new EC2 instances. The launch configuration contains information such as the Amazon Machine Image (AMI) (p. 128) ID, the instance type, key pairs, security group (p. 171)s, and block device mappings, among other configuration settings.
launch permission	An Amazon Machine Image (AMI) (p. 128) attribute that allows users to launch an AMI.
lifecycle	The lifecycle state of the EC2 instance (p. 148) contained in an Auto Scaling group (p. 132). EC2 instances progress through several states over their lifespan; these include *Pending*, *InService*, *Terminating* and *Terminated*.

lifecycle action	An action that can be paused by Auto Scaling, such as launching or terminating an EC2 instance.
lifecycle hook	A feature for pausing Auto Scaling after it launches or terminates an EC2 instance so that you can perform a custom action while the instance isn't in service.
link to VPC	The process of linking (or attaching) an EC2-Classic instance (p. 154) to a ClassicLink-enabled VPC (p. 179). See Also ClassicLink, unlink from VPC.
load balancer	A DNS name combined with a set of ports, which together provide a destination for all requests intended for your application. A load balancer can distribute traffic to multiple application instances across every Availability Zone (p. 132) within a Region (p. 167). Load balancers can span multiple Availability Zones within an AWS Region into which an Amazon EC2 (p. 125) instance was launched. But load balancers can't span multiple Regions.
local secondary index	An index that has the same partition key as the table, but a different sort key. A local secondary index is local in the sense that every partition of a local secondary index is scoped to a table partition that has the same partition key value. See Also local secondary index.
logical name	A case-sensitive unique string within an AWS CloudFormation (p. 133) template that identifies a resource (p. 168), mapping (p. 159), parameter, or output. In an AWS CloudFormation template, each parameter, resource (p. 168), property, mapping, and output must be declared with a unique logical name. You use the logical name when dereferencing these items using the Ref function.

M

Mail Transfer Agent (MTA)	Software that transports email messages from one computer to another by using a client-server architecture.
mailbox provider	An organization that provides email mailbox hosting services. Mailbox providers are sometimes referred to as internet service provider (ISP) (p. 155)s, even if they only provide mailbox services.
mailbox simulator	A set of email addresses that you can use to test an Amazon SES (p. 129)-based email-sending application without sending messages to actual recipients. Each email address represents a specific scenario (such as a bounce or complaint) and generates a typical response that's specific to the scenario.
main route table	The default route table (p. 169) that any new VPC (p. 179) subnet (p. 175) uses for routing. You can associate a subnet with a different route table of your choice. You can also change which route table is the main route table.
managed policy	A standalone IAM (p. 135) policy (p. 164) that you can attach to multiple user (p. 178)s, group (p. 152)s, and role (p. 169)s in your IAM account (p. 123). Managed policies can either be AWS managed policies (which are created and managed by AWS) or customer managed policies (which you create and manage in your AWS account).
manifest	When sending a *create job* request for an import or export operation, you describe your job in a text file called a manifest. The manifest file is a YAML-formatted

file that specifies how to transfer data between your storage device and the AWS Cloud.

manifest file	Amazon Machine Learning: The file used for describing batch predictions. The manifest file relates each input data file with its associated batch prediction results. It's stored in the Amazon S3 output location.
mapping	A way to add conditional parameter values to an AWS CloudFormation (p. 133) template. You specify mappings in the template's optional Mappings section and retrieve the desired value using the FN::FindInMap function.
marker	See pagination token.
master node	A process running on an Amazon Machine Image (AMI) (p. 128) that keeps track of the work its core and task nodes complete.
maximum price	The maximum price you will pay to launch one or more Spot Instance (p. 173)s. If your maximum price exceeds the current Spot price (p. 173) and your restrictions are met, Amazon EC2 (p. 125) launches instances on your behalf.
maximum send rate	The maximum number of email messages that you can send per second using Amazon SES (p. 129).
mean reciprocal rank at 25	Amazon Personalize (p. 128): An evaluation metric that assesses the relevance of a model's highest ranked recommendation. Amazon Personalize calculates this metric using the average accuracy of the model when ranking the most relevant recommendation out of the top 25 recommendations over all requests for recommendations. See Also metrics, recommendations.
mebibyte (MiB)	A contraction of mega binary byte, a mebibyte is 2^20 or 1,048,576 bytes. A megabyte (MB) is 10^6 or 1,000,000 bytes. 1,024 MiB is a gibibyte (GiB) (p. 152).
member resources	See resource.
message ID	Amazon Simple Email Service (Amazon SES) (p. 129): A unique identifier that's assigned to every email message that's sent. Amazon Simple Queue Service (Amazon SQS) (p. 129): The identifier returned when you send a message to a queue.
metadata	Information about other data or objects. In Amazon Simple Storage Service (Amazon S3) (p. 129) and Amazon EMR (p. 126) metadata takes the form of name–value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata (for example, Content-Type). Users can also specify custom metadata at the time they store an object. In Amazon EC2 (p. 125) metadata includes data about an EC2 instance (p. 148) that the instance can retrieve to determine things about itself, such as the instance type or the IP address.
metric	An element of time-series data defined by a unique combination of exactly one namespace (p. 160), exactly one metric name, and between zero and ten dimensions. Metrics and the statistics derived from them are the basis of Amazon CloudWatch (p. 124).
metrics	Amazon Personalize (p. 128): Evaluation data that Amazon Personalize generates when you train a model. You use metrics to evaluate the performance of the model, view the effects of modifying a solution's configuration, and compare results between solutions that use the same training data but were created with different recipes.

See Also solution, recipe.

metric name	The primary identifier of a metric, used in combination with a namespace (p. 160) and optional dimensions.
MFA	See multi-factor authentication (MFA).
micro instance	A type of EC2 instance (p. 148) that's more economical to use if you have occasional bursts of high CPU activity.
MIME	See Multipurpose Internet Mail Extensions (MIME).
ML model	In machine learning (ML), a mathematical model that generates predictions by finding patterns in data. Amazon Machine Learning supports three types of ML models: binary classification, multiclass classification, and regression. Also known as a *predictive model*. See Also binary classification model, multiclass classification model, regression model.
MTA	See Mail Transfer Agent (MTA).
Multi-AZ deployment	A primary DB instance (p. 146) that has a synchronous standby replica in a different Availability Zone (p. 132). The primary DB instance is synchronously replicated across Availability Zones to the standby replica.
multiclass classification model	A machine learning model that predicts values that belong to a limited, pre-defined set of permissible values. For example, "Is this product a book, movie, or clothing?"
multi-factor authentication (MFA)	An optional AWS account (p. 123) security feature. Once you enable AWS MFA, you must provide a six-digit, single-use code in addition to your sign-in credentials whenever you access secure AWS webpages or the AWS Management Console (p. 136). You get this single-use code from an authentication device that you keep in your physical possession. See Also https://aws.amazon.com/mfa/.
multi-valued attribute	An attribute with more than one value.
multipart upload	A feature that you can use to upload a single object as a set of parts.
Multipurpose Internet Mail Extensions (MIME)	An internet standard that extends the email protocol to include non-ASCII text and nontext elements, such as attachments.
Multitool	A cascading application that provides a simple command-line interface for managing large datasets.

