
AWS Snowcone User Guide

AWS Snowcone



AWS Snowcone User Guide: AWS Snowcone

Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is AWS Snowcone?	1
Use Cases	1
Pricing	2
How AWS Snowcone Works	3
AWS Snowcone Workflow	3
How Import and Export Jobs Work	5
Online Data Transfer Between Snowcone and AWS Storage Services with DataSync	5
Offline Data Transfer Between Snowcone and Amazon S3	5
Exporting Jobs from Amazon S3	6
How Export Jobs Work	6
Using Export Ranges	7
Export Range Examples	7
For Import Job Storage	9
For Compute Job Storage	9
Device Specifications	10
Summary	10
Hardware and Network	11
Top View	11
Rear Panel	11
Front Panel	12
AWS Snowcone Wi-Fi Specifications	13
Ruggedization Specifications	13
Snowcone Power Supply and Accessories	14
Disk and CPU Performance	14
Setting Up	15
Sign Up for AWS	15
Before You Order	16
About the Local Environment	16
Working with Special Characters	16
Amazon EC2	17
Pricing for Compute Instances on Snowcone	18
Prerequisites	18
Creating a Linux AMI from an Instance	18
Creating a Linux AMI from a Snapshot	18
Getting Started	21
Creating a Snowcone Job	21
Step 1: Plan Your Job	22
Step 2: Choose Your Shipping Preferences	22
Step 3: Choose Your Job Details	23
Step 4: Choose Your Security Preferences	23
Step 5: Choose Your Notification Preferences	27
Step 6: Download AWS OpsHub	28
Step 7: Review and Create Your Job	28
Using AWS OpsHub to Manage Devices	30
Unlocking a device	30
Unlocking a device locally	31
Unlocking a device remotely	31
Verifying the signature of AWS OpsHub (optional)	32
Managing AWS services	34
Using Compute Instances Locally	35
Using NFS file share to upload files	38
Using AWS DataSync to transfer files to AWS	40
Transferring files	41
Using AWS IoT Greengrass on EC2 instances	42

Setting up your Amazon EC2 instance	42
Managing Your Devices	43
Rebooting your device	43
Shutting down your device	44
Editing Your Device Alias	44
Getting Updates	44
Managing Profiles	45
Setting the NTP time servers for your device	45
Using the AWS Snowball Edge Client	47
Downloading and Installing the Snowball Edge Client	47
Commands for the Snowball Edge Client	47
Configuring a Profile for the Snowball Edge Client	48
Getting Your QR Code for NFC Validation	49
Unlocking an AWS Snowcone Device	49
Updating a Snowcone	49
Getting Credentials	52
Starting a Service on Your Snowcone Device	52
Stopping a Service on Your Snowcone Device	53
Getting Your Certificate for Transferring Data	53
AWS Snowcone Logs	54
Getting Device Status	55
Getting Service Status	56
Launching the AWS DataSync AMI	57
Starting NFS and Restricting Access	59
Restricting Access to NFS Shares When NFS is Running	60
Getting the Export Path for an Amazon S3 Bucket	60
Enabling Local AWS Operator Debugging	60
Disabling Local AWS Operator Debugging	60
Creating a Direct Network Interface	61
Getting Information About a Direct Network Interface	61
Updating a Direct Network Interface	61
Deleting a Direct Network Interface	62
Checking feature status	62
Changing feature status	62
Setting Time Servers	63
Checking Time Sources	63
Using Snow Device Management to Manage Devices	65
Managing devices remotely	66
Enabling Snow Device Management	66
Snow Device Management CLI commands	68
Create a task	69
Check task status	70
Check device info	70
Check Amazon EC2 instance state	72
Check task metadata	73
Cancel a task	74
List commands and syntax	75
List remote-manageable devices	75
List task status across devices	76
List available resources	77
List device or task tags	78
List tasks by status	78
Apply tags	79
Remove tags	79
Using AWS Services	81
Using Amazon EC2 for Compute	81
Overview	81

Prerequisites	82
Creating a Job with Compute Instances	82
Network Configuration for Compute Instances	83
Connecting to Your Compute Instance on a Snowcone Using SSH	88
Snowcone Client Commands for Compute Instances	88
Using the Amazon EC2 Endpoint	92
Autostarting Amazon EC2 Instances with Launch Templates	102
Using Block Storage with Your EC2 Instances	103
Security Groups in Snow Devices	104
Supported Instance Metadata and User Data	104
Troubleshooting Amazon EC2	106
Using DataSync to Transfer Files	107
Using NFS for Offline Data Transfer	108
NFS Configuration for Snowball Edge	108
Troubleshooting NFS Issues	110
Ports Required to Use AWS Services	111
Returning the Snowcone Device	113
Disconnect the Snowcone Device	113
Protecting Data on Your Device	114
Securing Your AWS Snowcone	114
Validating NFC Tags	114
Understanding Job Statuses	116
Understanding the Ordering Process	117
Understanding the Shipment Process	117
Shipping a Snowcone Device	117
Using the AWS Management Console	117
Ordering the Snowcone from the Console	117
Using the Job Management API	117
Common Uses of JMAPI	118
JMAPI Required Strings	118
JMAPI Endpoints	118
JMAPI CLI Commands	118
Examples	119
EC2 Jobs	120
Configuring an AMI to Use SSH to Connect to Compute Instances Launched on the Device	120
Creating Your Job Using the Console	121
Creating Your Job Using the AWS CLI	121
Shipping Considerations	122
Preparing an AWS Snowcone Device for Shipping	122
Region-Based Shipping Restrictions	122
Shipping an AWS Snowcone Device	117
Shipping Carriers	123
Best Practices	129
Security	129
Network	129
Resource Management	130
Managing EC2 Instances	130
Performance	131
Snowcone Quotas	132
Compute Resources Quotas	132
Limitations for Shipping a Snowcone Device	133
Limitations on Processing Your Returned Snowcone Device for Import	133
Available AWS Regions	133
Troubleshooting	134
Troubleshooting Compute Instances	134
Network Problems	134
IP Address is 0.0.0.0	134

Unable to Unlock Device	134
EC2 Instance on Datasync Problems	135
Error: Failed to Launch Instance	135
Data Transfer Issues	135
Access Denied by Server	135
Connection Times Out During Data Transfer	135
Spawn Showmount ENOENT	135
API Reference	137
Document History	138
AWS glossary	141

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.

Note

Wi-Fi is available only in AWS Regions in North America.

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.

Note

Wi-Fi is available only in AWS Regions in North America.

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 \(p. 3\)](#)
- [How Import and Export Jobs Work \(p. 5\)](#)
- [For Import Job Storage \(p. 9\)](#)
- [For Compute Job Storage \(p. 9\)](#)

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. 122).

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. 14).
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. 30).
 - 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. 30).
- 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. 30).
- 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 Flexible Retrieval 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 <i>car</i> is sorted after <i>Car</i> according to UTF-8 binary values.
Aar	Aarr	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 (sb1) 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 (sb1) 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, Wi-Fi, and power supply.

Topics

- [Features and Specifications Summary \(p. 10\)](#)
- [Hardware and Network \(p. 11\)](#)
- [AWS Snowcone Wi-Fi Specifications \(p. 13\)](#)
- [Ruggedization Specifications \(p. 13\)](#)
- [AWS Snowcone Power Supply and Accessories \(p. 14\)](#)
- [Disk and CPU Performance \(p. 14\)](#)

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
Wireless	Wi-Fi Note Wi-Fi is available only in AWS Regions in North America.

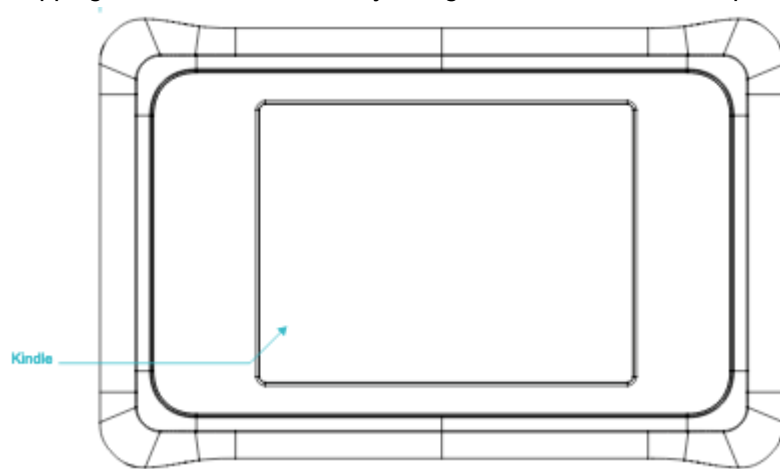
Item	Specification
Number of usable vCPUs	2 vCPUs
Available memory	4 GB
Network interfaces	2x 1/10 gigabit (Gb) - RJ45
AWS DataSync agent pre-installed	Yes
Typical job lifetime	Offline or online data transfer: Days to weeks Edge compute: Weeks to months
Max job length	Edge compute or on-going data transfer: Up to 360 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.

