AWS Snowcone

AWS Snowcone User Guide

AWS Snowcone User Guide: AWS Snowcone

Table of Contents

What Is AWS Snowcone?

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

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

Snowcone is available in two flavors:

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

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

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

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

Snowcone devices offer an 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-compatible 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.

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

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

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

Topics

AWS Snowcone Workflow

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

The workflow

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

- 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.
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.
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, 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 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 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.

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-compatible instances through a local Amazon Elastic Block Store (Amazon EBS) volume presentation. You can use 8 TB or 14 TB for the NFS storage volume and 150 GB for the capacity-optimized HDD (`sbg1`) Amazon EBS storage volume.

For Compute Job Storage

If the job type is local compute, you might create a total of 8 TB or 14 TB of local capacity-optimized HDD (`sbg1`) Amazon EBS volumes and attach them to Amazon EC2-compatible instances. Using Amazon EBS volumes allows the local Amazon EC2-compatible 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 or to run a AWS DataSync agent, 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

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. (2.04 kg) for Snowcone and 4.6 lbs (2.09 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

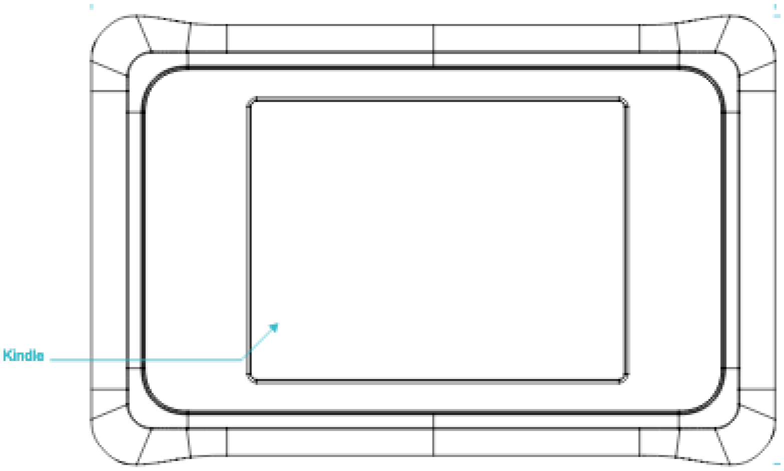
Item	Specification
Transfers through Network File System (NFS)	Yes
Transfers through Amazon S3 API	No
Portability	Battery-based operation
Wireless	Wi-Fi **Note** Wi-Fi is available only in AWS Regions in North America.
Number of usable vCPUs	2 vCPUs
Available memory	4 GB
Network interfaces	2x 1/10 gigabit (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.

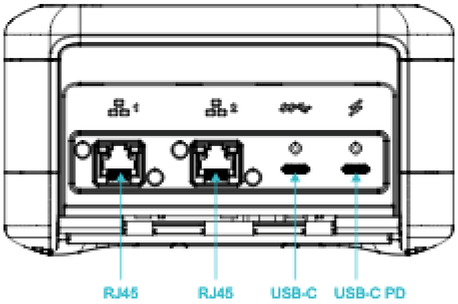
Top View

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.

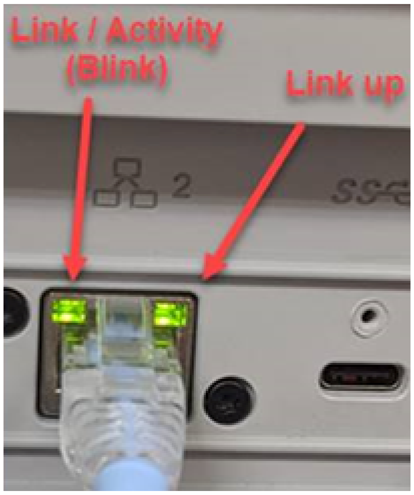
USB

The first USB-C connection is not active.

Ethernet connectors 1 and 2

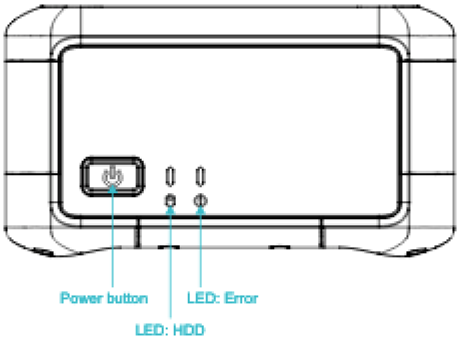
For wired networking, the Snowcone device provides two ports that auto-negotiate for 1 gigabit (Gb) or 10 Gb Ethernet networks.

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 use a Dell Notebook Power Bank Plus – USB C, 65Wh - PW7018LC or any USB-C battery that is rated for a minimum of 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

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 a 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.

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 a job to order a Snow Family device.

Topics

- Questions About the Local Environment
- Working with Files with Special Characters
- Using Amazon EC2 on Snowcone

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?

> 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

Operating System	File Name: `test file.txt`
Linux	/home/<username>/test\ file.txt

Note

The only object metadata that is transferred is the object name and size. 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-compatible 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-compatible 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-compatible 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-compatible instances. For more information, see Using AWS OpsHub for Snow Family to Manage Devices.

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.

Difference between Amazon EC2 and Amazon EC2-compatible instances on Snow Family devices

AWS Snow Family EC2-compatible instances allow customers to use and manage Amazon EC2-compatible instances using a subset of EC2 APIs and a subset of AMIs.

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.

  Before you add AMIs to your job request, make sure that you have one or more supported AMIs in your AWS account. When choosing an AMI from the Marketplace, make sure it has a supported product code and platofrm. For more information, see Checking product codes and platform details of AWS Marketplace AMIs.
- 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.

Checking product codes and platform details of AWS Marketplace AMIs

Before you begin the process to add an AMI from AWS Marketplace to your Snow Family device, ensure the product code and platform details of the AMI are supported in your AWS Region.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances and from which you will create the job to order the Snow Family device. You can select any Region that is available to you, regardless of your location.
3. In the navigation pane, choose **AMIs**.
4. Use the filter and search options to scope the list of displayed AMIs to see only the AMIs that match your criteria. For example, AMIs provided by the AWS Marketplace, choose **Public images**. Then use the search options to further scope the list of displayed AMIs:
    - (New console) Choose the **Search** bar and, from the menu, choose **Owner alias**, then the = operator, and then the value **amazon**.
    - (Old console) Choose the **Search** bar and, from the menu, choose **Owner** and then the value **Amazon images**.

    Note

    AMIs from AWS Marketplace include **aws-marketplace** in the **Source** column.
5. In the **AMI ID** column, choose the AMI ID of the AMI.
6. In the **Image summary** of the AMI, ensure the **Product codes** are supported by your Region. For more information, see the table below.

    Supported AWS Marketplace AMI product codes

AMI operating system	Product code
Ubuntu Server 14.04 LTS	b3dl4415quatdndl4qa6kcu45
CentOS 7 (x86_64)	aw0evgkw8e5c1q413zgy5pjce

AMI operating system	Product code
Ubuntu 16.04 LTS	csv6h7oyg29b7epjzg7qdr7no
Amazon Linux 2	avyfzznywektkgl5qv5f57ska
Ubuntu 20.04 LTS	a8jyynf4hjutohctm41o2z18m
Ubuntu 22.04 LTS	47xbqns9xujfkkjt189a13aqe

7. Then, also ensure the **Platform details** contains one of entries from the list below.

   - Amazon Linux, Ubuntu, or Debian
   - Red Hat Linux bring-your-own-license
   - Amazon RDS for Oracle bring-your-own-license
   - Windows bring-your-own-license

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*.
     - 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.

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.

To get started with AWS Snowcone, see Creating a job to order a Snow Family device.

Topics

Creating a job to order a Snow Family device

To order a Snow Family device, you create a job to order a Snow Family device in the AWS Snow Family Management Console. A *job* is a term that AWS uses to describe the lifecycle of the use of a Snow Family device by a customer. A job begins when you order a device, continues when AWS prepares the device and ships it to you and you use it, and completes after AWS receives and processes the device after you return it. Jobs are categorized by type: export, import, and local compute and storage. For more information, see Understanding AWS Snowball Edge jobs.

After you create the job to order a device, you can use the AWS Snow Family Management Console to view the job status and monitor the progress of the device you ordered as AWS prepares the device to ship to you and after it is returned. For more information, see Job Statuses. After the device is returned and processed by AWS, you can access a job completion report and logs through the AWS Snow Family Management Console. For more information, see Getting your job completion report and logs on the console.

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

Topics

Step 1: Choose a job type

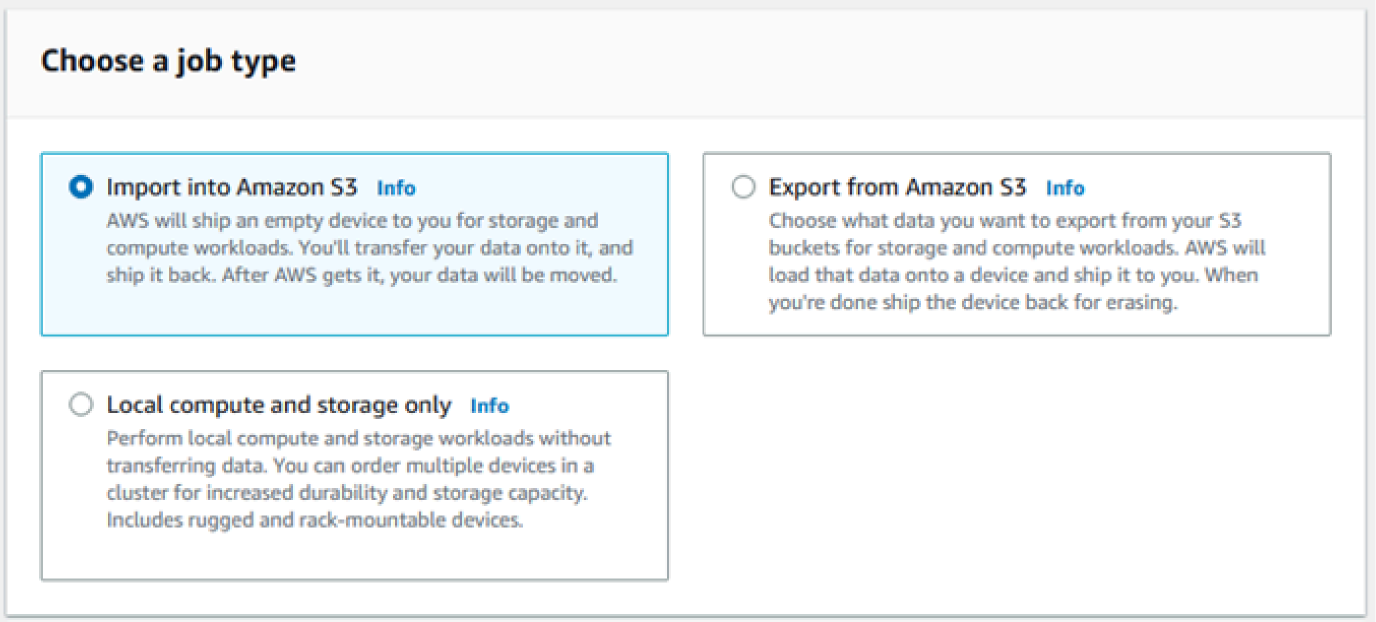
The first step in creating a job is to determine the type of job that you need and to start planning it using the AWS Snow Family Management Console.

To choose your job type

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 existing jobs.
2. If this is your first job, choose **Order an AWS Snow Family device**. If you're 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** in the left navigation bar. Choose **Next step** to open the **Plan your job** page.
3. In the **Job name** section, provide a name for your job in the **Job name** box.
4. Depending on your need, choose one of the following job types:

   - **Import into Amazon S3** – 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 to the device using NFS share, ship it back to AWS, and your data is uploaded to Amazon S3.

- **Local compute and storage only** – Perform compute and storage workloads on the device without transferring data.



Choose a job type

● Import into Amazon S3 Info
AWS will ship an empty device to you for storage and compute workloads. You'll transfer your data onto it, and ship it back. After AWS gets it, your data will be moved.