N

namespace	An abstract container that provides context for the items (names, or technical terms, or words) it holds, and allows disambiguation of homonym items residing in different namespaces.
NAT	Network address translation. A strategy of mapping one or more IP addresses to another while data packets are in transit across a traffic routing device. This

	is commonly used to restrict internet communication to private instances while allowing outgoing traffic. See Also Network Address Translation and Protocol Translation, NAT gateway, NAT instance.
NAT gateway	A NAT (p. 160) device, managed by AWS, that performs network address translation in a private subnet (p. 175), to secure inbound internet traffic. A NAT gateway uses both NAT and port address translation. See Also NAT instance.
NAT instance	A NAT (p. 160) device, configured by a user, that performs network address translation in a VPC (p. 179) public subnet (p. 175) to secure inbound internet traffic. See Also NAT gateway.
network ACL	An optional layer of security that acts as a firewall for controlling traffic in and out of a subnet (p. 175). You can associate multiple subnets with a single network ACL (p. 122), but a subnet can be associated with only one network ACL at a time.
Network Address Translation and Protocol Translation	(NAT (p. 160)-PT) An internet protocol standard defined in RFC 2766. See Also NAT instance, NAT gateway.
n-gram processor	A processor that performs n-gram transformations. See Also n-gram transformation.
n-gram transformation	Amazon Machine Learning: A transformation that aids in text string analysis. An n-gram transformation takes a text variable as input and outputs strings by sliding a window of size *n* words, where *n* is specified by the user, over the text, and outputting every string of words of size *n* and all smaller sizes. For example, specifying the n-gram transformation with window size =2 returns all the two-word combinations and all of the single words.
NICE Desktop Cloud Visualization	A remote visualization technology for securely connecting users to graphic-intensive 3D applications hosted on a remote, high-performance server.
node	Amazon OpenSearch Service (OpenSearch Service) (p. 126): An OpenSearch instance. A node can be either a data instance or a dedicated master instance. See Also dedicated master node.
NoEcho	A property of AWS CloudFormation (p. 133) parameters that prevent the otherwise default reporting of names and values of a template parameter. Declaring the `NoEcho` property causes the parameter value to be masked with asterisks in the report by the `cfn-describe-stacks` command.
normalized discounted cumulative gain (NCDG) at K (5/10/25)	Amazon Personalize (p. 128): An evaluation metric that tells you about the relevance of your model's highly ranked recommendations, where K is a sample size of 5, 10, or 25 recommendations. Amazon Personalize calculates this by assigning weight to recommendations based on their position in a ranked list, where each recommendation is discounted (given a lower weight) by a factor dependent on its position. The normalized discounted cumulative gain at K assumes that recommendations that are lower on a list are less relevant than recommendations higher on the list. See Also metrics, recommendations.
NoSQL	Nonrelational database systems that are highly available, scalable, and optimized for high performance. Instead of the relational model, NoSQL databases (for example, Amazon DynamoDB (p. 125)) use alternate models for data management, such as key–value pairs or document storage.

null object

A null object is one whose version ID is null. Amazon S3 (p. 129) adds a null object to a bucket (p. 140) when versioning (p. 179) for that bucket is suspended. It's possible to have only one null object for each key in a bucket.

number of passes

The number of times that you allow Amazon Machine Learning to use the same data records to train a machine learning model.

O

object

Amazon Simple Storage Service (Amazon S3) (p. 129): The fundamental entity type stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.

Amazon CloudFront (p. 124): Any entity that can be served either over HTTP or a version of RTMP.

observation

Amazon Machine Learning: A single instance of data that Amazon Machine Learning (Amazon ML) uses to either train a machine learning model how to predict or to generate a prediction. Each row in an Amazon ML input data file is an observation.

On-Demand Instance

An Amazon EC2 (p. 125) pricing option that charges you for compute capacity by the hour or second (minimum of 60 seconds) with no long-term commitment.

operation

An API function. Also called an *action*.

optimistic locking

A strategy to ensure that an item that you want to update has not been modified by others before you perform the update. For Amazon DynamoDB (p. 125), optimistic locking support is provided by the AWS SDKs.

organization

AWS Organizations (p. 136): An entity that you create to consolidate and manage your AWS accounts. An organization has one management account along with zero or more member accounts.

organizational unit

AWS Organizations (p. 136): A container for accounts within a root (p. 169) of an organization. An organizational unit (OU) can contain other OUs.

origin access identity

Also called OAI. When using Amazon CloudFront (p. 124) to serve content with an Amazon S3 (p. 129) bucket (p. 140) as the origin, a virtual identity that you use to require users to access your content through CloudFront URLs instead of Amazon S3 URLs. Usually used with CloudFront private content (p. 165).

origin server

The Amazon S3 (p. 129) bucket (p. 140) or custom origin containing the definitive original version of the content you deliver through CloudFront (p. 124).

original environment

The instances in a deployment group at the start of an CodeDeploy blue/green deployment.

OSB transformation

Orthogonal sparse bigram transformation. In machine learning, a transformation that aids in text string analysis and that's an alternative to the n-gram transformation. OSB transformations are generated by sliding the window of size *n* words over the text, and outputting every pair of words that includes the first word in the window.
See Also n-gram transformation.

OU	See organizational unit.
output location	Amazon Machine Learning: An Amazon S3 location where the results of a batch prediction are stored.

P

pagination	The process of responding to an API request by returning a large list of records in small separate parts. Pagination can occur in the following situations:
	• The client sets the maximum number of returned records to a value below the total number of records.
	• The service has a default maximum number of returned records that's lower than the total number of records.
	When an API response is paginated, the service sends a subset of the large list of records and a pagination token that indicates that more records are available. The client includes this pagination token in a subsequent API request, and the service responds with the next subset of records. This continues until the service responds with a subset of records and no pagination token, indicating that all records have been sent.
pagination token	A marker that indicates that an API response contains a subset of a larger list of records. The client can return this marker in a subsequent API request to retrieve the next subset of records until the service responds with a subset of records and no pagination token, indicating that all records have been sent. See Also pagination.
paid AMI	An Amazon Machine Image (AMI) (p. 128) that you sell to other Amazon EC2 (p. 125) users on AWS Marketplace (p. 136).
paravirtual virtualization	See PV virtualization.
part	A contiguous portion of the object's data in a multipart upload request.
partition key	A simple primary key, composed of one attribute (also known as a *hash attribute*). See Also partition key, sort key.
PAT	Port address translation.
pebibyte (PiB)	A contraction of peta binary byte, a pebibyte is 2^50 or 1,125,899,906,842,624 bytes. A petabyte (PB) is 10^15 or 1,000,000,000,000,000 bytes. 1,024 PiB is an exbibyte (EiB) (p. 150).
period	See sampling period.
permission	A statement within a policy (p. 164) that allows or denies access to a particular resource (p. 168). You can state any permission in the following way: "A has permission to do B to C." For example, Jane (A) has permission to read messages (B) from John's Amazon SQS (p. 129) queue (C). Whenever Jane sends a request to Amazon SQS to use John's queue, the service checks to see if she has permission. It further checks to see if the request satisfies the conditions John set forth in the permission.