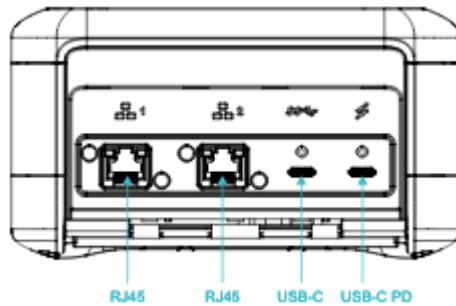
Top View

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 \(p. 14\)](#).

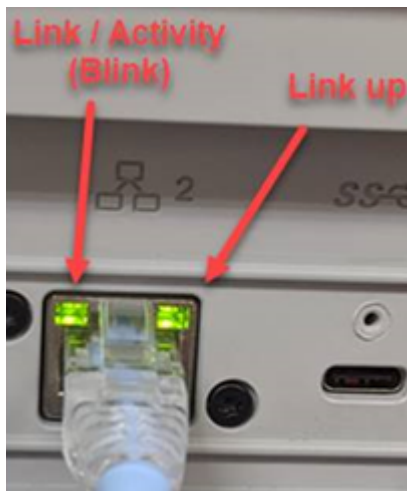
USB

The first USB-C connection is not active.

Ethernet connectors 1 and 2

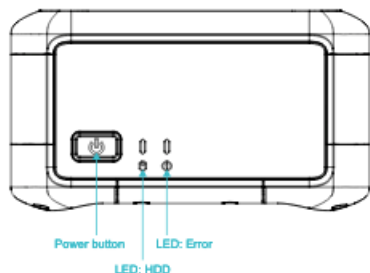
For wired networking, the Snowcone device provides two ports that auto-negotiate for 1 gigabit (Gb) or 10 Gb Ethernet networks.

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.

AWS Snowcone Wi-Fi Specifications

AWS Snowcone supports gigabit Wi-Fi networks with the IEEE 802.11ac standard, and also 802.11abgn networks. To maximize wireless throughput for Snowcone, use 2x2 802.11ac 160 MHz channels, which can be up to 10x faster than baseline 1x1 802.11bgn networks. The Snowcone Wi-Fi specifications are IEEE 802.11abgn+ac, 2x2, MIMO, dual band 2 GHz and 5 G (160MHz). The operating frequencies are 2 GHz and 5 GHz.

You can connect Snowcone to your on-premises network using a physical Ethernet cable, or you can connect it wirelessly using Wi-Fi. With the Wi-Fi connection, you can manage the Snowcone device using AWS OpsHub and transfer data between Snowcone and on-premises storage devices.

Note

Wi-Fi is available only in AWS Regions in North America.

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.

Disk and CPU Performance

AWS Snowcone is a purpose built edge compute/data transfer device. The disk and CPU performance will vary depending on a variety of factors. Some sample performance numbers for transfer are in [Snowcone Performance](#)

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 \(p. 15\)](#)

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.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

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. 122).

International shipments to locations outside of your AWS Region are supported on the AWS Snow Family Management Console for allow-listed 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. 16)
- [Working with Files with Special Characters](#) (p. 16)
- [Using Amazon EC2 on Snowcone](#) (p. 17)

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. 30).

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 \(p. 15\)](#).

To get started with AWS Snowcone, see [Creating an AWS Snowcone Job](#).

Creating an Snowball Edge Job

Regardless of the type of job that you're creating for Snowball Edge, 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 \(p. 22\)](#)
- [Step 2: Choose Your Shipping Preferences \(p. 22\)](#)
- [Step 3: Choose Your Job Details \(p. 23\)](#)
- [Step 4: Choose Your Security Preferences \(p. 23\)](#)
- [Step 5: Choose Your Notification Preferences \(p. 27\)](#)

- [Step 6: Download AWS OpsHub \(p. 28\)](#)
- [Step 7: Review and Create Your Job \(p. 28\)](#)

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 list of pre-existing jobs.
2. If this is your first job, choose **Order an AWS Snow family device**, or if you are expecting multiple jobs to migrate over 500 TB of data, choose **Create your large data migration plan greater than 500 TB**, otherwise choose **Create Job** located in the left navigation bar. Choose **Next step** 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 Snowcone device to you. You connect the device to your local network and configure the device using OpsHub. You copy data onto the device using NFS share, ship it back to AWS, and your data is uploaded to AWS.
 - **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 **Shipping Address** 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)**
 - See [this list of all other shipping speeds](#)
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.

Device Type	Storage (HDD)	Memory	Storage (SSD)	Compute
Snowcone (Selected)	8 TB	4 GB	-	2 vCPUs
Snowcone SSD	-	4 GB	14 TB	2 vCPUs
Snowball Edge Storage Optimized	80 TB	32 GB	-	24 vCPUs
Snowball Edge Compute Optimized	39.5 TB	208 GB	7.68 TB	52 vCPUs
Snowball Edge Compute Optimized with GPU	39.5 TB	208 GB	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 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.
5. 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](#).

6. 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, do one of the following:
- Choose **Create service role** to grant AWS Snow Family permissions to use Amazon S3 and Amazon Simple Notification Service (Amazon SNS) on your behalf. The role grants AWS Security Token Service (AWS STS) AssumeRole trust to the Snow service
 - Choose **Add an existing role to use**, to specify the **IAM role** that you want, or you can use the default role.

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

Add the Condition object and `aws:SourceAccount` or `aws:SourceARN` elements to the access policy to restrict the Snow service so that it acts as only one AWS account numbers. Or add both to be most restrictive.

To restrict the Snow service so that it acts as only one or more account numbers, add one `aws:SourceAccount` element to the access policy and the account ID as the value.

To restrict the Snow service so that it acts as only one or more ARNs, add one `aws:SourceArn` element to the access policy and the ARN as the value.

Example of Condition object to restrict Snow service actions

Example of restricting Snow service actions by ARN and account IDs.

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "AWS_ACCOUNT_ID"
  }
  "ArnLike": {
    "aws:SourceArn": "arn:aws:snowball:REGION:AWS_ACCOUNT_ID:RESOURCE_ID"
  }
}
```

The following shows these conditions included in a policy.

```
{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
```

```
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish"
    ],
    "Resource": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:sns:us-east-1:555555555555:my-sns-topic"
        }
    }
}
]
```

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:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET2"
      ]
    }
  ]
}
```

```
{
  "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"
}
```

Note

You can modify the trust relationship and restrict access to this role based on the customer account number and source arn. See [Restricting Access to the Snow Role Policy \(p. 26\)](#) on how to modify the trust relationship to restrict access.

3. Choose **Next**. If the selected **IAM role** has defined a restricted access, the **Create Job** procedure will fail if the access criteria is not met.
4. Choose **Allow**.
5. Choose **Next**.

Restricting Access to the Snow Role Policy

You can restrict access to the selected role based on the customer account number and source arn.

1. In the navigation pane of the IAM console, choose **Roles**. The console displays the roles for your account.
2. Choose the name of the role that you want to modify, and select the **Trust relationships** tab on the details page.
3. Choose **Edit trust relationships**. Update the trust policy to one of the following:

To restrict access by **customer account number**:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "importexport.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```



```
}
```

To restrict access by **source arn**:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "importexport.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:snowball:REGION:555555555555:RESOURCE_ID"
      }
    }
  }]
}
```

To restrict access by both **customer account number** and **source arn**:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "importexport.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:snowball:REGION:111122223333:RESOURCE_ID"
      }
    }
  }]
}
```

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 \(p. 30\)](#).

To download and install AWS OpsHub for Snow Family

1. In the [AWS Snowball resources](#), click AWS OpsHub. In the AWS OpsHub section with the Download links, choose the appropriate download link to install AWS OpsHub for your operating system.
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 Snow device types. 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 \(p. 30\)](#)
- [Verifying the signature of AWS OpsHub \(optional\) \(p. 32\)](#)
- [Managing AWS services on your device \(p. 34\)](#)
- [Using DataSync to transfer files to AWS \(p. 40\)](#)
- [Using AWS IoT Greengrass to run pre-installed software on Amazon EC2 instances \(p. 42\)](#)
- [Managing Your Devices \(p. 43\)](#)
- [Setting the NTP time servers for your device \(p. 45\)](#)

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](#) (p. 66).