○ Export from Amazon S3 Info
Choose what data you want to export from your S3 buckets for storage and compute workloads. AWS will load that data onto a device and ship it to you. When you're done ship the device back for erasing.

○ Local compute and storage only Info
Perform local compute and storage workloads without transferring data. You can order multiple devices in a cluster for increased durability and storage capacity. Includes rugged and rack-mountable devices.

5. Choose **Next** to continue.

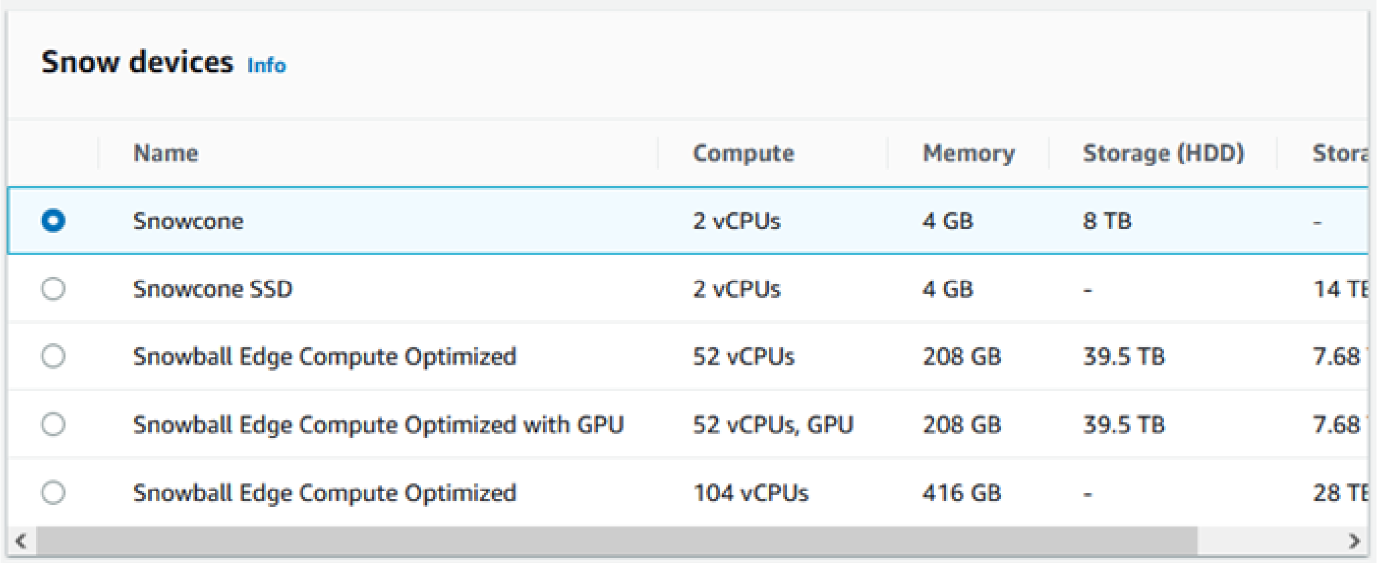
Step 2: Choose your compute and storage options

Choose the hardware specifications for your Snow Family device, which of your Amazon EC2-compatible instances to include on it, how data will be stored, and pricing.

To choose your device's compute and storage options

1. In the **Snow devices** section, choose the Snow Family device to order.

> **Note**
>
> Some Snow Family devices might not be available depending on the AWS Region you are ordering from and the job type you chose.



Snow devices Info

	Name	Compute	Memory	Storage (HDD)	Stora
●	Snowcone	2 vCPUs	4 GB	8 TB	-
○	Snowcone SSD	2 vCPUs	4 GB	-	14 T
○	Snowball Edge Compute Optimized	52 vCPUs	208 GB	39.5 TB	7.68
○	Snowball Edge Compute Optimized with GPU	52 vCPUs, GPU	208 GB	39.5 TB	7.68
○	Snowball Edge Compute Optimized	104 vCPUs	416 GB	-	28 T

2. 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.
3. In the **Choose your pricing option** section, from the **Choose your pricing option** menu, choose the type of pricing to apply to this job. For device pricing, see AWS Snowcone Pricing.
4. In the **Select the storage type** section, make a choice according to your need:

   - **NFS based data transfer**: Use Network File System (NFS) based data transfer to drag and drop files from your computer into Amazon S3 buckets on Snow Family devices.

> ⚠ **Warning**
>
> NFS based data transfer doesn't support the S3 adapter. If you proceed with NFS based data transfer, you must mount the NFS share to transfer objects. Using the AWS CLI to transfer objects will fail.
> See Using NFS for Offline Data Transfer in the AWS Snowcone User Guide for more information.

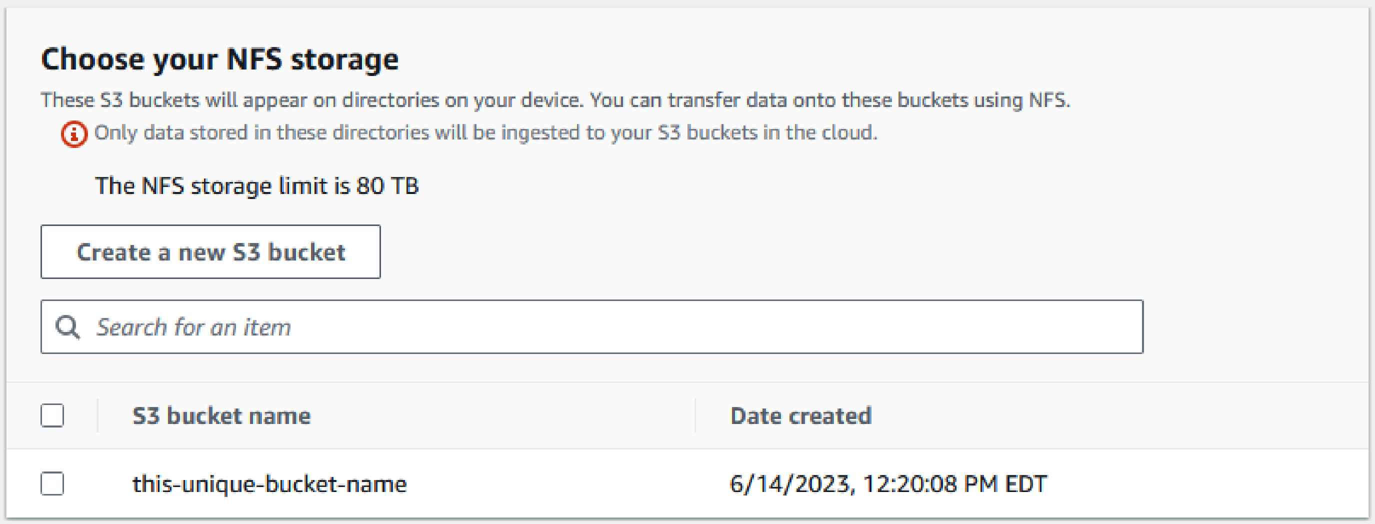
5. If you selected *NFS based data transfer* as the storage type, in the **Select your S3 buckets** section, do one or more of the following to select one or more S3 buckets:

   a. Choose the S3 bucket that you want to use in the **S3 bucket name** list.

b. In the **Search for an item** field, enter all or part a bucket name to filter the list of available buckets on your entry, then choose the bucket.

c. Choose the **Create a new S3 bucket** to create a new S3 bucket. The new bucket name appears in the **Bucket name** list. Choose it.

You can include one or more S3 buckets. These buckets appear on your device as local S3 buckets.



Choose your NFS storage

These S3 buckets will appear on directories on your device. You can transfer data onto these buckets using NFS.

ⓘ Only data stored in these directories will be ingested to your S3 buckets in the cloud.

The NFS storage limit is 80 TB

□	S3 bucket name	Date created
□	this-unique-bucket-name	6/14/2023, 12:20:08 PM EDT

6. In the **Compute using EC2-compatible instances - *optional*** section, choose Amazon EC2-compatible AMIs from your account to include on the device. Or, in the search field, enter all or part the name of an AMI to filter the list of available AMIs on your entry, then choose the AMI.

   For more information, see Creating a Job with Compute Instances in this guide.

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

7. Choose the **Next** button.

Step 3: Choose your features and options

Choose the features and options to include in your AWS Snow Family device job, including Amazon EKS Anywhere for Snow, an AWS IoT Greengrass instance, and remote device management capability.

To choose your features and options

1. To enable your wireless networking on your Snowcone device, select **Enable Wireless on Snowcone**.
2. To enable remote management of your Snow Family device by AWS OpsHub or Snowball Edge Client, select **Manage your Snow device remotely with AWS OpsHub or Snowball client**.
3. Select the **Next** button.

Step 4: Choose security, shipping, and notification preferences

Topics

Choose 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 **Choose service access type** section, do one of the following:
   - Choose **Snow console will create and use a service-linked role to access AWS resources on your behalf.** 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 service role to use**, to specify the **Role ARN** that you want, or you can use the default role.

3. Choose **Next**.

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. 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 Shipping Carriers.**

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

- 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.

  > ⓘ **Note**
  >
  > Jobs to order Snow devices created in the US West (N. California) and US West (Oregon) regions are routed through US East (N. Virginia) region. Because of this, service calls like Amazon SNS are also directed through US East (N. Virginia). We recommend creating any new SNS topics in the US East (N. Virginia) region for the best experience.

Notifications 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

For more information about job status change notifications and encrypted SNS topics, see Notifications for Snow Family devices in this guide.

Select the **Next**.

Step 5: Review job summary and create your job

After you provide all the necessary information for your AWS Snow Family devices job, review the job and create it. After you create the job, AWS will begin preparing the Snow Family device for shipment to you.

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.

1. In the **Job summary** 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**.

> **Note**
>
> After you create a job to order a Snow Family device, you can cancel it while it is in the *Job created* state without incurring any charges. For more information, see Cancelling a job through the AWS Snow Family Management Console.

> **Note**
>
> Snowcone devices are not provided with power cords, and one must be provided separately. For more informaiton, see AWS Snowcone Power Supply and Accessories.

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.

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.

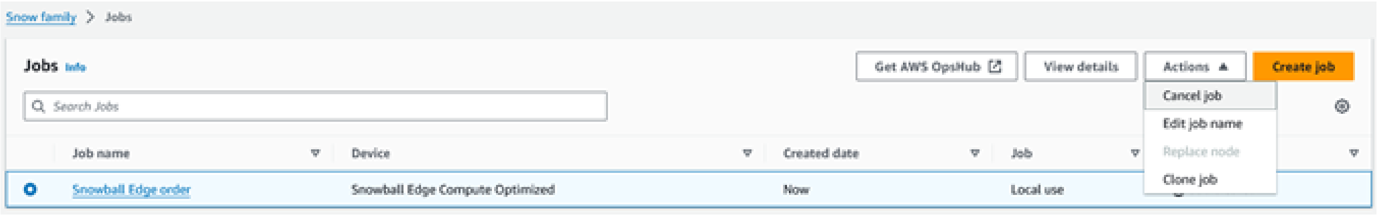
To download and install AWS OpsHub for Snow Family

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**.