persistent storage	A data storage solution where the data remains intact until it's deleted. Options within AWS (p. 130) include: Amazon S3 (p. 129), Amazon RDS (p. 129), Amazon DynamoDB (p. 125), and other services.
PERSONALIZED_RANKING recipes	Amazon Personalize (p. 128): Recipes that provide item recommendations in ranked order based on the predicted interest for a user. See Also recipe, recommendations, personalized-ranking recipe, popularity-count recipe.
personalized-ranking recipe	Amazon Personalize (p. 128): A PERSONALIZED_RANKING recipe that ranks a collection of items that you provide based on the predicted interest level for a specific user. Use the personalized-ranking recipe to create curated lists of items or ordered search results that are personalized for a specific user. See Also recipe, PERSONALIZED_RANKING recipes.
physical name	A unique label that AWS CloudFormation (p. 133) assigns to each resource (p. 168) when creating a stack (p. 174). Some AWS CloudFormation commands accept the physical name as a value with the --physical-name parameter.
pipeline	AWS CodePipeline (p. 133): A workflow construct that defines the way software changes go through a release process.
plaintext	Information that has not been encrypted (p. 149), as opposed to ciphertext (p. 141).
policy	IAM (p. 135): A document defining permissions that apply to a user, group, or role; the permissions in turn determine what users can do in AWS. A policy typically allow (p. 124)s access to specific actions, and can optionally grant that the actions are allowed for specific resource (p. 168)s, such as EC2 instance (p. 148)s or Amazon S3 (p. 129) bucket (p. 140)s. Policies can also explicitly deny (p. 146) access. Amazon EC2 Auto Scaling (p. 125): An object that stores the information needed to launch or terminate instances for an Auto Scaling group. Running the policy causes instances to be launched or terminated. You can configure an alarm (p. 123) to invoke an Auto Scaling policy.
policy generator	A tool in the IAM (p. 135) AWS Management Console (p. 136) that helps you build a policy (p. 164) by selecting elements from lists of available options.
policy simulator	A tool in the IAM (p. 135) AWS Management Console (p. 136) that helps you test and troubleshoot policies (p. 164) so you can see their effects in real-world scenarios.
policy validator	A tool in the IAM (p. 135) AWS Management Console (p. 136) that examines your existing IAM access control policies (p. 164) to ensure that they comply with the IAM policy grammar.
popularity-count recipe	Amazon Personalize (p. 128): A USER_PERSONALIZATION recipe that recommends the items that have had the most interactions with unique users. See Also recipe, USER_PERSONALIZATION recipes.
precision at K (5/10/25)	Amazon Personalize (p. 128): An evaluation metric that tells you how relevant your model's recommendations are based on a sample size of K (5, 10, or 25) recommendations. Amazon Personalize calculates this metric based on the number of relevant recommendations out of the top K recommendations, divided by K, where K is 5, 10, or 25. See Also metrics, recommendations.

prefix	See job prefix.
Premium Support	A one-on-one, fast-response support channel that AWS customers can subscribe to for support for AWS infrastructure services. See Also https://aws.amazon.com/premiumsupport/.
presigned URL	A web address that uses query string authentication (p. 166).
primary key	One or two attributes that uniquely identify each item in a Amazon DynamoDB (p. 125) table, so that no two items can have the same key. See Also partition key, sort key.
primary shard	See shard.
principal	The user (p. 178), service, or account (p. 123) that receives permissions that are defined in a policy (p. 164). The principal is A in the statement "A has permission to do B to C."
private content	When using Amazon CloudFront (p. 124) to serve content with an Amazon S3 (p. 129) bucket (p. 140) as the origin, a method of controlling access to your content by requiring users to use signed URLs. Signed URLs can restrict user access based on the current date and time, the IP addresses that the requests originate from, or both.
private IP address	A private numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 148)ss are assigned two IP addresses at launch, which are directly mapped to each other through network address translation (NAT (p. 160)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 130) are assigned only a private IP address.
private subnet	A VPC (p. 179) subnet (p. 175) whose instances can't be reached from the internet.
product code	An identifier provided by AWS when you submit a product to AWS Marketplace (p. 136).
properties	See resource property.
property rule	A JSON (p. 156)-compliant markup standard for declaring properties, mappings, and output values in an AWS CloudFormation (p. 133) template.
Provisioned IOPS	A storage option designed to deliver fast, predictable, and consistent I/O performance. When you specify an IOPS rate while creating a DB instance, Amazon RDS (p. 129) provisions that IOPS rate for the lifetime of the DB instance.
pseudo parameter	A predefined setting (for example, `AWS:StackName`) that can be used in AWS CloudFormation (p. 133) templates without having to declare them. You can use pseudo parameters anywhere you can use a regular parameter.
public AMI	An Amazon Machine Image (AMI) (p. 128) that all AWS accounts (p. 123) have permission to launch.
public dataset	A large collection of public information that can be seamlessly integrated into applications that are based in the AWS Cloud. Amazon stores public datasets at no charge to the community and, similar to other AWS services, users pay only for the compute and storage they use for their own applications. These datasets currently include data from the Human Genome Project, the US Census, Wikipedia, and other sources.

See Also https://aws.amazon.com/publicdatasets.

public IP address
A public numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). EC2 instance (p. 148)s are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT (p. 160)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 130) are assigned only a private IP address.

public subnet
A subnet (p. 175) whose instances can be reached from the internet.

PV virtualization
Paravirtual virtualization. Allows guest VMs to run on host systems that don't have special support extensions for full hardware and CPU virtualization. Because PV guests run a modified operating system that doesn't use hardware emulation, they can't provide hardware-related features, such as enhanced networking or GPU support.
See Also HVM virtualization.

Q

quartile binning transformation
Amazon Machine Learning: A process that takes two inputs, a numerical variable and a parameter called a bin number, and outputs a categorical variable. Quartile binning transformations discover non-linearity in a variable's distribution by enabling the machine learning model to learn separate importance values for parts of the numeric variable's distribution.