Topics

- [Unlocking a device locally](#) (p. 31)
- [Unlocking a device remotely](#) (p. 31)

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 your Snow Family devices type, and then choose **Sign in**.

If you don't have a device, you can order one.

6. On the **Sign in** 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. 45).
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. 34).

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. 65).

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 \(p. 34\)](#).

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.gpg`.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
xsFNBF/hGf8BEAC9HCDV8u1jDX02Jxspi6kmPu4xqf4ZZLQsSqJcHU61oL/c
/zAN+mUqJT9aJ1r0QFGVD1bMogecUPf1TWldKEEpG8ZbX5P8vR+EE10/iW/
WtqizSudy6qy59ZRK+YVSDx7DZyuJmI07j00UADCL+95ZQN9vqwHNjBHsgfQ
l/1Tqhy81ozTZXCi/+u+99YLaugJIP6ZYIEdfpxnghqyVtaappBFTAyfG67Y
N/5mea1VqJzd81iFpIFQn1+X7U2x6emDbM01yJWV3aMmPwhtQ7iBdt5a4x82
EF5bZJ8HSMvANDILD/9VTN8VfUQGKFjFY2GdX9ERwvFTb47bbv9Z28V1284
4lw2w1B1007Fo02v/Y0ukrN3VHCpmJQS1IiqZbYRa0DVK6UR5QNvU1j5fwWs
4qW9UDPhT/HDuaMrMFCeJEn/7wvRUrGvtzCT9F56A1/dwRSxBeljQQEb1AC8j
uuyi7gJaPdyNntROEFTD7i02L6X2jB4YLfvGxP7Xeq1Y37t8NKF8CYTp0ry/
Wvw0iKZFbo4AkiI0aLyBCK9HBXhUKa9x06g0nhh1UFQrPGrk60RPQKqL76HA
E2ewzGDa90w1RBUAt2nRQpyNYjoASBvz/cAr3e0nuWsIzopZienrxI5ffcjY
f6UWA/OK3ITHtYHewVhseDyEqTQ4MUIWQS4NAwARAQABzT1BV1MgT3BzSHVi
IGZvcibTbm93IEZhbWlseSA8YXdzLW9wc2h1Y11zaWduZXJAYW1hem9uLmNv
bT7CwY0EEAEIACAFA1/hGf8GCwkHCAMCBBUICgIEFgIBAAIZAQAIBAwIeAQAh
CRAHgc9adPNF8RYhBDCvpe1IaY930b0vqiGBz1p080XxGbcP+gPZX7LzKc1Y
w9CT3UHgkAIaw0SXYktujzoYVxAz8/j3jEkCY0dKnfyqvWZDiJAXnzmXWWbg
cxg1g0GXNXCM41Ad68CmbA0LoLTaWSQX30ZbswzhbtX2ADA1opV8RLBik7fm
bS9FyuuDRhFYRQq0fpjUGXFIEgwG6aMFxsrgLl4QD7t+6ftFie/mxLbjR4
iMgtr8FIPXbgn05YYY/LeF4NIgX4iLEqRbAnfWjPzqQ1spFWAotIzDmZqby+
WdWThrH4K1rwtYM8sDhqRnMnqJrGFZzk7aDhVPwF+FOVMmPeEN5JRazEeUr1
VZaSw6mu0n4FMGSXuwGgdvmkqnMe6I5/xLdU4IOPNhp0UmakDW0q/a1dREDE
ZLQMqDMInphmeQno4inGmwbRo63gitD4ZNR5sWwfuwty25l08Ekv7jkkp3mSv
pdxn5tptttnPaSPcSIX/4ED119Tu0i7aup+v30t7eikYDSZG6g9+jHB3Va9e
/VWShfSgy8Jm2+qq/ujUQDAGTCfSuY9jg1ITSog6ayEza/2upDJ1m+40HK4p
8DzEzP/3jTahT8q5ofFWSRDL17d31TSU+JBmPE3mz311FNXgi08w+taY320z
+irHtb3iSiukbjS8s0maVgzszRqS9mhaEn4LL0zoqrUicmXgTyFB7n2LuYv
07vxM05xxhGQwsF2BBABCAAJBQJf4RoCAhsDACEJEBFZvzT/tDi5FIEEi+09
V+UAYN9Gnw36EVm/NP+00LnnEQ/+J4C0Mn8j0AebXrwBiFs83sQo2q+WHL1S
MRc1g5gRFDXs6h1Gv+TGXRen7j1oeaddWvg0tUBxqmC0jr+8AKH00tiBWSu0
```

```
lsS8JU5rindEsKUrKTwcG2wyZFoe1z1E8xPkLRSRN5ZbbgKsTz1611HgCCId
Do+WJdDkWGwXmtDvzjM32EI/PVBd108ga9aPwXdhLwOdKAjZ4JrJXLUQJjRI
IVDSyMObEH0UM6a/+mWNZazNfo0LsGwqGva6Xn5WJWlwr1S78vPNf03BQYu0
YRjaVQR+kPtB9aSAZNi5sWfk6NrRNd1Q78d067uhhejsjRt7Mja2fEL4Kb1X
nK4U/ps7X103o/VjblneZ0hJK6kAKU172tnPJTJ31Jb0xX73wsMWDYZRZvK
9X9+GFrpwhKHKKPjpM0t/FRxNepvqR172TkgBPqGH2TM0FdB1f/uQprvqge
PBbS0JrmBIH9/anIqgtMdtcNQB/0erLdCDqI5af0uD10LcLwdJwG9/bSrfwT
TVEE3WbXmJ8pZgMzlHUiZE6V2DSadV/YItk50IOjJR0VH0Hv1FMwGCEAIFzf
9P/pNi8hpEmI RphRiOVVcdQ30bH0M0gPhu5V9f1IhyCL1zU3LjYTHkq0yJD5
YDA1x01MYq3DcSM5130VBblmuVS2GpcsTCYqlgQA6h/zzMwz+/70wU0EX+EZ
/wEQA0AY8ULmcJIQWIr14V0jylpJeD3qwj7wd+QsBzJ+m0pOB/3ZFAhQiN01
9yCD1HeiZeAmWYX90IXrNiIdcHy+WTAp4G+NaMpqE52qhbDjz+IbvLp1lyDH
bYEHPjnTHXEy21bvKAJ0Kkw/2RcQ0i4dodGnq5icyYj+9gcuHvnVwbrQ96Ia
0D7c+b5T+bzFqk90nIcztrMRuhDLJnJpi70jpvQwfq/TkkZA+mzupxfSkq/Y
N9qXNEToT/VI2gn/LS0X4Ar112KxBjzNEsQkwGsiWSYtMA5J+Tj5ED0uZ/qe
omNblA1D4bm7Na8NAoLxCTAiDq/f3To9Xb181Hsnd0mFLCb/BVgP4edQKTIi
C/OZHy9QJ1fmN0aq7JVLQAuvQNEL88RKW6YZBqkPd3P6zdc7sWDLTMXM0d3I
e6NUvU7pW0E9NyRfUF+oT4s9wAJhAodinAi8Zi9rEfHk1VCJ76j7bcQqYZe0
jXD3IJ7T+X2XA8M/BmypoMW0Soljzhwh044RAasr/fAzpKNPB318JwcQunIz
u2N3CeJ+zrsomjcPxzehwsSVq11zaL2ureJBL0KkBgYxUJYXpbS01ax1TsFG
091dAN0s9Ej8CND37GsNnuygj0gWxbX6MNgbvPs3H3zi/AbMunQ1VB1w07JX
zdM1hBQZh6w+NeiEsK1T6wHi7IhxABEBAHCwXYEGAEIAAKFAL/hGf8CGwwA
IQKQIYHPWnTzRfEWIQ3L6XpSGmPd9Gzr6ohgc9adPNF8TMBD/9TbU/+PVbF
ywkvw13GL01pY7BXn81QaHyunMGUavm080faRR0ynkH0ZqLHCp6bIajF0fvF
b7c0Jamzx8Hg+SIDl6yRpRY+fA4RQ6PNnmT93ZgWW3EbJPyJGlm0/r03SR
+0yn4/ldlg2KfBX4pqMoPCMkUdWxGmDETXsGihwZ0gmCZqXe81K122PYkSN
JQQ+L1fJkVcaxfPKEjXYTbIbfyyhCR6NzA0VZxCrzS2xDrYWP/V002K1xda
0ix6r2aEHf+xYEUh0aBt80HY5nXTuRReCVU789MUVtCMqD2u6amdo4BR0kWA
QNg4yavKwV+LVtyYh2Iju9VSyv4xL1Q4xKHvcAUrSH73bHG7b7jkUJckD0f4
twhjJk/Lfwe6RdnVo2WoeTvE93w+Naq2FXmnbG7e1t10XfQecvQU3QNbRvH
U8B96W0w8UXJdvTKg4f0NbJsw7iJ3x5naixQ+rA8hLV8x0gn2LX6wvxT/SEU
mn20KX+fPtJELK7v/NheFLX1jsKLXYo4jHrkfIXNsNUhg/x2E71kAjbET3s+
t9kCtxt2iXDDZvpIbmG04QkvLFvOR0aSmN6+8fupe3e+e2yN0e6xGTUE60gX
I2+X1p1g9IduDYTpOI20X1eHyyMqGEeIb4g0iisloTp5oi3EuAYRGf1XuqAT
VA19bKnkpBsJ0A==
=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