Cancelling a job through the AWS Snow Family Management Console

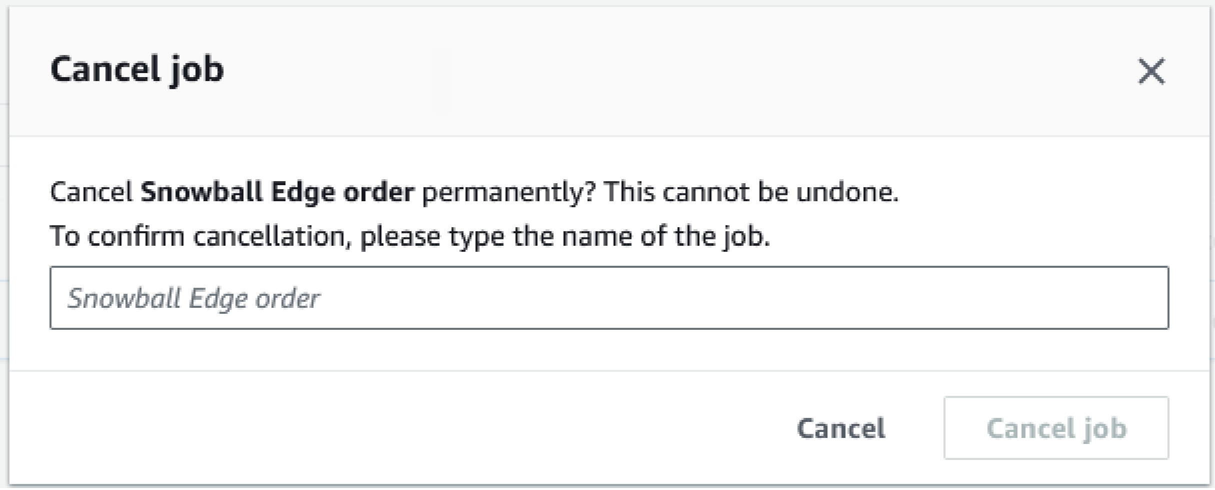
After creating a job to order a Snow Family device, you can cancel the job through the AWS Snow Family Management Console. If you cancel the job, you won't receive the device you ordered. You can only cancel the job while the job status is *Job created*. After the job progresses past this status, you cannot cancel the job. For more information, see Job Statuses.

1. Log in to the AWS Snow Family Management Console.
2. Choose the job to cancel.
3. Choose **Actions**. From the menu that appears, choose **Cancel job**.



Job name	Device	Created date	Job
Snowball Edge order	Snowball Edge Compute Optimized	Now	Local use

4. The **Cancel job** window appears. To confirm cancelling the job, enter the **`job name`** and choose **Cancel job**. In the list of jobs, **Cancelled** appears in the **Status** column.



Cancel job

Cancel **Snowball Edge order** permanently? This cannot be undone.

To confirm cancellation, please type the name of the job.

Getting credentials to access a Snow Family device

Each job has a set of credentials that you must get from the AWS Snow Family Management Console or the job management API to authenticate your access to the Snow Family device. These credentials are an encrypted manifest file and an associated unlock code. The manifest file contains important information about the job and permissions associated with it.

> **Note**
>
> You get your credentials after the device is in transit to you. You can see the status of your job in the AWS Snow Family Management Console. For more information, see Understanding AWS Snowcone Job Statuses.

To get your credentials using the console

1. Sign in to the AWS Management Console and open the AWS Snow Family Management Console.
2. On the console, search the table for the specific job to download the job manifest for, and then choose that job.
3. Expand that **Job status** pane, and choose **View job details**.
4. In the details pane that appears, expand **Credentials** and then do the following:

   - Make a note of the unlock code (including the hyphens), because you need to provide all 29 characters to unlock the device.

- In the dialog box, choose **Download manifest**, and follow the instructions to download the job manifest file to your computer. The name of your manifest file includes your **Job ID**.

> **Note**
>
> We recommend that you don't save a copy of the unlock code in the same location in the computer as the manifest for that job. For more information, see Best Practices for the AWS Snowcone Device.

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

Next: Using the AWS Snowball Edge Client

Unlocking the Snow Family device

This section describes unlocking the Snow Family device using the Snowball Edge CLI. To unlock the device using AWS OpsHub, a graphical user interface (GUI) tool for Snow Family devices, see Unlocking a device.

Before using a Snow Family device device to transfer data or perform edge compute tasks, you need to unlock the device. When unlocking the device, you authenticate your ability to access it by providing two forms of credentials: a 29-digit unlock code and a manifest file. After you unlock the device, you can further configure the device, move data to or from it, set up and use Amazon EC2-compatible instances, and more.

Before unlocking a device, the device must be plugged in to power and network, turned on, and an IP address assigned. See AWS Snowcone Device Specifications. You will need the following information about the Snow Family device:

- Download and install the Snowball Edge client. For more information, see Using the AWS Snowball Edge Client.
- Get the credentials from the AWS Snow Family Management Console. For one or more standalone devices, the unlock codes and manifest file for each Snow Family device. For more information on downloading credentials, see Getting credentials to access a Snow Family device.

- Power on each device and connect it to your network. For more information, see AWS Snowcone Device Specifications.

To unlock a standalone device with the Snowball Edge client

1. Find the IP address for the Snowcone on the LCD display of the device. Make a note of that IP address.
2. Use the `unlock-device` command to authenticate your access to the Snow Family device with the IP address of the Snow Family device and your credentials, as follows.

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

   The device indicates it was unlocked successfully with the following message.

   ```
   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.
   ```

   If the command returns `connection refused`, see Troubleshooting unlocking a Snow Family device.

   Example of `unlock-device` command

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

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

Troubleshooting unlocking a Snow Family device

If the `unlock-device` command returns `connection refused`, you may have mistyped the command syntax or the configuration of your computer or network may be preventing the command from reaching the Snow device. Take these actions to resolve the situation:

1. Ensure the command was entered correctly.
    a. Use the LCD screen on the device to verify the IP addressed used in the command is correct.
    b. Ensure that the path to the manifest file used in the command is correct, including the file name.
    c. Use the AWS Snow Family Management Console to verify the unlock code used in the command is correct.
2. Ensure the computer you are using is on the same network and subnet as the Snow device.
3. Ensure the computer you are using and the network are configured to allow access to the Snow device. Use the `ping` command for your operating system to determine if the computer can reach the Snow device over the network. Check the configurations of antivirus software, firewall configuration, virtual private network (VPN), or other configurations of your computer and network.

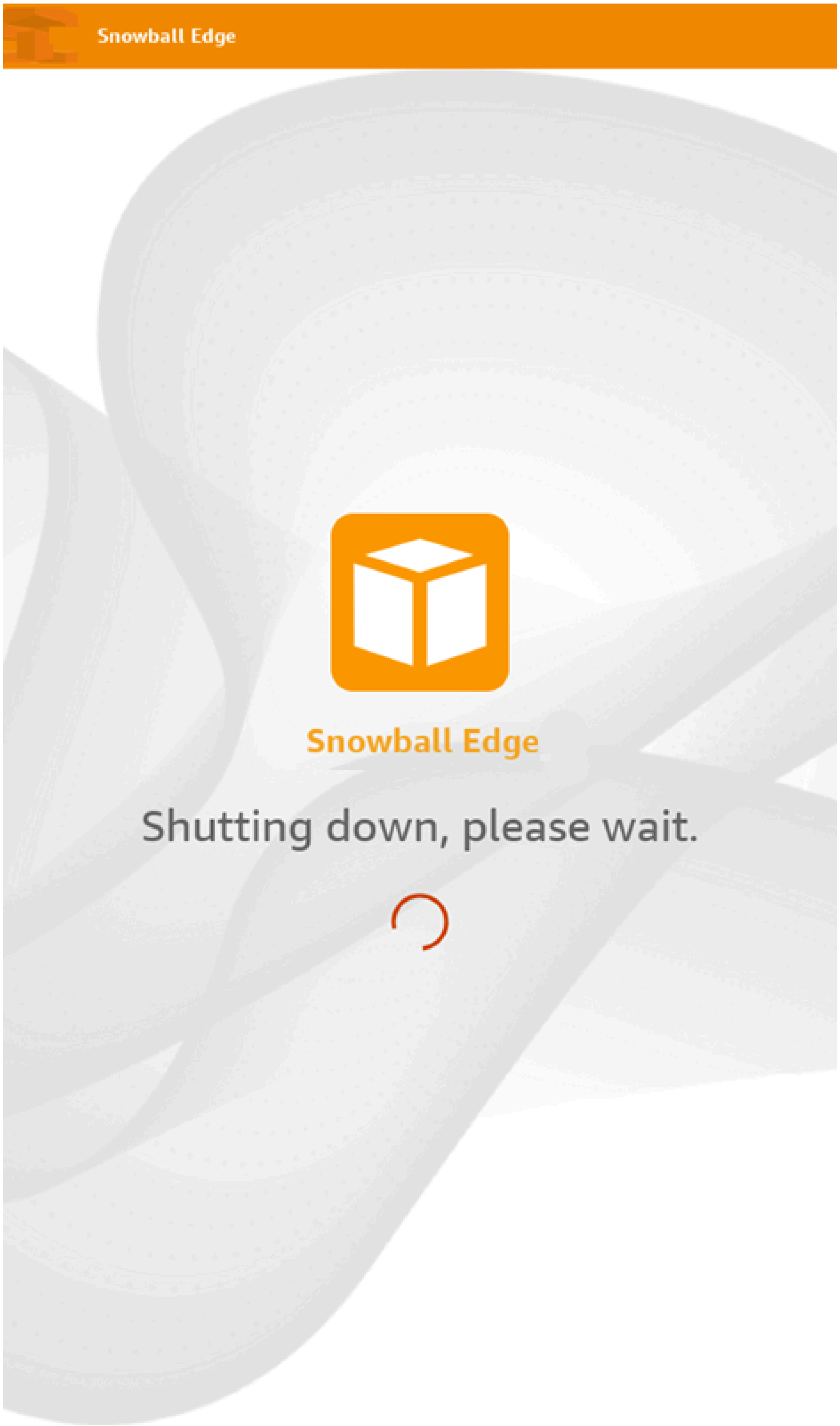
Now you can begin using the Snow Family device.

Rebooting the Snow Family device

Before you reboot a Snow Family device, make sure that all data transfer to the device has stopped. If you were using the NFS interface to transfer data, disable it before you power off the device. For more information, see Stopping the NFS interface.

To reboot the device using the power button:

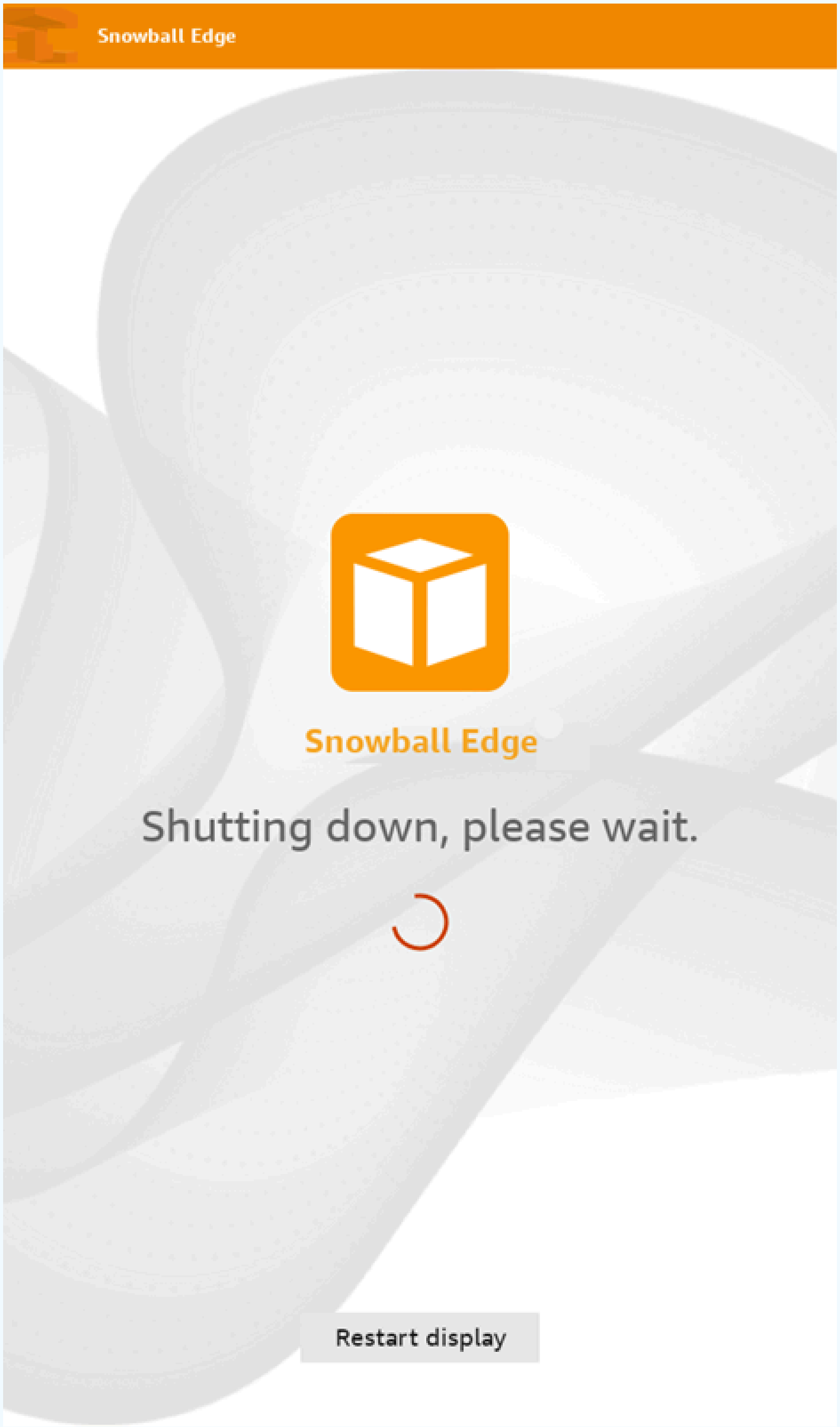
1. When all communication with the device has ended, turn it off by pressing the power button located on the front of the device. It takes about 20 seconds for the device to shut down. While the device is shutting down, the LCD screen displays a message indicating the device is shutting down.