Query
A type of web service that generally uses only the GET or POST HTTP method and a query string with parameters in the URL.
See Also REST.

query string authentication
An AWS feature that you can use to place the authentication information in the HTTP request query string instead of in the `Authorization` header, which provides URL-based access to objects in a bucket (p. 140).

queue
A sequence of messages or jobs that are held in temporary storage awaiting transmission or processing.

queue URL
A web address that uniquely identifies a queue.

quota
The maximum value for your resources, actions, and items in your AWS account

R

range GET
A request that specifies a byte range of data to get for a download. If an object is large, you can break up a download into smaller units by sending multiple range GET requests that each specify a different byte range to GET.

raw email	A type of *sendmail* request with which you can specify the email headers and MIME types.
RDS	See Amazon Relational Database Service (Amazon RDS).
read replica	Amazon RDS (p. 129): An active copy of another DB instance. Any updates to the data on the source DB instance are replicated to the read replica DB instance using the built-in replication feature of MySQL 5.1.
real-time predictions	Amazon Machine Learning: Synchronously generated predictions for individual data observations. See Also batch prediction.
recipe	Amazon Personalize (p. 128): An Amazon Personalize algorithm that's preconfigured to predict the items that a user will interact with (for USER_PERSONALIZATION recipes), or calculate items that are similar to specific items that a user has shown interest in (for RELATED_ITEMS recipes), or rank a collection of items that you provide based on the predicted interest for a specific user (for PERSONALIZED_RANKING recipes). See Also USER_PERSONALIZATION recipes, RELATED_ITEMS recipes, PERSONALIZED_RANKING recipes.
recommendations	Amazon Personalize (p. 128): A list of items that Amazon Personalize predicts that a user will interact with. Depending on the Amazon Personalize recipe used, recommendations can be either a list of items (with USER_PERSONALIZATION recipes and RELATED_ITEMS recipes), or a ranking of a collection of items you provided (with PERSONALIZED_RANKING recipes). See Also recipe, campaign, solution version, USER_PERSONALIZATION recipes, RELATED_ITEMS recipes, PERSONALIZED_RANKING recipes.
receipt handle	Amazon SQS (p. 129): An identifier that you get when you receive a message from the queue. This identifier is required to delete a message from the queue or when changing a message's visibility timeout.
receiver	The entity that consists of the network systems, software, and policies that manage email delivery for a recipient (p. 167).
recipient	Amazon Simple Email Service (Amazon SES) (p. 129): The person or entity receiving an email message. For example, a person named in the "To" field of a message.
Redis	A fast, open-source, in-memory key-value data structure store. Redis comes with a set of versatile in-memory data structures with which you can easily create a variety of custom applications.
reference	A means of inserting a property from one AWS resource (p. 168) into another. For example, you could insert an Amazon EC2 (p. 125) security group (p. 171) property into an Amazon RDS (p. 129) resource.
Region	A named set of AWS resource (p. 168)s in the same geographical area. A Region comprises at least two Availability Zone (p. 132)s.
regression model	Amazon Machine Learning: Preformatted instructions for common data transformations that fine-tune machine learning model performance.
regression model	A type of machine learning model that predicts a numeric value, such as the exact purchase price of a house.
regularization	A machine learning (ML) parameter that you can tune to obtain higher-quality ML models. Regularization helps prevent ML models from memorizing training

	data examples instead of learning how to generalize the patterns it sees (called overfitting). When training data is overfitted, the ML model performs well on the training data, but doesn't perform well on the evaluation data or on new data.
RELATED_ITEMS recipes	Amazon Personalize (p. 128)Recipes that recommend items that are similar to a specified item, such as the item-to-item (SIMS) recipe. See Also recipe, item-to-item similarities (SIMS) recipe.
replacement environment	The instances in a deployment group after the CodeDeploy blue/green deployment.
replica shard	See shard.
reply path	The email address that an email reply is sent to. This is different from the return path (p. 169).
representational state transfer	See REST.
reputation	1. An Amazon SES (p. 129) metric, based on factors that might include bounce (p. 140)s, complaint (p. 142)s, and other metrics, regarding whether or not a customer is sending high-quality email. 2. A measure of confidence, as judged by an internet service provider (ISP) (p. 155) or other entity that an IP address that they are receiving email from isn't the source of spam (p. 173).
requester	The person (or application) that sends a request to AWS to perform a specific action. When AWS receives a request, it first evaluates the requester's permissions to determine whether the requester is allowed to perform the request action (if applicable, for the requested resource (p. 168)).
Requester Pays	An Amazon S3 (p. 129) feature that allows a bucket owner (p. 140) to specify that anyone who requests access to objects in a particular bucket (p. 140) must pay the data transfer and request costs.
reservation	A collection of EC2 instance (p. 148)s started as part of the same launch request. Not to be confused with a Reserved Instance (p. 168).
Reserved Instance	A pricing option for EC2 instance (p. 148)s that discounts the on-demand (p. 162) usage charge for instances that meet the specified parameters. Customers pay for the entire term of the instance, regardless of how they use it.
Reserved Instance Marketplace	An online exchange that matches sellers who have reserved capacity that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instance (p. 168)s that you purchase from third-party sellers have less than a full standard term remaining and can be sold at different upfront prices. The usage or reoccurring fees remain the same as the fees set when the Reserved Instances were originally purchased. Full standard terms for Reserved Instances available from AWS run for one year or three years.
resource	An entity that users can work with in AWS, such as an EC2 instance (p. 148), an Amazon DynamoDB (p. 125) table, an Amazon S3 (p. 129) bucket (p. 140), an IAM (p. 135) user, or an AWS OpsWorks (p. 136) stack (p. 174).
resource property	A value required when including an AWS resource (p. 168) in an AWS CloudFormation (p. 133) stack (p. 174). Each resource can have one or more properties associated with it. For example, an `AWS::EC2::Instance` resource might have a `UserData` property. In an AWS CloudFormation template, resources must declare a properties section, even if the resource has no properties.