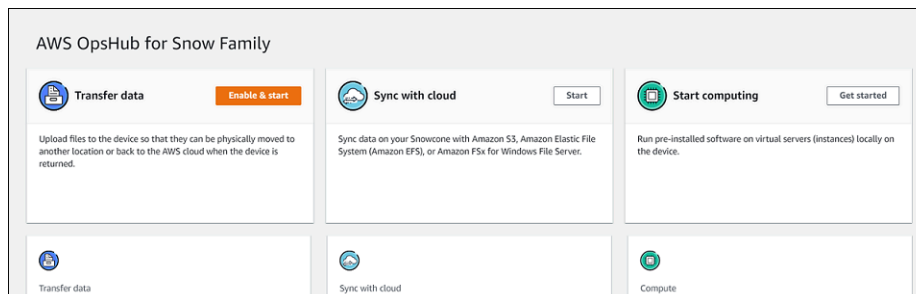
```
gpg: Signature made Mon Dec 21 13:44:47 2020 PST
gpg:                using RSA key 1655BBDE2B770256
gpg: Good signature from "AWS OpsHub for Snow Family <aws-opshub-
signer@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication 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 \(p. 35\)](#)
- [Using NFS file shares to manage file storage \(p. 38\)](#)

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 \(p. 35\)](#)
- [Stopping an Amazon EC2 instance \(p. 36\)](#)
- [Starting an Amazon EC2 instance \(p. 36\)](#)
- [Working with key pairs \(p. 37\)](#)
- [Terminating an Amazon EC2 instance \(p. 37\)](#)
- [Using storage volumes locally \(p. 37\)](#)

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 be 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 \(p. 37\)](#).
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

- [Mounting NFS on a Windows client \(p. 38\)](#)
- [Configuring NFS automatically \(quick setup\) \(p. 39\)](#)
- [Configuring NFS manually \(p. 39\)](#)
- [Stopping data transfer \(p. 40\)](#)

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 than 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 DHCP 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 nolog rsize=128 wsize=128 mtype=hard ipaddress:/buckets/BucketName *`
 - Linux: `mount -t nfs ipaddress:/buckets/BucketName mount_point`
 - macOS: `mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 ipaddress:/buckets/$bucketname 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 Snowcone device as a source location for DataSync. 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 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 \(p. 43\)](#)
- [Shutting down your device \(p. 44\)](#)
- [Editing Your Device Alias \(p. 44\)](#)
- [Getting Updates for Your Device and the AWS OpsHub Application \(p. 44\)](#)
- [Managing Profiles \(p. 45\)](#)

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 **Device Power** menu, then choose **Reboot**. A dialog box appears.
3. In the dialog box, choose **Reboot**. Your device starts to reboot.

Shutting down your device

Follow these steps to use AWS OpsHub to shut down your Snow device.

Important

We highly recommend that you suspend all activities on the device before you shut down the device. Shutting down 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 shut down 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 **Device Power** menu, then choose **Shutdown**. A dialog box appears.
3. In the dialog box, choose **Shutdown**. Your device starts to shut down.

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 \(p. 30\)](#) 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**.

Setting the NTP time servers for your device

Follow these steps to view and update which time servers your device must synchronize time with.

To check time sources

1. On the AWS OpsHub dashboard, find your device under **Devices**. Choose the device to open the device details page.
2. You will see a list of time sources that your device is synchronizing time with in the **Time sources** table.

The **Time sources** table has four columns:

- **Address:** The DNS name / IP address of the time source
- **State:** The current connection status between the device and that time source, there are 5 possible states:
 - **CURRENT:** Time source is currently being used to synchronize time
 - **COMBINED:** Time source is combined with the current source
 - **EXCLUDED:** Time source is excluded by the combining algorithm
 - **LOST:** Connection with the time source has been lost
 - **UNAVAILABILITY:** An invalid time source where the combining algorithm has deemed to be either a falseticker or has too much variability
- **Type:** Network Time Protocol (NTP) sources can be a server or peer. A server can be set by the user using the **update-time-server** command, whereas a peer can only be set up using other Snowball Edge devices in the cluster and are automatically set up when the cluster is associated.
- **Stratum:** The stratum of the source. **Stratum 1** indicates a source with a locally attached reference clock. A source that is synchronized to a Stratum 1 source is set at **Stratum 2**. A source that is synchronized to a stratum 2 source is set at **Stratum 3**, and so on.

To update the time servers

1. On the AWS OpsHub dashboard, find your device under **Devices**. Choose the device to open the device details page.
2. You will see a list of time sources that your device is synchronizing time with in the **Time sources** table.
3. Choose **Update time servers** on the **Time sources** table.
4. Provide the DNS name or the IP address of the time servers you would like your device to synchronize time with, and choose **Update**.

Using the AWS 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 \(p. 30\)](#).

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

- [Configuring a Profile for the Snowball Edge Client \(p. 48\)](#)
- [Getting Your QR Code for NFC Validation \(p. 49\)](#)
- [Unlocking an AWS Snowcone Device \(p. 49\)](#)
- [Updating a Snowcone \(p. 49\)](#)
- [Getting Credentials \(p. 52\)](#)
- [Starting a Service on Your Snowcone Device \(p. 52\)](#)
- [Stopping a Service on Your Snowcone Device \(p. 53\)](#)
- [Getting Your Certificate for Transferring Data \(p. 53\)](#)
- [AWS Snowcone Logs \(p. 54\)](#)

- [Getting Device Status \(p. 55\)](#)
- [Getting Service Status \(p. 56\)](#)
- [Launching the AWS DataSync AMI \(p. 57\)](#)
- [Starting NFS and Restricting Access \(p. 59\)](#)
- [Restricting Access to NFS Shares When NFS is Running \(p. 60\)](#)
- [Getting the Export Path for an Amazon S3 Bucket \(p. 60\)](#)
- [Enabling Local AWS Operator Debugging \(p. 60\)](#)
- [Disabling Local AWS Operator Debugging \(p. 60\)](#)
- [Creating a Direct Network Interface \(p. 61\)](#)
- [Getting Information About a Direct Network Interface \(p. 61\)](#)
- [Updating a Direct Network Interface \(p. 61\)](#)
- [Deleting a Direct Network Interface \(p. 62\)](#)
- [Checking feature status \(p. 62\)](#)
- [Changing feature status \(p. 62\)](#)
- [Setting Time Servers \(p. 63\)](#)
- [Checking Time Sources \(p. 63\)](#)

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 \(p. 52\)](#).

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 \(p. 81\)](#).

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 \(p. 114\)](#).

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

Example 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 \(p. 49\)](#).

`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.

For more information see [AWS credentials](#) and [using the Amazon EC2 Endpoint](#).

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 \(p. 89\)](#).

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-JIEXAMPLEf-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-JIEXAMPLEf-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" : "JIDbEXAMPLE-7eed-1234-ABCD-7EXAMPLE123c",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "192.168.1.2"
  },
  "PhysicalNetworkInterfaces" : [ {
    "PhysicalNetworkInterfaceId" : "s.ni-8bEXAMPLE5EXAMPLE",
    "PhysicalConnectorType" : "RJ45",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.168.1.13",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.168.1.1",
    "MacAddress" : "EX:AM:PL:E0:12:34"
  }, {
    "PhysicalNetworkInterfaceId" : "s.ni-84EXAMPLE3EXAMPLE",
    "PhysicalConnectorType" : "RJ45_2",
    "IpAddressAssignment" : "STATIC",
    "IpAddress" : "0.0.0.0",
    "Netmask" : "0.0.0.0",
    "DefaultGateway" : "192.168.1.1",
    "MacAddress" : "EX:AM:PL:E0:12:34"
  }, {
    "PhysicalNetworkInterfaceId" : "s.ni-87EXAMPLE5EXAMPLE",
    "PhysicalConnectorType" : "WIFI",
    "IpAddressAssignment" : "STATIC",
    "IpAddress" : "0.0.0.0",
    "Netmask" : "0.0.0.0",
    "DefaultGateway" : "192.168.1.1",
    "MacAddress" : "EX:AM:PL:E0:12:34"
  } ],
  "DeviceCapacities" : [ {
    "Name" : "HDD Storage",
    "Unit" : "Byte",
```

```

    "Total" : 157242114048,
    "Used" : 81604378624,
    "Available" : 75637735424
  }, {
    "Name" : "SSD Storage",
    "Unit" : "Byte",
    "Total" : 0,
    "Used" : 0,
    "Available" : 0
  }, {
    "Name" : "vCPU",
    "Unit" : "Number",
    "Total" : 3,
    "Used" : 3,
    "Available" : 0
  }, {
    "Name" : "Memory",
    "Unit" : "Byte",
    "Total" : 5368709120,
    "Used" : 5368709120,
    "Available" : 0
  }, {
    "Name" : "GPU",
    "Unit" : "Number",
    "Total" : 0,
    "Used" : 0,
    "Available" : 0
  } ],
  "DeviceType" : "SNC1_HDD"
}