Shutting down, please wait.

Note

If the LCD screen is displaying the shutdown message when the device is not actually being shut down, press the **Restart display** button on the screen to return the screen to normal operation.



Snowball Edge

Snowball Edge

Shutting down, please wait.

2. Press the power button. When the device is ready, the LCD display shows a short video while the device is getting ready to start. After about 10 minutes, the device is ready to be unlocked.

3. Unlock the device. See Unlocking an AWS Snowcone Device.

To reboot the device using the Snowball Edge client:

1. When all communication with the device has ended, use the `reboot-device` command to reboot it. When the device is ready, the LCD display shows a short video while the device is getting ready to start. After about 10 minutes, the device is ready to be unlocked.

   ```
   snowballEdge reboot-device
   ```

2. Unlock the device. See Unlocking an AWS Snowcone Device.

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-compatible 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.

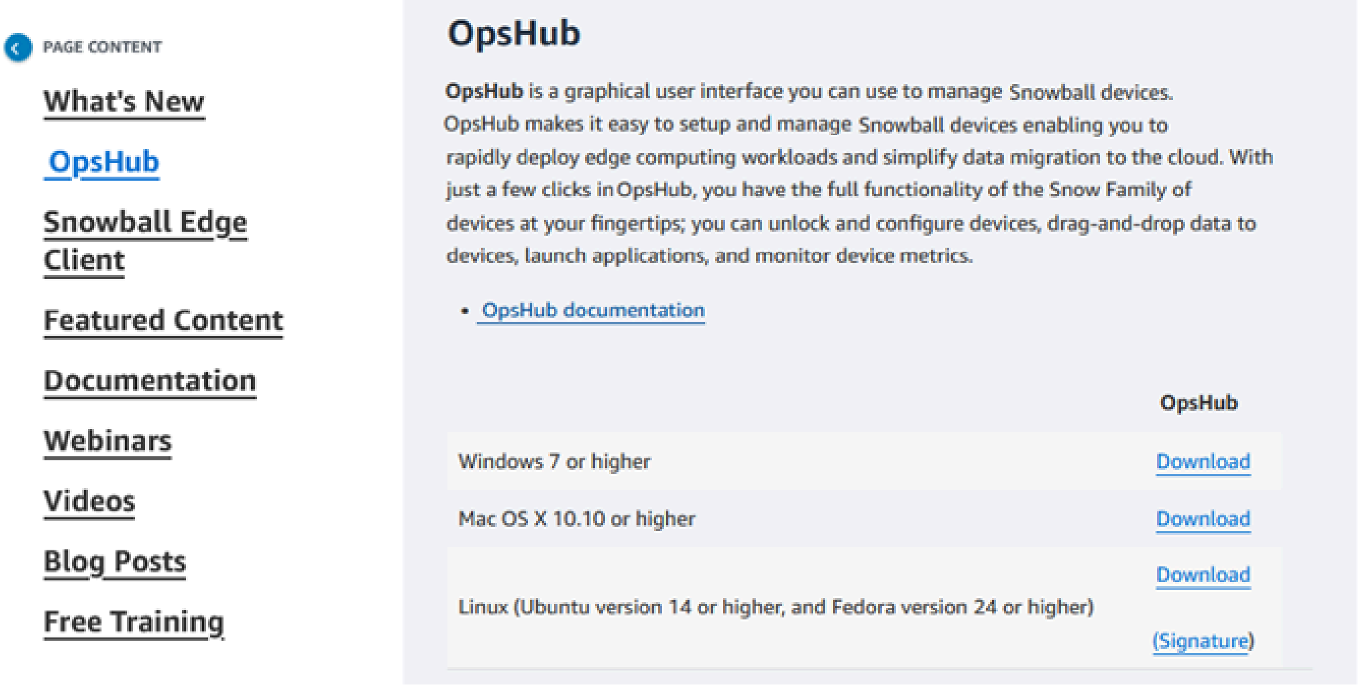
Topics

- Verifying the PGP signature of AWS OpsHub (optional)
- Managing AWS services on your device
- Using DataSync to transfer files to AWS
- Managing Your Devices
- Setting the NTP time servers for your device

Download AWS OpsHub for Snow Family devices

To download AWS OpsHub

1. Navigate to the AWS Snowball resources website.



OpsHub

OpsHub is a graphical user interface you can use to manage Snowball devices. OpsHub makes it easy to setup and manage Snowball devices enabling you to rapidly deploy edge computing workloads and simplify data migration to the cloud. With just a few clicks in OpsHub, you have the full functionality of the Snow Family of devices at your fingertips; you can unlock and configure devices, drag-and-drop data to devices, launch applications, and monitor device metrics.

- OpsHub documentation

	OpsHub
Windows 7 or higher	Download
Mac OS X 10.10 or higher	Download
Linux (Ubuntu version 14 or higher, and Fedora version 24 or higher)	Download (Signature)

2. In the **AWS OpsHub** section, choose **Download** for your operating system, and follow the installation steps.

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 Activating Snow Device Management.

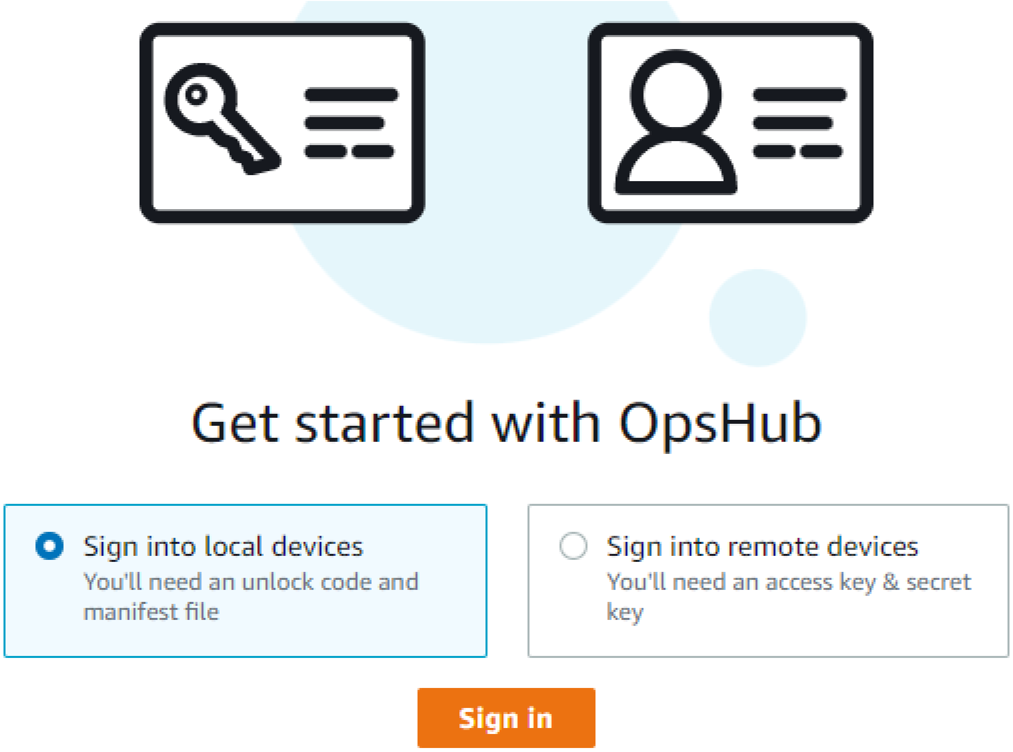
Topics

- Unlocking a device locally
- Unlocking a device remotely

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 a network cable (typically an Ethernet 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**.



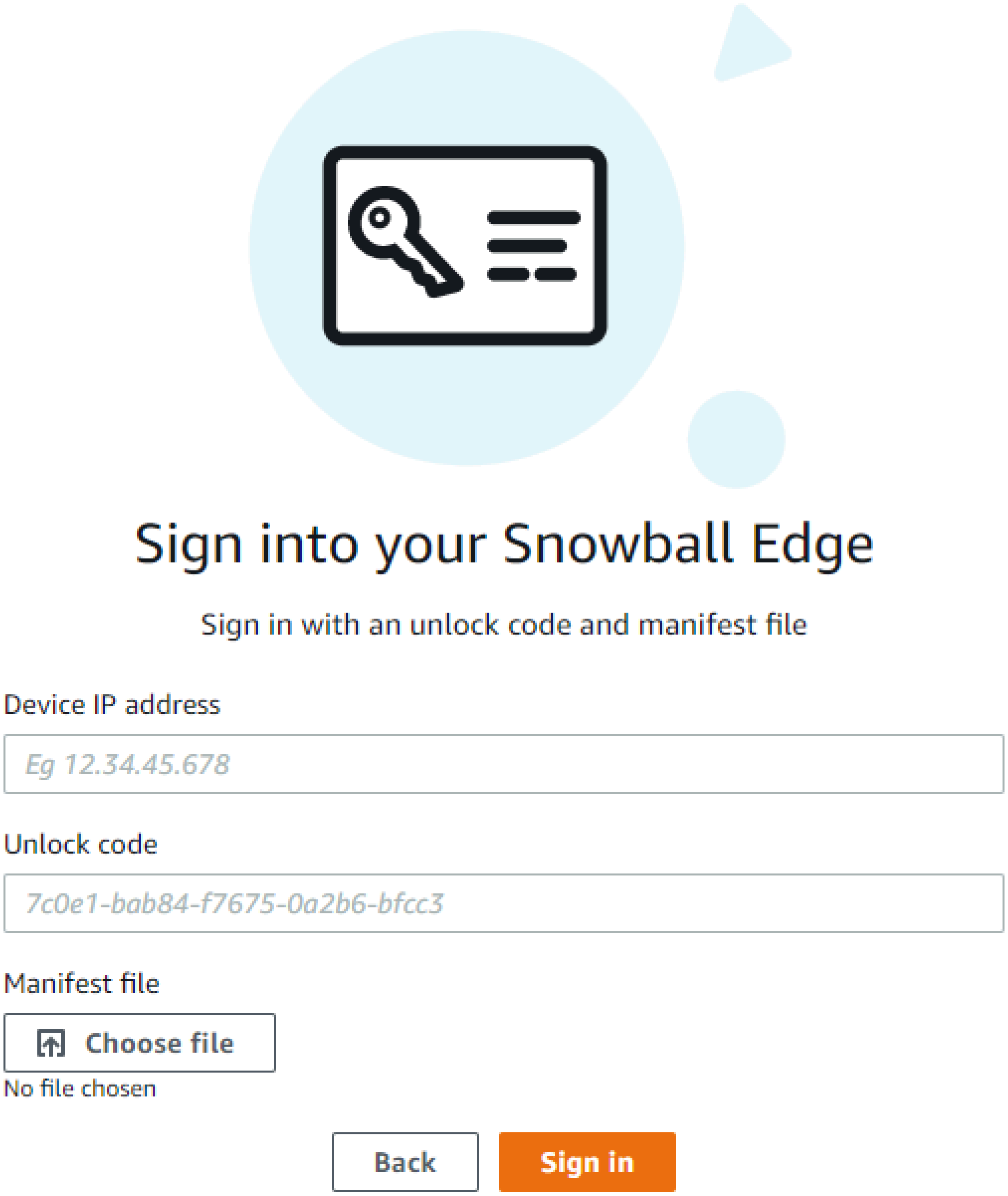
Get started with OpsHub

● Sign into local devices
You'll need an unlock code and manifest file

○ Sign into remote devices
You'll need an access key & secret key

5. On the **Sign in to local devices** page, choose your Snow Family devices type, and then choose **Sign in**.

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**.