resource record	Also called *resource record set*. The fundamental information elements in the Domain Name System (DNS). See Also Domain Name System in Wikipedia.
REST	Representational state transfer. A simple stateless architecture that generally runs over HTTPS/TLS. REST emphasizes that resources have unique and hierarchical identifiers (URIs), are represented by common media types (such as HTML, XML, or JSON (p. 156)), and that operations on the resources are either predefined or discoverable within the media type. In practice, this generally results in a limited number of operations. See Also Query, WSDL, SOAP.
RESTful web service	Also known as RESTful API. A web service that follows REST (p. 169) architectural constraints. The API operations must use HTTP methods explicitly; expose hierarchical URIs; and transfer either XML, JSON (p. 156), or both.
return enabled	Amazon CloudSearch (p. 124): An index field option that enables the field's values to be returned in the search results.
return path	The email address that bounced email is returned to. The return path is specified in the header of the original email. This is different from the reply path (p. 168).
revision	AWS CodePipeline (p. 133): A change made to a source that's configured in a source action, such as a pushed commit to a GitHub (p. 152) repository or an update to a file in a versioned Amazon S3 (p. 129) bucket (p. 140).
role	A tool for giving temporary access to AWS resource (p. 168)s in your AWS account (p. 123).
rollback	A return to a previous state that follows the failure to create an object, such as AWS CloudFormation (p. 133) stack (p. 174). All resource (p. 168)s associated with the failure are deleted during the rollback. For AWS CloudFormation, you can override this behavior using the `--disable-rollback` option on the command line.
root	AWS Organizations (p. 136): A parent container for the accounts in your organization. If you apply a service control policy (p. 171) to the root, it applies to every organizational unit (p. 162) and account in the organization.
root credentials	Authentication information associated with the AWS account (p. 123) owner.
root device volume	A volume (p. 179) that contains the image used to boot the instance (p. 154) (also known as a *root device*). If you launched the instance from an AMI (p. 128) backed by instance store (p. 155), this is an instance store volume (p. 179) created from a template stored in Amazon S3 (p. 129). If you launched the instance from an AMI backed by Amazon EBS (p. 126), this is an Amazon EBS volume created from an Amazon EBS snapshot.
route table	A set of routing rules that controls the traffic leaving any subnet (p. 175) that's associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.
row identifier	Amazon Machine Learning: An attribute in the input data that you can include in the evaluation or prediction output to make it easier to associate a prediction with an observation.
rule	AWS WAF (p. 139): A set of conditions that AWS WAF searches for in web requests to AWS resource (p. 168)s such as Amazon CloudFront (p. 124) distributions. You add rules to a web ACL (p. 180), and then specify whether you want to allow or block web requests based on each rule.

S

S3	See Amazon Simple Storage Service (Amazon S3).
sampling period	A defined duration of time, such as one minute, which Amazon CloudWatch (p. 124) computes a statistic (p. 174) over.
sandbox	A testing location where you can test the functionality of your application without affecting production, incurring charges, or purchasing products. Amazon SES (p. 129): An environment that's designed for developers to test and evaluate the service. In the sandbox, you have full access to the Amazon SES API, but you can only send messages to verified email addresses and the mailbox simulator. To get out of the sandbox, you need to apply for production access. Accounts in the sandbox also have lower sending limits (p. 171) than production accounts.
scale in	To remove EC2 instances from an Auto Scaling group (p. 132).
scale out	To add EC2 instances to an Auto Scaling group (p. 132).
scaling policy	A description of how Auto Scaling should automatically scale an Auto Scaling group (p. 132) in response to changing demand. See Also scale in, scale out.
scaling activity	A process that changes the size, configuration, or makeup of an Auto Scaling group (p. 132) by launching or terminating instances.
scheduler	The method used for placing task (p. 176)s on container instance (p. 143)s.
schema	Amazon Machine Learning: The information needed to interpret the input data for a machine learning model, including attribute names and their assigned data types, and the names of special attributes.
score cut-off value	Amazon Machine Learning: A binary classification model outputs a score that ranges from 0 to 1. To decide whether an observation should be classified as 1 or 0, you pick a classification threshold, or cut-off, and Amazon ML compares the score against it. Observations with scores higher than the cut-off are predicted as target equals 1, and scores lower than the cut-off are predicted as target equals 0.
SCP	See service control policy.
search API	Amazon CloudSearch (p. 124): The API that you use to submit search requests to a search domain (p. 170).
search domain	Amazon CloudSearch (p. 124): Encapsulates your searchable data and the search instances that handle your search requests. You typically set up a separate Amazon CloudSearch domain for each different collection of data that you want to search.
search domain configuration	Amazon CloudSearch (p. 124): A domain's indexing options, analysis scheme (p. 130)s, expression (p. 151)s, suggester (p. 175)s, access policies, and scaling and availability options.

search enabled	Amazon CloudSearch (p. 124): An index field option that enables the field data to be searched.
search endpoint	Amazon CloudSearch (p. 124): The URL that you connect to when sending search requests to a search domain. Each Amazon CloudSearch domain has a unique search endpoint that remains the same for the life of the domain.
search index	Amazon CloudSearch (p. 124): A representation of your searchable data that facilitates fast and accurate data retrieval.
search instance	Amazon CloudSearch (p. 124): A compute resource (p. 168) that indexes your data and processes search requests. An Amazon CloudSearch domain has one or more search instances, each with a finite amount of RAM and CPU resources. As your data volume grows, more search instances or larger search instances are deployed to contain your indexed data. When necessary, your index is automatically partitioned across multiple search instances. As your request volume or complexity increases, each search partition is automatically replicated to provide additional processing capacity.
search request	Amazon CloudSearch (p. 124): A request that's sent to an Amazon CloudSearch domain's search endpoint to retrieve documents from the index that match particular search criteria.
search result	Amazon CloudSearch (p. 124): A document that matches a search request. Also referred to as a *search hit*.
secret access key	A key that's used in conjunction with the access key ID (p. 122) to cryptographically sign programmatic AWS requests. Signing a request identifies the sender and prevents the request from being altered. You can generate secret access keys for your AWS account (p. 123), individual IAM user (p. 178)s, and temporary sessions.
security group	A named set of allowed inbound network connections for an instance. (Security groups in Amazon VPC (p. 130) also include support for outbound connections.) Each security group consists of a list of protocols, ports, and IP address ranges. A security group can apply to multiple instances, and multiple groups can regulate a single instance.
sender	The person or entity sending an email message.
Sender ID	A Microsoft-controlled version of SPF (p. 173). An email authentication and anti-spoofing system. For more information about Sender ID, see Sender ID in Wikipedia.
sending limits	The sending quota (p. 171) and maximum send rate (p. 159) that are associated with every Amazon SES (p. 129) account.
sending quota	The maximum number of email messages that you can send using Amazon SES (p. 129) in a 24-hour period.
server-side encryption (SSE)	The encrypting (p. 149) of data at the server level. Amazon S3 (p. 129) supports three modes of server-side encryption: SSE-S3, where Amazon S3 manages the keys; SSE-C, where the customer manages the keys; and SSE-KMS, where AWS Key Management Service (AWS KMS) (p. 136) manages keys.
service control policy	AWS Organizations (p. 136): A policy-based control that specifies the services and actions that users and roles can use in the accounts that the service control policy (SCP) affects.
service endpoint	See endpoint.