```

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 \(p. 89\)](#). 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"
}
```

Setting Time Servers

You can set up an external Network Time Protocol (NTP) server. You can use the NTP CLI commands when the device is in both locked and unlocked states. The manifest and unlock code are required. You can set these either with the `snowballEdge configure` command or by using the `--manifest-file` and `--unlock-code` options. Note that you can use the `snowballEdge` CLI on both AWS Snowcone Edge and AWS Snowcone.

It is your responsibility to provide a secure NTP time server. To set which NTP time servers the device connects to, use the `update-time-servers` CLI command.

Note

The `update-time-servers` command will override the previous NTP time servers settings.

Usage

```
snowballEdge update-time-servers time.google.com
```

Example Example Output

```
Updating time servers now.
```

Checking Time Sources

To see which NTP time sources the device are currently connected to, use the `describe-time-sources` Snowball Edge CLI command.

Usage

```
snowballEdge describe-time-sources
```

Example Example Output

```
{
  "Sources" : [ {
    "Address" : "172.31.2.71",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.3.203",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.0.178",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.3.178",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "216.239.35.12",
    "State" : "CURRENT",
    "Type" : "SERVER",
    "Stratum" : 1
  } ]
}
```

The `describe-time-sources` command returns a list of time source states. Each time source state contains the `Address`, `State`, `Type`, and `Stratum` fields. Following are the meanings of these fields.

- **Address** – The DNS name / IP address of the time source.
- **State** – The current connection status between the device and that time source. There are five possible states:
 - **CURRENT** – The time source is currently being used to synchronize time.
 - **COMBINED** – The time source is combined with the current source.
 - **EXCLUDED** – The time source is excluded by the combining algorithm.
 - **LOST** – The connection with the time source has been lost.
 - **UNACCEPTABLE** – An invalid time source where the combining algorithm has deemed to be either a falseticker or has too much variability.
- **Type** – An NTP time source can be either a server or a peer. Servers can be set by the `update-time-servers` command. Peers can only be other Snowball Edge devices in the cluster and are automatically set up when the cluster is associated.
- **Stratum** – This field shows the stratum of the source. Stratum 1 indicates a source with a locally attached reference clock. A source that is synchronized to a stratum 1 source is at stratum 2. A source that is synchronized to a stratum 2 source is at stratum 3, and so on..

An NTP time source can either be a server or a peer. A server can be set by the user with the `update-time-servers` command, whereas a peer could only be other Snowball Edge devices in the cluster. In the example output, `describe-time-sources` is called on a Snowball Edge that is in a cluster of 5. The output contains 4 peers and 1 server. The peers have a stratum of 10 while the server has a stratum of 1; therefore, the server is selected to be the current time source.

Using AWS Snow Device Management to Manage Devices

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

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

- When you order a new Snow Family 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 Snow Family 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 Snow Family 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 a Snow Family device that is 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 SN1_HDD \
  --region us-west-2 \
```

Topics

- [Managing devices remotely \(p. 66\)](#)

- [Enabling Snow Device Management](#) (p. 66)
- [Snow Device Management CLI commands](#) (p. 68)

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 Snow Family device arrives and is powered on for the first time.

If you specified `INSTALLED_ONLY` when ordering your 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

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 \(p. 31\)](#).

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 \(p. 69\)](#)
- [Check task status \(p. 70\)](#)
- [Check device info \(p. 70\)](#)
- [Check Amazon EC2 instance state \(p. 72\)](#)
- [Check task metadata \(p. 73\)](#)
- [Cancel a task \(p. 74\)](#)
- [List commands and syntax \(p. 75\)](#)
- [List remote-manageable devices \(p. 75\)](#)
- [List task status across devices \(p. 76\)](#)
- [List available resources \(p. 77\)](#)
- [List device or task tags \(p. 78\)](#)
- [List tasks by status \(p. 78\)](#)
- [Apply tags \(p. 79\)](#)
- [Remove tags \(p. 79\)](#)

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
InternalServerError
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-ficthmqoc2pht1ef \
--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  
InternalServerError  
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": "Byte",  
      "used": 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
InternalServerError
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
InternalServerError
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  
InternalServerError  
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  
InternalServerError  
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
InternalServerError
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
InternalServerError
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  
InternalServerError  
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 K8VAMqKiP2Cf4xGkmH8GMyZrg0F8Fub+d10KTP9+P4pUb+8PhW+6MiXh4= \  
--max-results 10
```

Exceptions

```
ValidationException  
InternalServerError  
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  
InternalServerError  
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  
InternalServerError  
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 AWS Snowball Edge Client \(p. 47\)](#).

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.

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

Topics

- [Using Amazon EC2 Compute Instances \(p. 81\)](#)
- [Using AWS DataSync to Transfer Files \(p. 107\)](#)
- [Using NFS for Offline Data Transfer \(p. 108\)](#)
- [Ports Required to Use AWS Services on an AWS Snowcone device \(p. 111\)](#)

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 \(p. 17\)](#).

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 \(p. 30\)](#).

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. 132).

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. 21).

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. 52).

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.

Note

VNI: All physical interfaces (RJ45, SFP+, and QSFP) are supported.

Direct network interface (DNI)

A direct network interface (DNI) is an advanced network feature that enables use cases like multicast streams, transitive 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 support VLAN tags and customizing the MAC address. Traffic on DNIs is not protected by security groups.

Snowcone devices support eight DNIs per EC2 instance, with a maximum of 63 per device.

On Snowball Edge devices, DNIs can be associated with the SFP or QSFP ports. DNIs cannot be associated with RJ45 ports. Each optical port supports a maximum of seven DNIs. DNIs do not have to be associated to the physical network port you use to control your Snow Family device. One EC2 instance can support four DNIs and another instance can support three, for a maximum of seven per device.

Topics

- [Prerequisites \(p. 84\)](#)
- [Setting Up a Virtual Network Interface \(VNI\) \(p. 85\)](#)
- [Setting Up a Direct Network Interface \(DNI\) \(p. 85\)](#)

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 Snowcone 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 \(p. 84\)](#).

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 \(VNI\) \(p. 85\)](#).

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 \(p. 61\)](#) 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 DHCP or static IP when booting.

The following are examples of shell scripts for Amazon Linux 2 and CentOS Linux that make these configuration changes.

Amazon Linux 2

```
# Mac address of the direct network interface.  
# You got this when you created the direct network interface.  
DNI_MAC=[MAC ADDRESS FROM CREATED DNI]  
  
# Configure routing so that packets meant for the VNI always are sent through eth0.  
PRIVATE_IP=$(curl -s http://169.254.169.254/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
echo "169.254.169.254 dev eth0" >> /etc/sysconfig/network-scripts/route-eth0

# Query the persistent DNI name, assigned by udev via ec2net helper.
#   changable in /etc/udev/rules.d/70-persistent-net.rules
DNI=$(ip --oneline link | grep -i $DNI_MAC | awk -F ':' '{ print $2 }')

# Configure DNI 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
MAINROUTES=no
EOF

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

CentOS Linux

```
# 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=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
PRIVATE_GATEWAY=$(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=$(LANG=C ip -o link | awk -F ':' -v IGNORECASE=1 '{!link~/ieee802.11/ && /'"$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. 82). 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. 83).
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. 83).

```
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 AWS Snowball Edge Client](#) (p. 47).

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. 93).

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 \(p. 92\)](#)
- [Unsupported Amazon EC2 Features for Snowcone \(p. 93\)](#)
- [Supported AWS CLI Commands for Amazon EC2 on a Snowcone \(p. 93\)](#)
- [Supported Amazon EC2 API Operations \(p. 101\)](#)

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 \(p. 53\)](#).
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.snowcone.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 \(p. 104\)](#).
 - `--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"
  }
}
}
}

```

- `[--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 \(p. 92\)](#).
- `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 \(p. 92\)](#).
- `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 [Specifying the Amazon EC2 Endpoint as the AWS CLI Endpoint \(p. 92\)](#).
- **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 \(p. 92\)](#).
- [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 \(p. 92\)](#).
- `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 \(p. 92\)](#).
- [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 \(p. 92\)](#).
- [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 `sbgl`.