Sign into your Snowball Edge

Sign in with an unlock code and manifest file

Device IP address

Eg 12.34.45.678

Unlock code

7c0e1-bab84-f7675-0a2b6-bfcc3

Manifest file

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.

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.

> **Note**
>
> Available storage space on the Snowcone device is not accurate until the NFS service is started. See Managing the NFS interface.

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.

Unlocking a device remotely

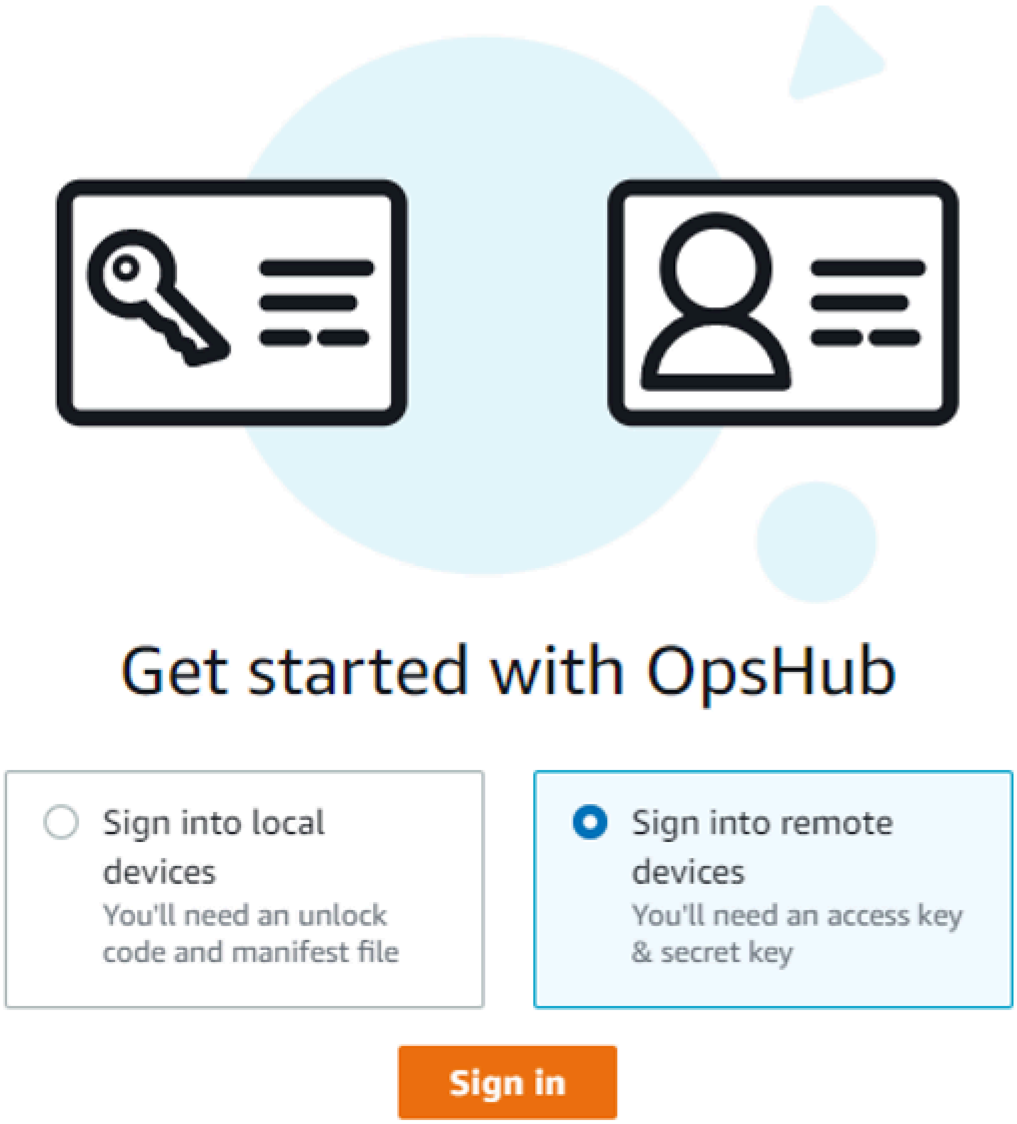
To unlock a Snow Family device not

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 into remote devices**, and then choose **Sign in**.

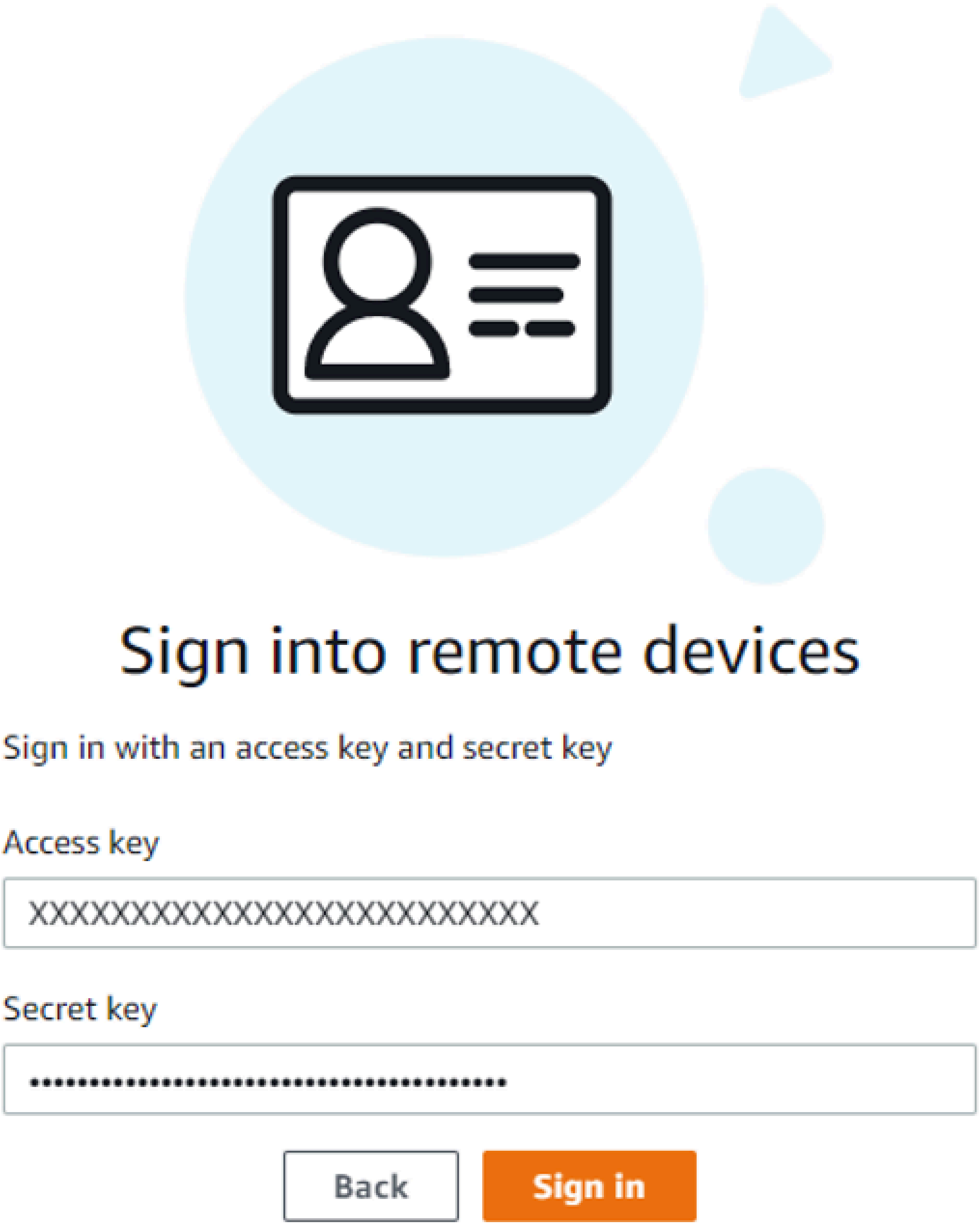


Get started with OpsHub

○ Sign into local devices
You'll need an unlock code and manifest file

● Sign into remote devices
You'll need an access key & secret key

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**.



Sign into remote devices

Sign in with an access key and secret key

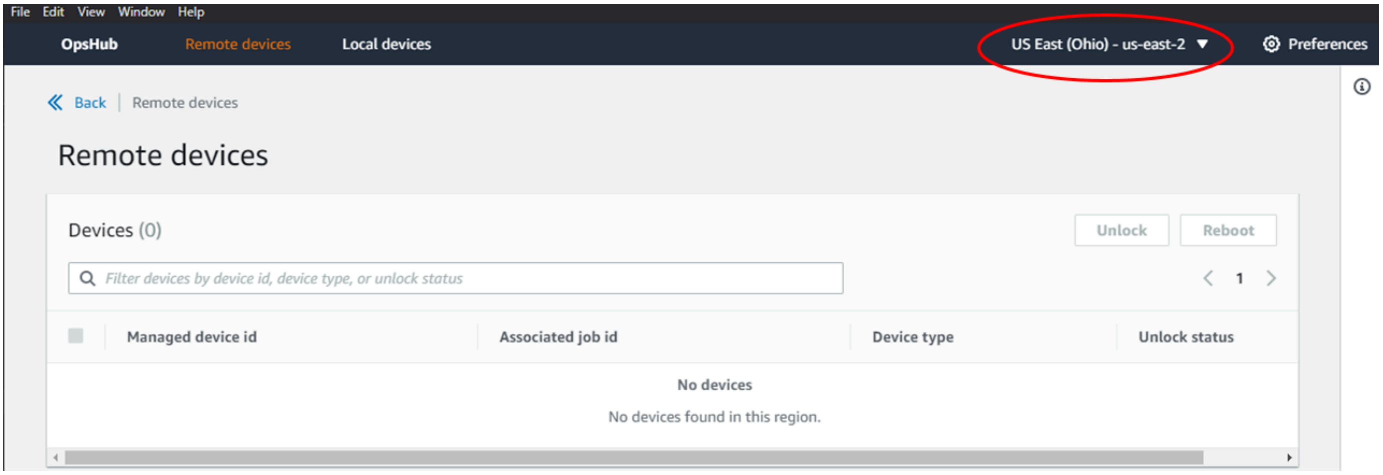
Access key

XXXXXXXXXXXXXXXXXXXXXXXXX

Secret key

••

6. At the top of the **Remote devices** tab, choose the region of the Snow device to unlock remotely.


File Edit View Window Help
OpsHub
Remote devices
Local devices
US East (Ohio) - us-east-2 ▼
Preferences
Back
Remote devices
Unlock
Reboot
Filter devices by device id, device type, or unlock status
1

Remote devices

Devices (0)

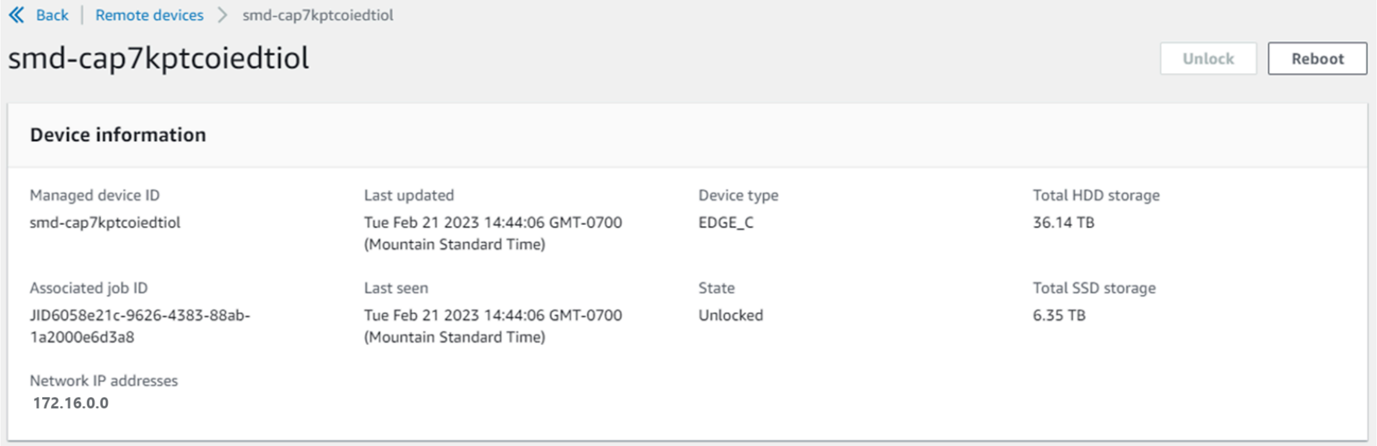
□	Managed device id	Associated job id	Device type	Unlock status

No devices

No devices found in this region.