service health dashboard	A webpage showing up-to-the-minute information about AWS service availability. The dashboard is located at http://status.aws.amazon.com/.
Service Quotas	A service for viewing and managing your quotas easily and at scale as your AWS workloads grow. Quotas, also referred to as limits, are the maximum number of resources that you can create in an AWS account.
service role	An IAM (p. 135) role (p. 169) that grants permissions to an AWS service so it can access AWS resource (p. 168)s. The policies that you attach to the service role determine which AWS resources the service can access and what it can do with those resources.
SES	See Amazon Simple Email Service (Amazon SES).
session	The period when the temporary security credentials provided by AWS Security Token Service (AWS STS) (p. 138) allow access to your AWS account.
SHA	Secure Hash Algorithm. SHA1 is an earlier version of the algorithm, which AWS has replaced with SHA256.
shard	Amazon OpenSearch Service (OpenSearch Service) (p. 126): A partition of data in an index. You can split an index into multiple shards, which can include primary shards (original shards) and replica shards (copies of the primary shards). Replica shards provide failover, which means that a replica shard is promoted to a primary shard if a cluster node that contains a primary shard fails. Replica shards also can handle requests.
shared AMI	An Amazon Machine Image (AMI) (p. 128) that a developer builds and makes available for others to use.
shutdown action	Amazon EMR (p. 126): A predefined bootstrap action that launches a script that runs a series of commands in parallel before terminating the job flow.
signature	Refers to a *digital signature*, which is a mathematical way to confirm the authenticity of a digital message. AWS uses signatures to authenticate the requests you send to our web services. For more information, to https://aws.amazon.com/security.
SIGNATURE file	AWS Import/Export (p. 135): A file you copy to the root directory of your storage device. The file contains a job ID, manifest file, and a signature.
Signature Version 4	Protocol for authenticating inbound API requests to AWS services in all AWS Regions.
Simple Mail Transfer Protocol	See SMTP.
Simple Object Access Protocol	See SOAP.
Simple Storage Service	See Amazon Simple Storage Service (Amazon S3).
SIMS recipe	See item-to-item similarities (SIMS) recipe.
Single Sign-On	See AWS Single Sign-On.
Single-AZ DB instance	A standard (non-Multi-AZ) DB instance (p. 146) that's deployed in one Availability Zone (p. 132), without a standby replica in another Availability Zone. See Also Multi-AZ deployment.
sloppy phrase search	A search for a phrase that specifies how close the terms must be to one another to be considered a match.

SMTP	Simple Mail Transfer Protocol. The standard that's used to exchange email messages between internet hosts for the purpose of routing and delivery.
snapshot	Amazon Elastic Block Store (Amazon EBS) (p. 126): A backup of your volume (p. 179)s that's stored in Amazon S3 (p. 129). You can use these snapshots as the starting point for new Amazon EBS volumes or to protect your data for long-term durability. See Also DB snapshot.
SNS	See Amazon Simple Notification Service (Amazon SNS).
SOAP	Simple Object Access Protocol. An XML-based protocol that you can use to exchange information over a particular protocol (for example, HTTP or SMTP) between applications. See Also REST, WSDL.
soft bounce	A temporary email delivery failure such as one resulting from a full mailbox.
software VPN	A software appliance-based VPN connection over the internet.
solution	Amazon Personalize (p. 128): The recipe, customized parameters, and trained models (solution versions) that can be used to generate recommendations. See Also recipe, solution version, recommendations.
solution version	Amazon Personalize (p. 128): A trained model that you create as part of a solution in Amazon Personalize. You deploy a solution version in a campaign to generate recommendations. See Also solution, campaign, recommendations.
sort enabled	Amazon CloudSearch (p. 124): An index field option that enables a field to be used to sort the search results.
sort key	An attribute used to sort the order of partition keys in a composite primary key (also known as a *range attribute*). See Also partition key, primary key.
source/destination checking	A security measure to verify that an EC2 instance (p. 148) is the origin of all traffic that it sends and the ultimate destination of all traffic that it receives; that is, that the instance isn't relaying traffic. Source/destination checking is turned on by default. For instances that function as gateways, such as VPC (p. 179) NAT (p. 160) instances, source/destination checking must be disabled.
spam	Unsolicited bulk email.
spamtrap	An email address that's set up by an anti-spam (p. 173) entity, not for correspondence, but to monitor unsolicited email. This is also called a *honeypot*.
SPF	Sender Policy Framework. A standard for authenticating email.
Spot Instance	A type of EC2 instance (p. 148) that you can bid on to take advantage of unused Amazon EC2 (p. 125) capacity.
Spot price	The price for a Spot Instance (p. 173) at any given time. If your maximum price exceeds the current price and your restrictions are met, Amazon EC2 (p. 125) launches instances on your behalf.
SQL injection match condition	AWS WAF (p. 139): An attribute that specifies the part of web requests (such as a header or a query string) that AWS WAF inspects for malicious SQL code. Based on the specified conditions, you can configure AWS WAF to allow or block web

	requests to an AWS resource (p. 168), such as an Amazon CloudFront (p. 124) distribution.
SQS	See Amazon Simple Queue Service (Amazon SQS).
SSE	See server-side encryption (SSE).
SSL	Secure Sockets Layer See Also Transport Layer Security (TLS).
SSO	See AWS Single Sign-On.
stack	AWS CloudFormation (p. 133): A collection of AWS resources that you create and delete as a single unit. AWS OpsWorks (p. 136): A set of instances that you manage collectively, typically because they have a common purpose such as serving PHP applications. A stack serves as a container and handles tasks that apply to the group of instances as a whole, such as managing applications and cookbooks.
station	AWS CodePipeline (p. 133): A portion of a pipeline workflow where one or more actions are performed.
station	A place at an AWS facility where your AWS Import/Export data is transferred on to, or off of, your storage device.
statistic	One of five functions of the values submitted for a given sampling period (p. 170). These functions are `Maximum`, `Minimum`, `Sum`, `Average`, and `SampleCount`.
stem	The common root or substring shared by a set of related words.
stemming	The process of mapping related words to a common stem. This enables matching on variants of a word. For example, a search for "horse" could return matches for horses, horseback, and horsing, as well as horse. Amazon CloudSearch (p. 124) supports both dictionary based and algorithmic stemming.
step	Amazon EMR (p. 126): A single function applied to the data in a job flow (p. 156). The sum of all steps comprises a job flow.
step type	Amazon EMR (p. 126): The type of work done in a step. There are a limited number of step types, such as moving data from Amazon S3 (p. 129) to Amazon EC2 (p. 125) or from Amazon EC2 to Amazon S3.
sticky session	A feature of the Elastic Load Balancing (p. 148) load balancer that binds a user's session to a specific application instance so that all requests coming from the user during the session are sent to the same application instance. By contrast, a load balancer defaults to route each request independently to the application instance with the smallest load.
stopping	The process of filtering stop words from an index or search request.
stopword	A word that isn't indexed and is automatically filtered out of search requests because it's either insignificant or so common that including it would result in too many matches to be useful. Stopwords are language specific.
streaming	Amazon EMR (p. 126): A utility that comes with Hadoop (p. 153) that you can use to develop MapReduce executables in languages other than Java. Amazon CloudFront (p. 124): The ability to use a media file in real time—as it's transmitted in a steady stream from a server.