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 \(p. 82\)](#).
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 \(p. 93\)](#).

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 \(p. 88\)](#).
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 \(p. 93\)](#).

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 (sbp1) or >performance-optimized SSD (sbp1). The default volume type is sbp1.
- 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.
- 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. 93). For more information about supported API operations, see [Supported Amazon EC2 API Operations](#) (p. 101).

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/ <i>mac</i> /local-hostname	The interface's local hostname.
network/interfaces/macs/ <i>mac</i> /local-ipv4s	The private IPv4 addresses associated with the interface.
network/interfaces/macs/ <i>mac</i> /mac	The instance's MAC address.
network/interfaces/macs/ <i>mac</i> /public-ipv4s	The Elastic IP addresses associated with the interface.
public-ipv4	The public IPv4 address.
public-keys/ <i>0</i> /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 <code>instanceId</code> , <code>imageId</code> , <code>privateIp</code> , and <code>instanceType</code> have values, and the other returned attributes are null. For more information, see Instance identity documents in the <i>Amazon EC2 User Guide for Linux Instances</i> .

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 \(p. 106\)](#)
- [Snowcone Hangs When Launching a Large Compute Instance \(p. 106\)](#)
- [My Instance Has One Root Volume \(p. 106\)](#)
- [Unprotected Private Key File Error \(p. 107\)](#)

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 RJ45 interface. The Snowcone has two RJ45 interfaces, you may be specifying the wrong physical interface

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 \(p. 83\)](#).
- Update an existing VNIC. For more information, see [Updating a Virtual Network Interface \(p. 91\)](#).

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.

For additional information around adding additional volumes, see <https://docs.aws.amazon.com/snowball/latest/snowcone-guide/snowcone-snowcone-ebs.html>

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.
- Ensure you use a static IP address for the VNI for the DataSync agent. Using DHCP may cause issues because any device reboots can mean that you're using a different IP address, forcing a the need to reconfigure both the DataSync agent and DataSync activation in AWS cloud.
- To use Datasync agent on Snowcone, you need to ensure that the "AllowedHosts" entries includes the DataSync Agent IP address in your NFS configuration on the Snowcone. This may require you to stop and restart the NFS service on Snowcone to enable the DataSync agent.
- Whenever you stop the NFS service, you should also stop the DataSync agent too. Note: If you stop the NFS service which has allow the listed DataSync agent, it will stop Datasync agent from working.
- Check the [maximum number of files per task for a Snowcone HDD device](#) (the # might be higher for Snowcone SSD). If you exceed the 200,000 files limit, the DataSync task will report a memory allocation error and abort execution.

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 \(p. 30\)](#).

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 than 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*.

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 only transfer up to 40M files using a single Snow Family device. If you require to transfer more than 40M files in a single job, please batch the files in order to reduce the file numbers per each transfer. Individual files can be of any size and the maximum file size can be up to size of 5TB. This is not a snow limitation it is a S3 limitation. S3 limits file size to max of 5TB.
- 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](#).
- For NFS based transfers, standard POSIX style meta-data will be added to your objects as they get imported to Amazon S3 from Snowball Edge. In addition, you'll see meta-data "x-amz-meta-user-agent aws-datasync" as we currently use AWS DataSync as part of the internal import mechanism to Amazon S3 for Snowball Edge import with NFS option.

NFS Configuration for Snowball Edge

NFS is ubiquitous (existing or being everywhere at the same time). Every laptop, workstation, and server has an IP connection and every modern operating system has a built-in NFS client. Object storage systems typically use the Amazon S3 API which is less ubiquitous than NFS, especially in the private data center. Amazon S3 does not support NFS. Hence before ordering the SBE, you must decide to use either the S3 interface or NFS interface. If you choose the NFS interface, mount NFS bucket of SBE as a volume.

Prerequisite

- 2 IP addresses:
 - One for physical network interface to activate SBE (e.g. 192.168.0.9)
 - The 2nd for virtual network interface to mount SBE (e.g. 192.168.0.10)
- Workstation

How to Configure Snowball Edge

confirm NFS service is Active

```
snowballEdge describe-service --service-id nfs --profile sbe1
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "ACTIVE"
  }
}
```

If the State is Active, you can mount the SBE volume, otherwise you have to re-start service or create virtual network interface

Create a Virtual Network Interface and Start NFS Service

Check if virtual network interface exists already

```
$ snowballEdge describe-virtual-network-interfaces --profile sbe1
```

Identify physical interface ID

```
$ snowballEdge describe-device --profile sbe1
```

Create virtual network interface and identify v-nic arn

```
$ snowballEdge create-virtual-network-interface \
--physical-network-interface-id s.ni-abcd1234 \
--ip-address-assignment STATIC \
--static-ip-address-configuration IpAddress=192.168.0.10,Netmask=255.255.255.0
--profile sbe1
```

start NFS service

```
$ 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 sbe1
```

check service status

```
$ snowballEdge describe-service --service-id nfs --profile sbe1
{
  "ServiceId" : "nfs",
```

```
"Status" : {  
  "State" : "ACTIVE"  
},  
"Endpoints" : [ {  
  "Protocol" : "nfs",  
  "Port" : 2049,  
  "Host" : "192.168.0.10"  
} ],  
"ServiceConfiguration" : {  
  "AllowedHosts" : [ "192.168.0.6/32", "11.160.2.156/32", "192.168.0.10/32" ]  
}  
}
```

Mount SBE

On the workstation, you can mount the SBE volume which is the bucket name of NFS storage.

```
$ sudo mkdir /mnt/local  
$ sudo mount -t nfs 192.168.0.10:/buckets/your-nfs-bucket-name /mnt/local
```

After mounting SBE volume, you can check the volume size with 'df' command.

```
$ df -h* */mnt/local  
Filesystem      Size  Used Avail Use% Mounted on  
192.168.0.10    80TB  20G   79TB  1% /mnt/local
```

Note

- Check capacity of mounting point **/mnt/local**, shows near "80TB" available capacity for storage optimized Snow Family devices, as an example.
- For more information, see [NFS data transfer for Storage Optimized devices](#).

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" : "192.168.0.10"
  } ],
  "ServiceConfiguration" : {
    "AllowedHosts" : [ "192.168.0.6/32", "11.160.2.156/32", "192.168.0.10/32" ]
  }
}
```

Ports Required to Use AWS Services on an AWS Snowcone device

For AWS services to work properly on an AWS Snowcone device, you must allow the network ports for the service.

The following is a list of network ports that are required for each AWS service.

Port	Protocol	Comment
22	SSH	Device health check and for EC2 SSH

Port	Protocol	Comment
2049	NFS	NFS endpoint
8008	HTTP	EC2 HTTP endpoint
8243	HTTPS	EC2 HTTPS endpoint
9091	HTTP	Endpoint for device management

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. It takes about 20 seconds for the device to shut down.

Disconnect the Snowcone Device

Disconnect the Snowcone cables. 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 \(p. 122\)](#).

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 \(p. 122\)](#).

Protecting Data on Your Device

Consider the following recommendations to help protect the data on your AWS Snowcone device.

Topics

- [Securing Your AWS Snowcone \(p. 114\)](#)
- [Validating NFC Tags \(p. 114\)](#)

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 \(p. 49\)](#).

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 \(p. 52\)](#).
5. Download and install the Snowball Edge client. For more information, see [Using the AWS Snowball Edge Client \(p. 47\)](#).

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 \(p. 49\)](#).

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. Click on your job name within the console.
4. The Job Status pane will be located near the top and reflects the status of the job.

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 \(p. 123\)](#). 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 \(p. 21\)](#).

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 \(p. 118\)](#)
- [JMAPI Required Strings \(p. 118\)](#)
- [JMAPI Endpoints \(p. 118\)](#)
- [JMAPI CLI Commands \(p. 118\)](#)
- [Examples \(p. 119\)](#)

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	<code>aws snowball list-jobs</code>
Describe Job	<code>aws snowball describe-job --job-id <i>[JOB ID]</i></code>
Describe Address	<code>aws snowball describe-address --address-id</code>
Create Address	<code>aws snowball create-address --cli-input-json file://create-address.json</code>
Create Job	<code>aws snowball create-job --cli-input-json file://create-job.json</code>
Cancel Job	<code>aws snowball cancel-job --job-id <i>[JOB ID]</i></code>

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 IMPORT --resources file://path/to/resources.json  
--address-id ADID3be640c8-1111-1111-1111-917f201ffa42 --region us-east-1 --role-  
arn arn:aws:iam::123456789123:role/example_role --kms-key-arn arn:aws:kms:us-  
west-2:000000000000:key/Example --snowball-capacity-preference T14 --device-  
configuration file://path/to/configuration.json --shipping-option SECOND_DAY --snowball-  
type SNC1_SSD
```

The above `create-job` command will create an import job in `us-east-1` region with a `SNC1_SSD` type snowcone device having `T14` capacity preference with a `SECOND_DAY` shipping option.