7. 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.

> **Note**
>
> Available storage space on the Snowcone device is not accurate until the NFS service is started. See Managing the NFS interface.



smd-cap7kptcoiedtiol

Device information

Managed device ID
smd-cap7kptcoiedtiol

Last updated
Tue Feb 21 2023 14:44:06 GMT-0700 (Mountain Standard Time)

Device type
EDGE_C

Total HDD storage
36.14 TB

Associated job ID
JID6058e21c-9626-4383-88ab-1a2000e6d3a8

Last seen
Tue Feb 21 2023 14:44:06 GMT-0700 (Mountain Standard Time)

State
Unlocked

Total SSD storage
6.35 TB

Network IP addresses
172.16.0.0

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.

Verifying the PGP signature of AWS OpsHub (optional)

The AWS OpsHub application installer package for the Linux operating system 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 operating system installer from AWS Snowcone Resources or Snowball Edge Resources.

To verify the AWS OpsHub install package on for the Linux operating system

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

   ```
   -----BEGIN PGP PUBLIC KEY BLOCK-----
   xsFNBF/hGf8BEAC9HCDV8uljDX02Jxspi6kmPu4xqf4ZZLQsSqJcHU61oL/c
   /zAN+mUqJT9aJ1rr0QFGVD1bMogecUPflTWlDkEEpG8ZbX5P8vR+EElO/rW/
   WtqizSudy6qy59ZRK+YVSDx7DZyuJmIO7j00UADCL+95ZQN9vqwHNjBHsgfQ
   l/1Tqhy81ozTZXcI/+u+99YLaugJIP6ZYIeDfpxnghqyVtaappBFTAyfG67Y
   N/5mea1VqJzd8liFpIFQnl+X7U2x6emDbM01yJWV3aMmPwhtQ7iBdt5a4x82
   EF5bZJ8HSRMvANDILD/9VTN8VfUQGKFjFY2GdX9ERwvfTb47bbv9Z28Vl284
   4lw2w1Bl007FoO2v/Y0ukrN3VHCpmJQS1IiqZbYRa0DVK6UR5QNvUlj5fwWs
   4qW9UDPhT/HDuaMrMFCejEn/7wvRUrGVtzCT9F56Al/dwRSxBejQQEb1AC8j
   uuyi7gJaPdyNntROEFTD7iO2L6X2jB4YLfvGxP7Xeq1Y37t8NKF8CYTpOry/
   Wvw0iKZFbo4AkiI0aLyBCk9HBXhUKa9x06gOnhh1UFQrPGrk60RPQKqL76HA
   E2ewzGDa90wlRBUAt2nRQpyNYjoASBvz/cAr3e0nuWsIzopZIenrxI5ffcjY
   f6UWA/OK3ITHtYHewVhseDyEqTQ4MUIWQS4NAwARAQABzTlBV1MgT3BzSHVi
   IGZvciBTbm93IEZhbWlseSA8YXdzLW9wc2h1Yi1zaWduZXJAYW1hem9uLmNv
   bT7CwY0EEAEIACAFAl/hGf8GCwkHCAMCBBUICgIEFgIBAAIZAQIbAwIeAQAh
   CRAhgc9adPNF8RYhBDcvpelIaY930bOvqiGBz1p080XxGbcP+gPZX7LzKc1Y
   w9CT3UHgkAIawOSXYktujzoYVxAz8/j3jEkCY0dKnfyqvWZDiJAXnzmxWWbg
   cxg1g0GXNXCM4lAd68CmbAOLoLTaWSQX30ZbswzhbtX2ADAlopV8RLBik7fm
   bS9FyuubDRhfYRQq0fpjUGXFiEgwg6aMFxsrGLlv4QD7t+6ftFIe/mxLbjR4
   iMgtr8FIPXbgn05YYY/LeF4NIgX4iLEqRbAnfWjPzqQ1spFWAotIzDmZqby+
   WdWThrH4K1rwtYM8sDhqRnMnqJrGFZzk7aDhVPwF+FOVMmPeEN5JRazEeUrl
   VZaSw6mu0n4FMGSXuwGgdvmkqnMe6I5/xLdU4IOPNhp0UmakDWOq/a1dREDE
   ZLMQDMINphmeQno4inGmwbRo63gitD4ZNR5sWwfuwty25lo8Ekv7jkkp3mSv
   pdxn5tptttnPaSPcSIX/4EDl19Tu0i7aup+v30t7eikYDSZG6g9+jHB3Va9e
   /VWShFSgy8Jm2+qq/ujUQDAGTCfSuY9jg1ITsog6ayEZa/2upDJ1m+4OHK4p
   8DrEzP/3jTahT8q5ofFWSRDL17d3lTSU+JBmPE3mz311FNXgiO8w+taY320z
   ```

```
+irHtb3iSiiukbjS8s0maVgzszRqS9mhaEn4LL0zoqrUicmXgTyFB7n2LuYv
O7vxMO5xxhGQwsF2BBABCAAJBQJf4RoCAhsDACEJEBFZvzT/tDi5FiEEi+O9
V+UAYN9Gnw36EVm/NP+0OLnnEQ/+J4C0Mn8j0AebXrwBiFs83sQo2q+WHL1S
MRc1g5gRFDXs6h1Gv+TGXRen7j1oeaddWvgOtUBxqmCOjr+8AKH0OtiBWSuO
lsS8JU5rindEsKUrKTwcG2wyZFoe1zlE8xPkLRSRN5ZbbgKsTz16l1HgCCId
Do+WJdDkWGWxmtDvzjM32EI/PVBd108ga9aPwXdhLwOdKAjZ4JrJXLUQJjRI
IVDSyMObEHOUM6a/+mWNZazNfo0LsGWqGVa6Xn5WJWlwR1S78vPNfO3BQYuO
YRjaVQR+kPtB9aSAZNi5sWfk6NrRNd1Q78d067uhhejsjRt7Mja2fEL4Kb1X
nK4U/ps7XlO3o/VjblneZOhJK6kAKU172tnPJTJ31JbOxX73wsMWDYZRZVcK
9X9+GFrpwhKHWKKPjpMOt/FRxNepvqRl72TkgBPqGH2TMOFdB1f/uQprvqge
PBbS0JrmBIH9/anIqgtMdtcNQB/0erLdCDqI5afOuD1OLcLwdJwG9/bSrfwT
TVEE3WbXmJ8pZgMzlHUiZE6V2DSadV/YItk50IOjjrOVHOHvlFMwGCEAIFzf
9P/pNi8hpEmlRphRiOVVcdQ30bH0M0gPHu5V9flIhyCL1zU3LjYTHkq0yJD5
YDA1xO1MYq3DcSM513OVBbLmuVS2GpcsTCYqlgQA6h/zzMwz+/7OwU0EX+EZ
/wEQAOAY8ULmcJIQWIr14V0jylpJeD3qwj7wd+QsBzJ+mOpOB/3ZFAhQiNOl
9yCDlHeiZeAmWYX9OIXrNiIdcHy+WTAp4G+NaMpqE52qhbDjz+IbvLpl1yDH
bYEHPjnTHXEy2lbvKAJOKkw/2RcQOi4dodGnq5icyYj+9gcuHvnVwbrQ96Ia
0D7c+b5T+bzFqk90nIcztrMRuhDLJnJpi7OjpvQwfq/TkkZA+mzupxfSkq/Y
N9qXNEToT/VI2gn/LS0X4Ar1l2KxBjzNEsQkwGSiWSYtMA5J+Tj5ED0uZ/qe
omNblAlD4bm7Na8NAoLxCtAiDq/f3To9Xb18lHsndOmfLCb/BVgP4edQKTIi
C/OZHy9QJlfmN0aq7JVLQAuvQNEL88RKW6YZBqkPd3P6zdc7sWDLTMXMOd3I
e6NUvU7pW0E9NyRfUF+oT4s9wAJhAodinAi8Zi9rEfhK1VCJ76j7bcQqYZe0
jXD3IJ7T+X2XA8M/BmypwMW0SoljzhwhO44RAasr/fAzpKNPB318JwcQunIz
u2N3CeJ+zrsomjcPxzehwsSVq1lzaL2ureJBLOKkBgYxUJYXpbS01ax1TsFG
09ldANOs9Ej8CND37GsNnuygjOgWXbX6MNgbvPs3H3zi/AbMunQ1VBlw07JX
zdM1hBQZh6w+NeiEsK1T6wHi7IhxABEBAAHCwXYEGAEIAAkFAl/hGf8CGwwA
IQkQIYHPWnTzRfEWIQQ3L6XpSGmPd9Gzr6ohgc9adPNF8TMBD/9TbU/+PVbF
ywKvwi3GLOlpY7BXn8lQaHyunMGuavmO8OfaRROynkH0ZqLHCp6bIajFOfvF
b7c0Jamzx8Hg+SIdl6yRpRY+fA4RQ6PNnnmT93ZgWW3EbjPyJGlm0/rt03SR
+0yn4/ldlg2KfBX4pqMoPCMKUdWxGrmDETXsGihwZ0gmCZqXe8lK122PYkSN
JQQ+LlfjKvCaxfPKEjXYTbIbfyyhCR6NzAOVZxCrzSz2xDrYWp/V002Klxda
0ix6r2aEHf+xYEUhOaBt8OHY5nXTuRReCVU789MUVtCMqD2u6amdo4BR0kWA
QNg4yavKwV+LVtyYh2Iju9VSyv4xL1Q4xKHvcAUrSH73bHG7b7jkUJckD0f4
twhjJk/Lfwe6RdnVo2WoeTvE93w+NAq2FXmvbiG7eltl0XfQecvQU3QNbRvH
U8B96W0w8UXJdvTKg4f0NbjSw7iJ3x5naixQ+rA8hLV8xOgn2LX6wvxT/SEu
mn20KX+fPtJELK7v/NheFLX1jsKLXYo4jHrkfIXNsNUhg/x2E71kAjbeT3s+
t9kCtxt2iXDDZvpIbmGO4QkvLFvoROaSmN6+8fupe3e+e2yN0e6xGTuE60gX
I2+X1p1g9IduDYTpoI2OXleHyyMqGEeIb4gOiiSloTp5oi3EuAYRGflXuqAT
VA19bKnpkBsJ0A==
=tD2T
-----END PGP PUBLIC KEY BLOCK-----
```

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

GPG

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

Example output

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

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

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

This command returns output similar to the following.

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

The fingerprint should match the following:

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

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

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

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

GPG

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

This command returns output similar to the following.