streaming distribution	A special kind of distribution (p. 147) that serves streamed media files using a Real Time Messaging Protocol (RTMP) connection.
Streams	See Amazon Kinesis Data Streams.
string-to-sign	Before you calculate an HMAC (p. 153) signature, you first assemble the required components in a canonical order. The preencrypted string is the string-to-sign.
string match condition	AWS WAF (p. 139): An attribute that specifies the strings that AWS WAF searches for in a web request, such as a value in a header or a query string. Based on the specified strings, you can configure AWS WAF to allow or block web requests to an AWS resource (p. 168), such as a CloudFront (p. 124) distribution.
strongly consistent read	A read process that returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful—regardless of the Region. See Also data consistency, eventual consistency, eventually consistent read.
structured query	Search criteria specified using the Amazon CloudSearch (p. 124) structured query language. You use the structured query language to construct compound queries that use advanced search options and combine multiple search criteria using Boolean operators.
STS	See AWS Security Token Service (AWS STS).
subnet	A segment of the IP address range of a VPC (p. 179) that an EC2 instance (p. 148) can be attached to. You can create subnets to group instances according to security and operational needs.
Subscription button	An HTML-coded button that provides an easy way to charge customers a recurring fee.
suggester	Amazon CloudSearch (p. 124): Specifies an index field for getting autocomplete suggestions and options that can enable fuzzy matches and control how suggestions are sorted.
suggestions	Documents that contain a match for the partial search string in the field designated by the suggester (p. 175). Amazon CloudSearch (p. 124) suggestions include the document IDs and field values for each matching document. To be a match, the string must match the contents of the field starting from the beginning of the field.
supported AMI	An Amazon Machine Image (AMI) (p. 128) similar to a paid AMI (p. 163), except that the owner charges for additional software or a service that customers use with their own AMIs.
SWF	See Amazon Simple Workflow Service (Amazon SWF).
symmetric encryption	Encryption (p. 149) that uses a private key only. See Also asymmetric encryption.
synchronous bounce	A type of bounce (p. 140) that occurs while the email servers of the sender (p. 171) and receiver (p. 167) are actively communicating.
synonym	A word that's the same or nearly the same as an indexed word and that should produce the same results when specified in a search request. For example, a search for "Rocky Four" or "Rocky 4" should return the fourth *Rocky* movie. This can be done by designating that `four` and `4` are synonyms for `IV`. Synonyms are language specific.

T

table	A collection of data. Similar to other database systems, DynamoDB stores data in tables.
tag	Metadata that you can define and assign to AWS resource (p. 168)s, such as an EC2 instance (p. 148). Not all AWS resources can be tagged.
tagging	Tagging resources: Applying a tag (p. 176) to an AWS resource (p. 168).
	Amazon SES (p. 129): Also called *labeling*. A way to format return path (p. 169) email addresses so that you can specify a different return path for each recipient of a message. You can use tagging to support VERP (p. 179). For example, if Andrew manages a mailing list, he can use the return paths andrew+recipient1@example.net and andrew+recipient2@example.net so that he can determine which email bounced.
target attribute	Amazon Machine Learning (Amazon ML): The attribute in the input data that contains the "correct" answers. Amazon ML uses the target attribute to learn how to make predictions on new data. For example, if you were building a model for predicting the sale price of a house, the target attribute would be "target sale price in USD."
target revision	AWS CodeDeploy (p. 133): The most recent version of the application revision that has been uploaded to the repository and will be deployed to the instances in a deployment group. In other words, the application revision currently targeted for deployment. This is also the revision that will be pulled for automatic deployments.
task	An instantiation of a task definition (p. 176) that's running on a container instance (p. 143).
task definition	The blueprint for your task. Specifies the name of the task (p. 176), revisions, container definition (p. 143)s, and volume (p. 179) information.
task node	An EC2 instance (p. 148) that runs Hadoop (p. 153) map and reduce tasks, but doesn't store data. Task nodes are managed by the master node (p. 159), which assigns Hadoop tasks to nodes and monitors their status. While a job flow is running you can increase and decrease the number of task nodes. Because they don't store data and can be added and removed from a job flow, you can use task nodes to manage the EC2 instance capacity your job flow uses, increasing capacity to handle peak loads and decreasing it later.
	Task nodes only run a TaskTracker Hadoop daemon.
tebibyte (TiB)	A contraction of tera binary byte, a tebibyte is 2^40 or 1,099,511,627,776 bytes. A terabyte (TB) is 10^12 or 1,000,000,000,000 bytes. 1,024 TiB is a pebibyte (PiB) (p. 163).
template format version	The version of an AWS CloudFormation (p. 133) template design that determines the available features. If you omit the `AWSTemplateFormatVersion` section from your template, AWS CloudFormation assumes the most recent format version.

template validation	The process of confirming the use of JSON (p. 156) code in an AWS CloudFormation (p. 133) template. You can validate any AWS CloudFormation template using the `cfn-validate-template` command.
temporary security credentials	Authentication information that's provided by AWS STS (p. 138) when you call an STS API action. Includes an access key ID (p. 122), a secret access key (p. 171), a session (p. 172) token, and an expiration time.
throttling	The automatic restricting or slowing down of a process based on one or more limits. Examples: Amazon Kinesis Data Streams (p. 127) throttles operations if an application (or group of applications operating on the same stream) attempts to get data from a shard at a rate faster than the shard limit. Amazon API Gateway (p. 124) uses throttling to limit the steady-state request rates for a single account. Amazon SES (p. 129) uses throttling to reject attempts to send email that exceeds the sending limits (p. 171).
time-series data	Data provided as part of a metric. The time value is assumed to be when the value occurred. A metric is the fundamental concept for Amazon CloudWatch (p. 124) and represents a time-ordered set of data points. You publish metric data points into CloudWatch and later retrieve statistics about those data points as a time-series ordered dataset.
timestamp	A date/time string in ISO 8601 format.
TLS	See Transport Layer Security (TLS).
tokenization	The process of splitting a stream of text into separate tokens on detectable boundaries such as white space and hyphens.
topic	A communication channel to send messages and subscribe to notifications. It provides an access point for publishers and subscribers to communicate with each other.
Traffic Mirroring	An Amazon VPC feature that you can use to copy network traffic from an elastic network interface of Amazon EC2 instances, and then send it to out-of-band security and monitoring appliances for content inspection, threat monitoring, and troubleshooting. See Also https://aws.amazon.com/vpc/.
training datasource	A datasource that contains the data that Amazon Machine Learning uses to train the machine learning model to make predictions.
transition	AWS CodePipeline (p. 133): The act of a revision in a pipeline continuing from one stage to the next in a workflow.
Transport Layer Security (TLS)	A cryptographic protocol that provides security for communication over the internet. Its predecessor is Secure Sockets Layer (SSL).
trust policy	An IAM (p. 135) policy (p. 164) that's an inherent part of an IAM role (p. 169). The trust policy specifies which principals are allowed to use the role.
trusted key groups	Amazon CloudFront key groups whose public keys CloudFront can use to verify the signatures of CloudFront signed URLs and signed cookies.
trusted signers	See trusted key groups (p. 177).
tuning	Selecting the number and type of AMIs (p. 128) to run a Hadoop (p. 153) job flow most efficiently.
tunnel	A route for transmission of private network traffic that uses the internet to connect nodes in the private network. The tunnel uses encryption and secure

protocols such as PPTP to prevent the traffic from being intercepted as it passes through public routing nodes.