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 \(p. 120\)](#)
- [Creating Your Job Using the Console \(p. 121\)](#)
- [Creating Your Job Using the AWS CLI \(p. 121\)](#)

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 using a compatible AMI image (See <https://docs.aws.amazon.com/snowball/latest/developer-guide/using-ami.html>).

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 [Creating an Amazon EBS-Backed Linux AMI](#).
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::123456789012:role/snowcone-import-snowcone-role --shipping-option SECOND_DAY --snowball-type SNC1_HDD --snowball-capacity-preference T8 --device-configuration '{"SnowconeDeviceConfiguration":{"WirelessConnection":{"IsWifiEnabled": false}}}'
```

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 \(p. 52\)](#).

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 Snowcone pricing](#).

Topics

- [Preparing an AWS Snowcone Device for Shipping](#) (p. 122)
- [Region-Based Shipping Restrictions](#) (p. 122)
- [Shipping an AWS Snowcone Device](#) (p. 117)

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. [Unlock the device](#).
2. Press the power button on the front of the device, located near the indicator lights, opposite the network ports. 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:

- AWS Snowcone devices are not shipped between international countries—for example, from Asia Pacific (India) to Asia Pacific (Australia).

The sole exception to crossing international countries is among EU Member Countries. For data transfers in the EU Regions, we only ship AWS Snowcone devices to the EU member countries listed:

- Austria, Belgium, Bulgaria, Croatia, Republic of Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Italy, Ireland, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain and Sweden.

Shipments domestically within the same country is permitted. Examples:

- For data transfers in the United Kingdom Region, we ship devices domestically within the UK.
- For data transfers in Asia Pacific (Mumbai), we ship devices within India.

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 [the section called “Shipping Carriers” \(p. 123\)](#). 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 India, Blue Dart is the carrier.
- For Korea, Japan, Australia, and Indonesia, Kuehne + Nagel, is the carrier.
- For China, S.F. Express is the carrier.
- For all other regions, [UPS](#) is the carrier.

AWS Snowcone Pickups in the US, EU, and Canada

In the US, EU, and Canada 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.
- 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.

UPS services for Snow family of products is domestic only within a country.

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 Brazil

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

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

Note

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

AWS Snowcone Pickups in UK

In the United Kingdom, keep the following information in mind for UPS to pick up a 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.
- 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.

UPS services for Snow family of products is domestic only within a country.

Note

Since January 2021, UK is no longer a part of EU. Orders between UK and other EU countries are international orders, a non-general Availability process only approved through a special international process. If a customer has been approved and is returning a device from an EU-country back to LHR or from UK back to an EU-country, they must first request a return to <snowball-shipping@amazon.com> so a Commercial Invoice can be provided prior to arranging pick up/drop off with UPS.

AWS Snowcone Pickups in the Singapore Region

When you are ready to return your device, scan the QR code displayed on the return E Ink label with your mobile phone. This will take you directly onto an email template. Please fill in pick up date/time and contact details.

RETURN

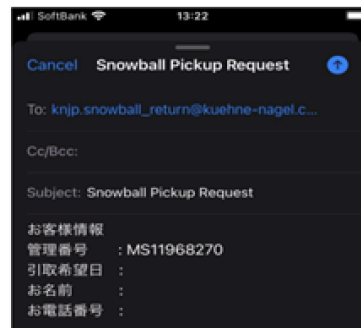
AMS Jobs ID QF6LNZGKTZPF
シリアル番号 2R 207138750022
管理番号 MS14003547



返送のご案内

“以下のQRコードをスキャンし情報を入力の上、
メールにてご連絡をお願いします。”

送信先アドレス knjp.snowball_return@kuehne-nagel.com
件名 Snow Ball Pickup Request
お客様情報
管理番号 : MS14003547
引取希望日 : 要記入
お名前 : 要記入
お電話番号 : 要記入

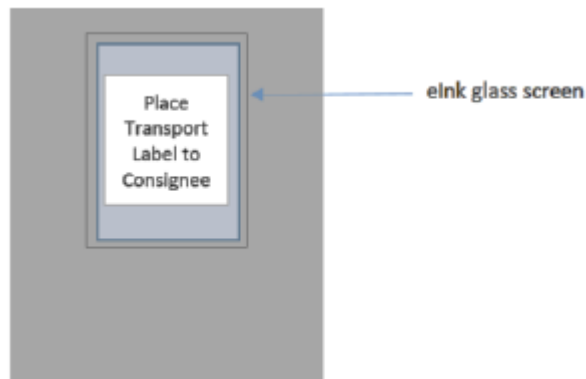


Note

If your pickup address is different from the address where the device was delivered, please add the new address in the email body so the appointed carrier can be informed.

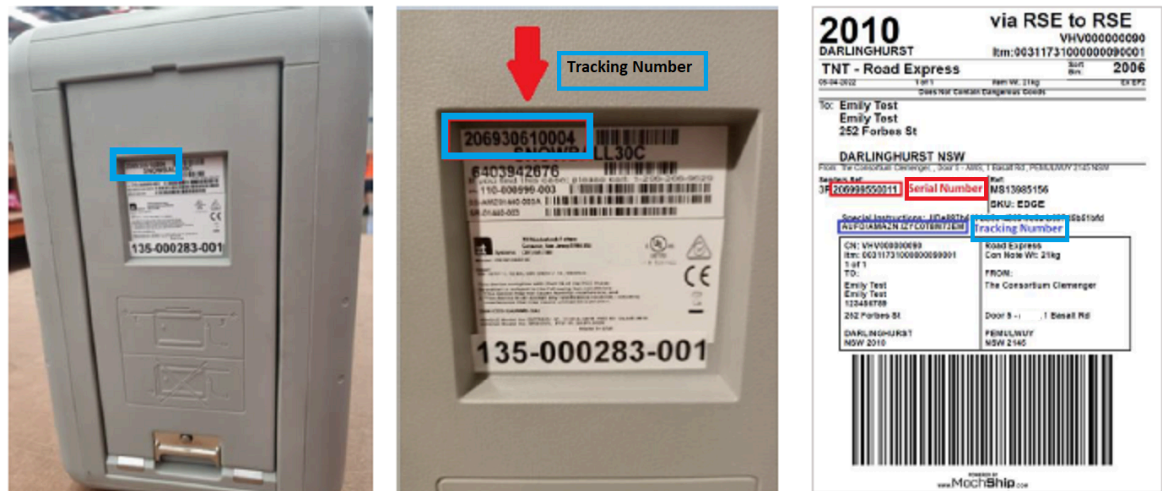
AWS Snowcone Pickups in Australia

In Australia, if you're shipping an AWS Snowcone device back to AWS, place the return transport label (found in the pouch containing these instructions) over the eInk label on the Snow device.



If you have not received a return label with your Snow device, please email knau.snowball_return@kuehne-nagel.com with your device serial number or your reference number.

Example where to locate tracking number and serial number



To arrange return of the Snow device, please email knau.snowball_return@kuehne-nagel.com with the information:

- Your name
- Tracking number (as shown in blue)
- Device serial number (as shown in red)
- Full collection address
- Contact person at pick up
- Contact phone number at pick up
- Collection date
- Collection window (minimum of 3 hour window within business hours)

Note

Collection day and time must be a business day within business hours
Kuehne-nagel team will respond to confirm receipt of pick up request.

AWS Snowcone Pickups in India

In India, Blue Dart picks up the Snowcone device. When you are ready to return your Snowcone device, turn it off and prepare it for return shipping. To schedule pickup, email snowball-pickup@amazon.com with **Snowcone Pickup Request** in the subject line. In the email, include the following information:

- **Job ID** – The job ID associated with the Snowcone 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 Snowcone picked up.
- **Latest Pickup Time** (your local time) – The latest time of day that you want the Snowcone picked up.
- **Special Instructions** (optional) – Any special instructions for picking up the Snowcone, including contact details for coordinating pickup.

The Snowcone team arranges the pickup with Blue Dart and sends you a confirmation email. Blue Dart provides you with a paper shipping label and picks up the Snowcone device.

Important

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

AWS Snowball Pickups in Japan

In Japan, when you are ready to return your device, scan the QR code displayed on the return elnk label with your mobile phone. This will take you directly onto an email template, please fill in pick up date/ time and contact details.

RETURN

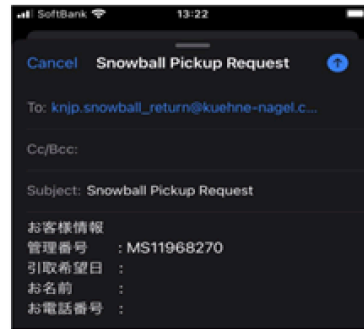
AWS Jobs ID QF6LNZGKTZPF
シリアル番号 2R 207138750022
管理番号 MS14003547



返送のご案内

以下のQRコードをスキャンし情報を入力の上、
メールにてご連絡をお願いします。

送信先アドレス knjp.snowball_return@kuehne-nagel.com
件名 Snow Ball Pickup Request
お客様情報 管理番号 : MS14003547
引取希望日 : 要記入
お名前 : 要記入
お電話番号 : 要記入



AWS Snowball Pickups in Indonesia

Example where to locate tracking number and serial number



In Indonesia, if you're shipping an AWS Snowball Edge device back to AWS to return it, please email knau.snowball_return@kuehne-nagel.com with the information below:

- Your name
- Tracking number (as shown in blue)
- Device serial number (as shown in red)
- Full collection address

- Contact person at pick up
- Contact phone number at pick up
- Collection date
- Collection window (minimum of 3 hour window within business hours)

Note

Collection day and time must be a business day within business hours.
Kuehne-nagel team will respond to confirm receipt of pick up request.

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.
- **United Kingdom (UK)** – When shipping within the UK, you have access to express shipping. Typically, AWS Snow Family devices shipped express are delivered in about a day. In addition, you have access to standard shipping, which typically takes less than a week, one way.
- **Australia** – When shipping to Australia, you have access to the standard shipping speed of 1 – 3 days.