GPG

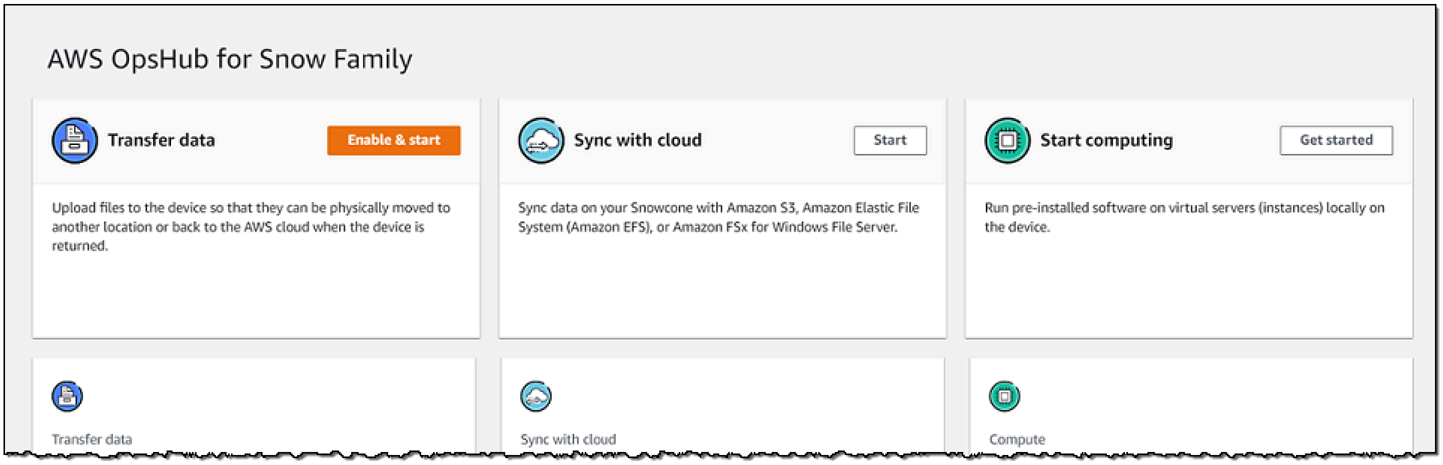
```
gpg: Signature made Mon Dec 21 13:44:47 2020 PST
gpg:                using RSA key 1655BBDE2B770256
gpg: Good signature from "AWS OpsHub for Snow Family <aws-opshub-
signer@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication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

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

Managing AWS services on your device

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

- Amazon Elastic Compute Cloud (Amazon EC2) instances – Use Amazon EC2-compatible 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.



Upload files to the device so that they can be physically moved to another location or back to the AWS cloud when the device is returned.

Sync data on your Snowcone with Amazon S3, Amazon Elastic File System (Amazon EFS), or Amazon FSx for Windows File Server.

Run pre-installed software on virtual servers (instances) locally on the device.

Topics

- Using Amazon EC2-compatible compute instances locally
- Managing the NFS interface

Using Amazon EC2-compatible 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-compatible instance
- Stopping an Amazon EC2-compatible instance
- Starting an Amazon EC2-compatible instance
- Working with key pairs
- Terminating an Amazon EC2-compatible instance
- Using storage volumes locally

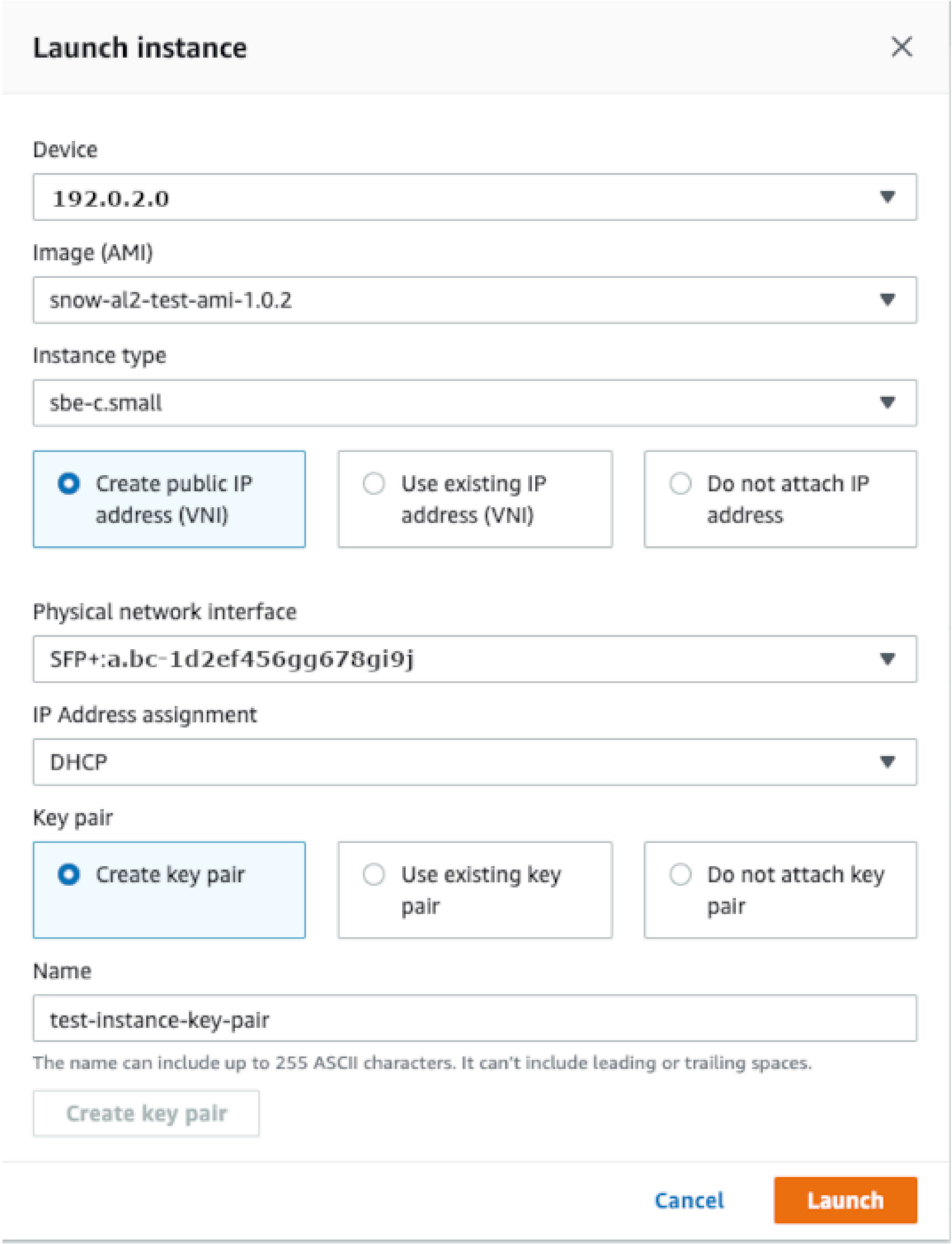
Launching an Amazon EC2-compatible instance

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

To launch an Amazon EC2-compatible 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-compatible 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-compatible.



Launch instance ✕

Device

192.0.2.0 ▼

Image (AMI)

snow-al2-test-ami-1.0.2 ▼

Instance type

sbe-c.small ▼

● Create public IP address (VNI) ○ Use existing IP address (VNI) ○ Do not attach IP address

Physical network interface

SFP+:a.bc-1d2ef456gg678gi9j ▼

IP Address assignment

DHCP ▼

Key pair

● Create key pair ○ Use existing key pair ○ Do not attach key pair

Name

test-instance-key-pair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Create key pair

Cancel Launch

6. For **Image (AMI)**, choose an Amazon Machine Image (AMI) from the list. This AMI is used to launch your instance.
7. For **Instance type**, choose one from the list.
8. Choose how you want to attach an IP address to the instance. You have the following options:

   - **Create public IP address (VNI)** – Choose this option to create a new IP address using a physical network interface. Choose a physical network interface and IP address assignment.
   - **Use existing IP address (VNI)** – Choose this option to use an existing IP address and then use existing virtual network interfaces. Choose a physical network interface and a virtual network interface.
   - **Do not attach IP address** – Choose this option if you don't want to attach an IP address.
9. Choose how you want to attach a key pair to the instance. You have the following options:

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

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

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

   For more information, see Working with key pairs.
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-compatible instance

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

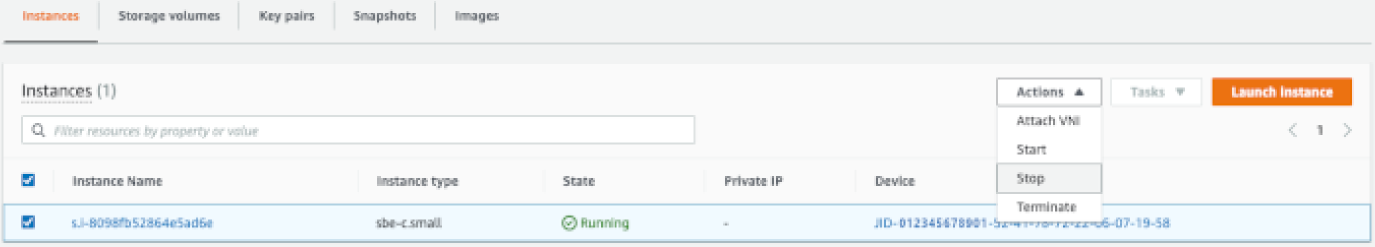
To stop an Amazon EC2-compatible instance

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-compatible instances running on your device, they appear in the **Instance name** column under **Instances**.

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



☑	Instance Name	Instance type	State	Private IP	Device
☑	s.i-8098fb52864e5ad6e	sbe-c.small	✓ Running	-	JID-012345678901-52-41-78-72-22-06-07-19-58

Starting an Amazon EC2-compatible instance

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

To start an Amazon EC2-compatible instance

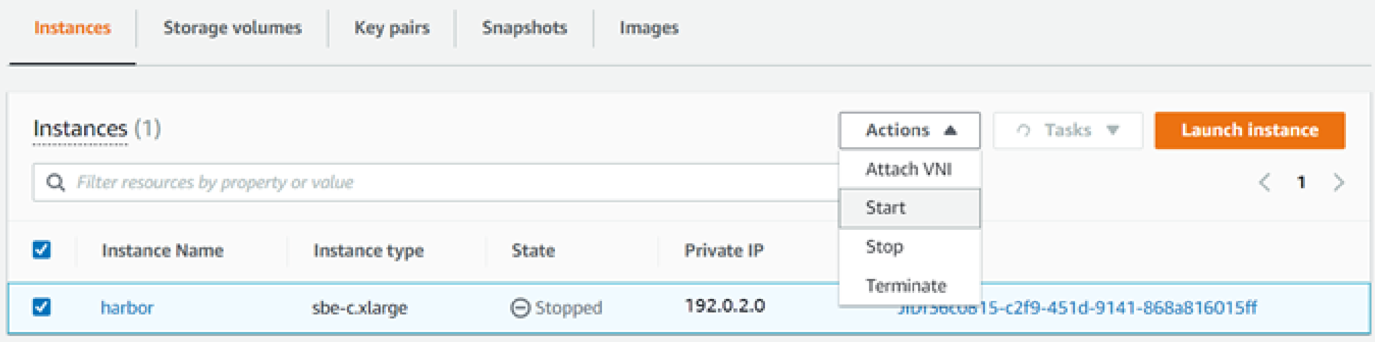
1. Open the AWS OpsHub application.

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.



☑	Instance Name	Instance type	State	Private IP	
☑	harbor	sbe-c.xlarge	⊖ Stopped	192.0.2.0	JID156c0815-c2f9-451d-9141-868a816015ff

Working with key pairs

When you launch an Amazon EC2-compatible 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

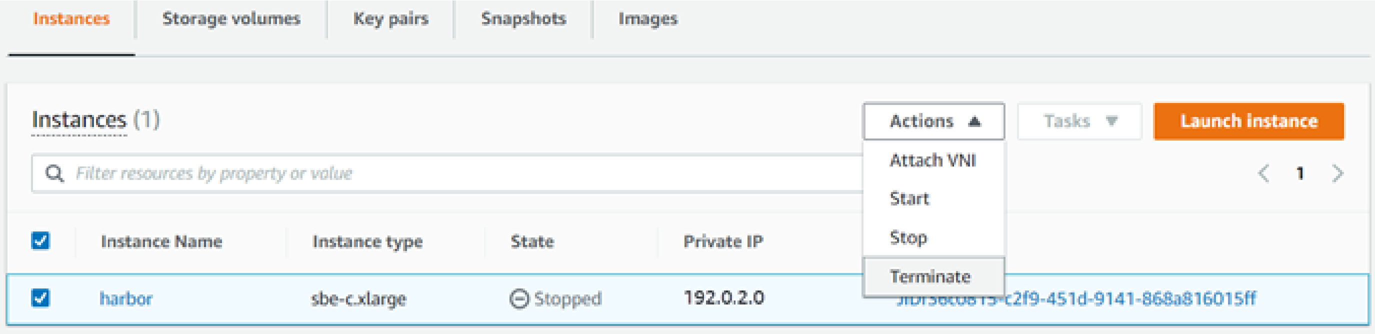
1. Open **Compute** on the AWS OpsHub dashboard.
2. In the navigation pane, choose the **Compute (EC2)** page, and then choose the **Key Pairs** tab. You are redirected to the Amazon EC2 console where you can create, import, or manage your key pairs.
3. For instructions on how to create and import key pairs, see Amazon EC2 key pairs and Linux instances in the *Amazon EC2 User Guide*.

Terminating an Amazon EC2-compatible instance

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