U

unbounded	The number of potential occurrences isn't limited by a set number. This value is often used when defining a data type that's a list (for example, `maxOccurs="unbounded"`), in WSDL (p. 180).
unit	Standard measurement for the values submitted to Amazon CloudWatch (p. 124) as metric data. Units include seconds, percent, bytes, bits, count, bytes/second, bits/second, count/second, and none.
unlink from VPC	The process of unlinking (or detaching) an EC2-Classic instance (p. 154) from a ClassicLink-enabled VPC (p. 179). See Also ClassicLink, link to VPC.
usage report	An AWS record that details your usage of a particular AWS service. You can generate and download usage reports from https://aws.amazon.com/usage-reports/.
user	A person or application under an account (p. 123) that needs to make API calls to AWS products. Each user has a unique name within the AWS account, and a set of security credentials not shared with other users. These credentials are separate from the security credentials for the AWS account. Each user is associated with one and only one AWS account.
Users dataset	Amazon Personalize (p. 128): A container for metadata about your users, such as age, gender, or loyalty membership. See Also dataset.
user-personalization recipe	Amazon Personalize (p. 128): An HRNN-based USER_PERSONALIZATION recipe that predicts the items that a user will interact with. The user-personalization recipe can use item exploration and impressions data to generate recommendations for new items. See Also HRNN, recipe, USER_PERSONALIZATION recipes, item exploration, impressions data, recommendations.
USER_PERSONALIZATION recipes	Amazon Personalize (p. 128): Recipes used to build a recommendation system that predicts the items that a user will interact with based on data provided in Interactions, Items, and Users datasets. See Also recipe, user-personalization recipe, popularity-count recipe, HRNN.

V

validation	See template validation.

value	Instances of attributes (p. 131) for an item, such as cells in a spreadsheet. An attribute might have multiple values.
	Tagging resources: A specific tag (p. 176) label that acts as a descriptor within a tag category (key). For example, you might have EC2 instance (p. 148) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 168) with up to 10 key–value pairs. Not all AWS resources can be tagged.
Variable Envelope Return Path	See VERP.
verification	The process of confirming that you own an email address or a domain so that you can send email from or to it.
VERP	Variable Envelope Return Path. A way that email-sending applications can match bounce (p. 140)d email with the undeliverable address that caused the bounce by using a different return path (p. 169) for each recipient. VERP is typically used for mailing lists. With VERP, the recipient's email address is embedded in the address of the return path, which is where bounced email is returned. This makes it possible to automate the processing of bounced email without having to open the bounce messages, which might vary in content.
versioning	Every object in Amazon S3 (p. 129) has a key and a version ID. Objects with the same key, but different version IDs can be stored in the same bucket (p. 140). Versioning is enabled at the bucket layer using PUT Bucket versioning.
VGW	See virtual private gateway (VGW).
virtualization	Allows multiple guest virtual machines (VM) to run on a host operating system. Guest VMs can run on one or more levels above the host hardware, depending on the type of virtualization. See Also PV virtualization, HVM virtualization.
virtual private cloud	See VPC.
virtual private gateway (VGW)	The Amazon side of a VPN connection (p. 180) that maintains connectivity. The internal interfaces of the virtual private gateway connect to your VPC (p. 179) through the VPN attachment. The external interfaces connect to the VPN connection, which leads to the customer gateway (p. 144).
visibility timeout	The period of time that a message is invisible to the rest of your application after an application component gets it from the queue. During the visibility timeout, the component that received the message usually processes it, and then deletes it from the queue. This prevents multiple components from processing the same message.
VM Import/Export	A service for importing virtual machine (VM) images from your existing virtualization environment to Amazon EC2 and then exporting them back. See Also https://aws.amazon.com/ec2/vm-import.
volume	A fixed amount of storage on an instance (p. 154). You can share volume data between more than one container (p. 143) and persist the data on the container instance (p. 143) when the containers are no longer running.
VPC	Virtual private cloud. An elastic network populated by infrastructure, platform, and application services that share common security and interconnection.
VPC endpoint	A feature that you can use to create a private connection between your VPC (p. 179) and another AWS service without requiring access over the

internet, through a NAT (p. 160) instance, a VPN connection (p. 180), or AWS Direct Connect (p. 134).

VPG	See virtual private gateway (VGW).
VPN CloudHub	See AWS VPN CloudHub.
VPN connection	Amazon Web Services (AWS) (p. 130): The IPsec connection between a VPC (p. 179) and some other network, such as a corporate data center, home network, or colocation facility.

W

WAM	See Amazon WorkSpaces Application Manager (Amazon WAM).
web access control list (web ACL)	AWS WAF (p. 139): A set of rules that defines the conditions that AWS WAF searches for in web requests to an AWS resource (p. 168), such as a Amazon CloudFront (p. 124) distribution. A web access control list (web ACL) specifies whether to allow, block, or count the requests.
Web Services Description Language	See WSDL.
WSDL	Web Services Description Language. A language used to describe the actions that a web service can perform, along with the syntax of action requests and responses. See Also REST, SOAP.

X, Y, Z

X.509 certificate	A digital document that uses the X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the entity described in the certificate (p. 141).
yobibyte (YiB)	A contraction of yotta binary byte, a yobibyte is 2^80 or 1,208,925,819,614,629,174,706,176 bytes. A yottabyte (YB) is 10^24 or 1,000,000,000,000,000,000,000,000 bytes.
zebibyte (ZiB)	A contraction of zetta binary byte, a zebibyte is 2^70 or 1,180,591,620,717,411,303,424 bytes. A zettabyte (ZB) is 10^21 or 1,000,000,000,000,000,000,000 bytes. 1,024 ZiB is a yobibyte (YiB) (p. 180).
zone awareness	Amazon OpenSearch Service (OpenSearch Service) (p. 126): A configuration that distributes nodes in a cluster across two Availability Zone (p. 132)s in the same Region. Zone awareness helps to prevent data loss and minimizes downtime in the event of node and data center failure. If you enable zone awareness, you must have an even number of data instances in the instance count, and you also must use the Amazon OpenSearch Service Configuration API to replicate your data for your OpenSearch cluster.