- **Brazil** – When shipping within Brazil, you have access to UPS Domestic Express Saver shipping, which delivers within two business days during commercial hours. Shipping speeds might be affected by interstate border delays.
- **Singapore** – When shipping within Singapore, you have access to Domestic Express Saver shipping.
- **India** – When shipping within India, Snowcone are sent out within 7 working days of AWS receiving all related tax documents.
- **Indonesia** – When shipping to Indonesia, you have access to the standard shipping speed of 1 – 3 days.
- **Korea** – When shipping to Korea, you have access to the standard shipping speed of 1 – 3 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 \(p. 129\)](#)
- [Network \(p. 129\)](#)
- [Resource Management \(p. 130\)](#)
- [Managing EC2 Instances \(p. 130\)](#)

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 \(p. 60\)](#).
- 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

NFS Mount 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 \(p. 131\)](#).

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 \(p. 132\)](#)
- [Limitations for Shipping a Snowcone Device \(p. 133\)](#)
- [Limitations on Processing Your Returned Snowcone Device for Import \(p. 133\)](#)
- [Available AWS Regions \(p. 133\)](#)

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 [Shipping Considerations for AWS Snowcone \(p. 122\)](#).

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*.

Troubleshooting Snowcone Issues

This section provides guidance and insights with the AWS Snowcone device/ service to troubleshoot some of the issues encountered.

Topics

- [Troubleshooting Compute Instances \(p. 134\)](#)
- [Troubleshooting Network Problems \(p. 134\)](#)
- [Troubleshooting Amazon EC2 Instance on Datasync \(p. 135\)](#)
- [Troubleshooting Data Transfer Issues \(p. 135\)](#)

Troubleshooting Compute Instances

Troubleshooting Compute Instances are documented [here](#).

Troubleshooting Network Problems

IP Address is 0.0.0.0

You are plugged in the network and the power cables but the device IP Address shows 0.0.0.0

Action to take

Ensure the cabling between the Snowcone and the network devices is checked properly.

If the device still fails to show an IP address, check the router for problems related to malfunctioning or configuration that may lead to the IP configuration failure.

If the network doesn't have DHCP enabled ensure that you set up a static IP on the device using the STATIC option from the Snowcone display and programming the device with the appropriate static IP address.

Unable to Unlock Device

You are unable to unlock the device using OpsHub or Snowball edge client.

Action to take

Ensure that the client is in the same subnet and in the same network as the AWS Snowcone device.

If the Snowcone has multiple IP Address assignments (for example, WiFi and Ethernet), then ensure that you are using the **same** IP address to connect if multiple clients are trying to respond to Snowcone. Note that you can only use one interface at a time.

If the steps above do not work, please engage AWS Support providing the logs from the device. Use the Snowball Edge client and the command: snowballEdge get-support-logs

Troubleshooting Amazon EC2 Instance on Datasync

Error: Failed to Launch Instance

Snowcone has insufficient capacity to launch the instance for this request.

Action to take

The requirements to run DataSync on an AWS Snowcone device as an agent uses the default instance `snc1.medium`, which provides 2 CPU cores and 4 GiB of memory. Ensure that you have enough resources in the Snowcone device to provision the new Amazon EC2 instance launch request.

Troubleshooting Data Transfer Issues

Access Denied by Server

Error: `mount.nfs - access denied by server while mounting 192.168.1.214`

Action to take

If you configured NFS using quick setup, then by default, only your system will have access to transfer files to this device. If you wish to allow other hosts to upload data to the Snowcone, then disable the NFS service, re-enable the service with the list of IP addresses you want to allow in the NFS configuration.

Connection Times Out During Data Transfer

The connection times out when attempting to transfer data to AWS Snowcone using the Amazon S3 Interface (via AWS CLI).

Reason for this expected behavior

The AWS Snowcone device only supports transfers via the NFS mount for offline transfers and AWS DataSync for online transfers; Amazon S3 Interface is not currently supported. This would explain why you experience timeouts when connecting using the AWS CLI.

Spawn Showmount ENOENT

Error: Uncaught Error - spawn showmount ENOENT. It's an abbreviation of Error NO ENTry (or Error NO ENtity), and can actually be used for more than files/directories.

Action to take

This error can occur if NFS client is not running on the workstation. If the NFS service was not started prior to configuring NFS in AWS OpsHub, you might see this error.

On a Windows client, you can verify if the NFS Service is running by performing the following steps:

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. Reference: [Mounting NFS on a Windows client](#)

This error can also occur if there is a firewall/antivirus between the workstation running AWS OpsHub and the device that could block the connection.

From the workstation, run the telnet to the AWS Snowcone device IP address command on the NFS port number 2049 to check if the connection is established successfully.

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:** November 03, 2021

Change	Description	Date
New AWS Region supported (p. 138)	Snowcone devices are now available in Europe (Paris) AWS Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> . For information on shipping, see Shipping Considerations for AWS Snowcone .	June 29, 2022
New Snowcone troubleshooting tips (p. 138)	AWS Snowcone troubleshooting tips section added similar to Snowball Edge. For new section added, see Troubleshooting Snowcone Issues .	April 15, 2022
New AWS Region supported (p. 138)	AWS Snowcone is now available in the Asia Pacific (Mumbai), and Brazil Regions. Pickup information for EU, Canada, Singapore, and Brazil were added. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	February 23, 2022
New AWS Region supported (p. 138)	AWS Snowcone is now available in the Europe (London) Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	January 5, 2022
Support for Network Time Protocol (NTP) server configuration (p. 138)	Snowcone devices now support external Network Time Protocol (NTP) server configuration.	November 16, 2021
New AWS Region supported (p. 138)	AWS Snowcone SSD is now available in the US West (N. California), US East (Ohio), Asia Pacific (Singapore), Asia Pacific (Tokyo), and Asia Pacific	November 3, 2021

	(Sydney) Regions. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	
New AWS Region supported (p. 138)	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 <i>AWS General Reference</i> .	September 29, 2021
New AWS Region supported (p. 138)	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 <i>AWS General Reference</i> .	August 26, 2021
Support for offline data export from Amazon S3 using Snowcone devices (p. 138)	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. 138)	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. 138)	AWS Snowcone is now available in the Canada (Central) Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	April 28, 2021
New AWS Region supported (p. 138)	AWS Snowcone is now available in the Asia Pacific (Sydney) Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	March 24, 2021

Support for direct network interface (p. 138)	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. 138)	AWS Snowcone is now available in the Europe (Frankfurt) Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	November 18, 2020
New AWS Region supported (p. 138)	AWS Snowcone is now available in the Europe (Ireland) Region. For more information, see AWS Snow Family endpoints and quotas in the <i>AWS General Reference</i> .	September 16, 2020
Introducing AWS Snowcone (p. 138)	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

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS General Reference*.