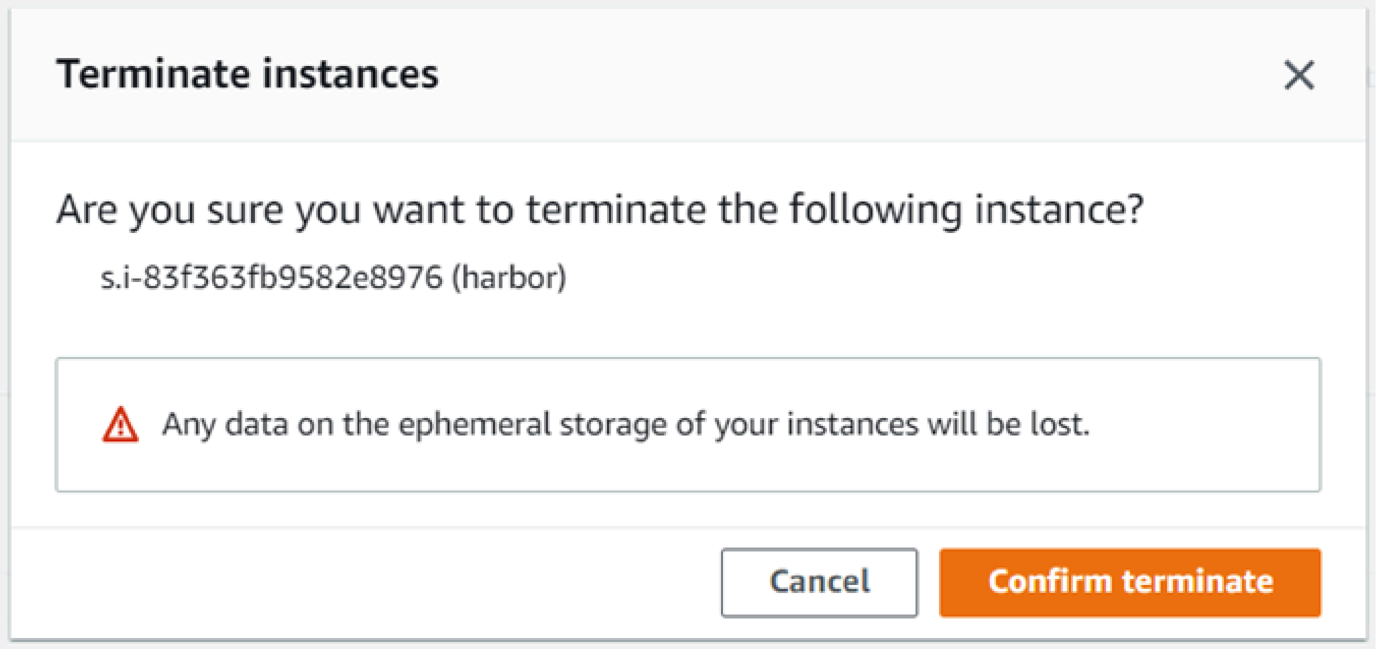
To terminate an Amazon EC2-compatible instance

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 the **Actions**menu. From the **Actions** menu, choose **Terminate**.



☑	Instance Name	Instance type	State	Private IP	
☑	harbor	sbe-c.xlarge	⊖ Stopped	192.0.2.0	c2f9-451d-9141-868a816015ff

5. In the **Terminate instances window, choose Confirm terminate**.



Terminate instances

Are you sure you want to terminate the following instance?

s.i-83f363fb9582e8976 (harbor)

⚠ Any data on the ephemeral storage of your instances will be lost.

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

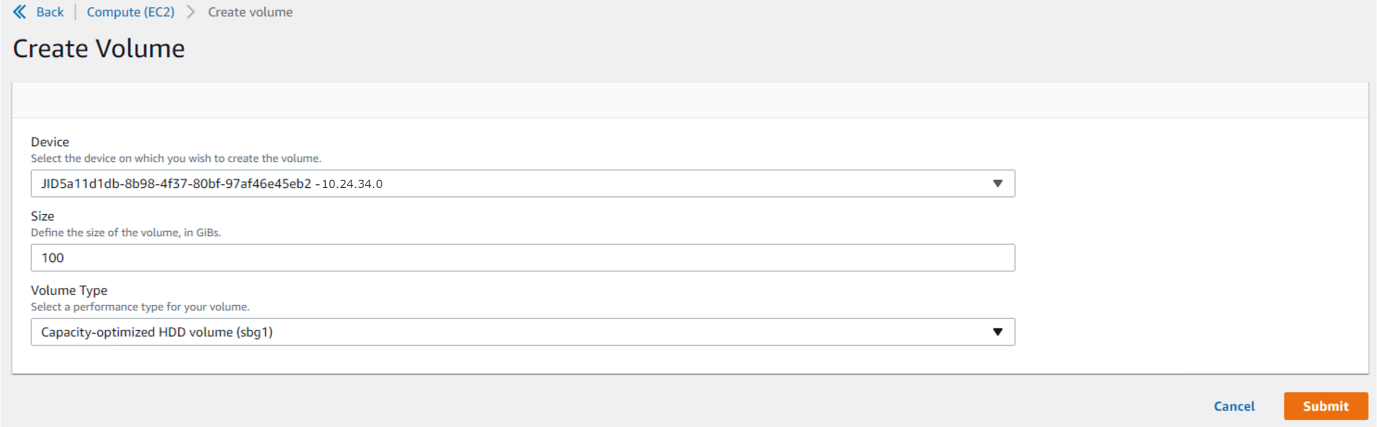
The **State** changes to **Terminating**, and then to **Terminated** when done.

Using storage volumes locally

Amazon EC2-compatible 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.



Create Volume

Device
Select the device on which you wish to create the volume.
JID5a11d1db-8b98-4f37-80bf-97af46e45eb2 - 10.24.34.0

Size
Define the size of the volume, in GiBs.
100

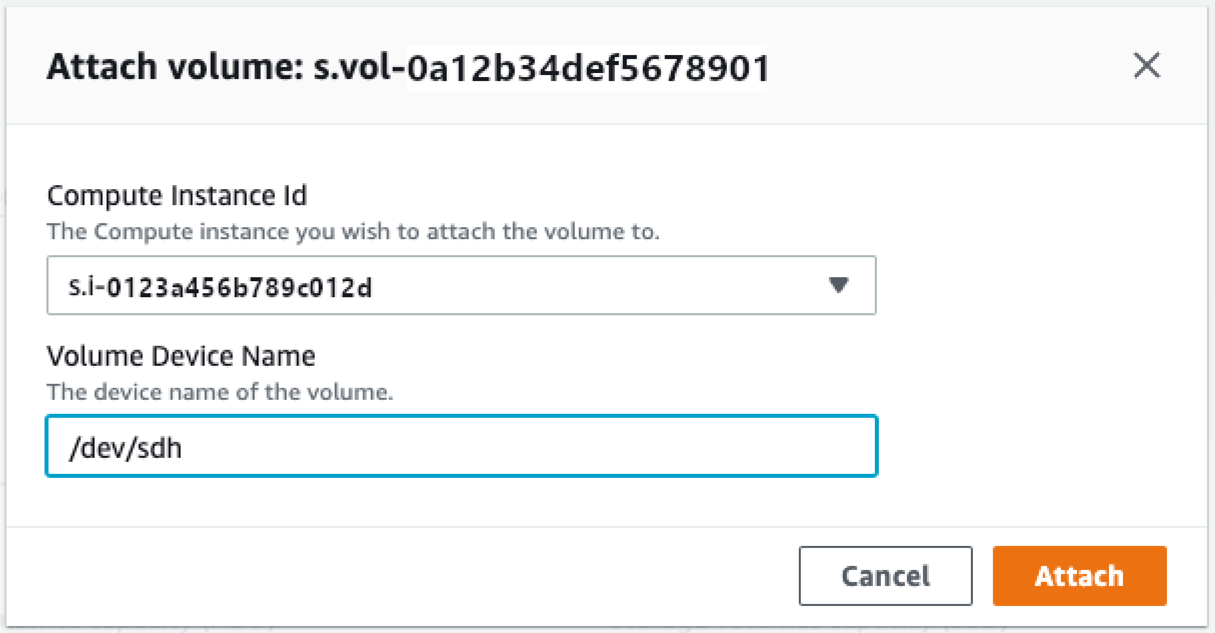
Volume Type
Select a performance type for your volume.
Capacity-optimized HDD volume (sbg1)

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**.


Attach volume: s.vol-0a12b34def5678901

Compute Instance Id
The Compute instance you wish to attach the volume to.
s.i-0123a456b789c012d

Volume Device Name
The device name of the volume.
/dev/sdh

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.

Managing the NFS interface

Use the Network File System (NFS) interface to upload files to the Snow Family device as if the device is local storage to your operating system. This allows for a more user-friendly approach to transferring data because you can use features of your operating system, like copying files, dragging and dropping them, or other graphical user interface features. Each S3 bucket on the device is available as an NFS interface endpoint and can be mounted to copy data to. The NFS interface is available for import jobs.

When started, the NFS interface uses 1 GB of memory and 1 CPU. This may limit the number of other services running on the Snow Family device or the number of EC2-compatible instances that can run.

Data transferred through the NFS interface is not encrypted in transit. When configuring the NFS interface, you can provide CIDR blocks and the Snow Family device will restrict access to the NFS interface from client computers with addresses in those blocks.

Files on the device will be transferred to Amazon S3 when it is returned to AWS. For more information, see How AWS Snowcone Works.

For more information about using NFS with your computer operating system, see the documentation for your operating system.

Keep the following details in mind when using the NFS interface.

- File names are object keys in your local S3 bucket on the Snow Family device. The key name is a sequence of Unicode characters whose UTF-8 encoding is at most 1,024 bytes long. We recommend using NFSv4.1 where possible and encode file names with Unicode UTF-8 to ensure a successful data import. File names that are not encoded with UTF-8 might not be uploaded to S3 or might be uploaded to S3 with a different file name depending on the NFS encoding you use.
- Ensure that the maximum length of your file path is less than 1024 characters. Snow Family devices do not support file paths that are greater that 1024 characters. Exceeding this file path length will result in file import errors.
- For more information, see Object keys in the Amazon Simple Storage Service User Guide.
- For NFS based transfers, standard POSIX style meta-data will be added to your objects as they get imported to Amazon S3 from Snow Family devices. In addition, you will 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 Snow Family device import with NFS option.

> **Note**
>
> In the device details page in AWS OpsHub, available storage space on Snowcone devices is not accurate until the NFS interface is started.

You can also configure and manage the NFS interface with the Snowball Edge client, a command line interface (CLI) tool. For more information, see Managing the NFS interface.

Topics

- Starting the NFS service on a Windows operating system
- Configuring the NFS interface automatically
- Configuring the NFS interface manually
- Managing NFS endpoints on the Snow Family device
- Mounting NFS endpoints on client computers
- Stopping the NFS interface

Starting the NFS service on a Windows operating system

If your client computer is using the Windows 10 Enterprise or Windows 7 Enterprise operating system, start the NFS service on the client computer before configuring NFS in the AWS OpsHub application.

1. On your client computer, open **Start**, choose **Control Panel** and choose **Programs**.
2. Choose **Turn Windows features on or off**.

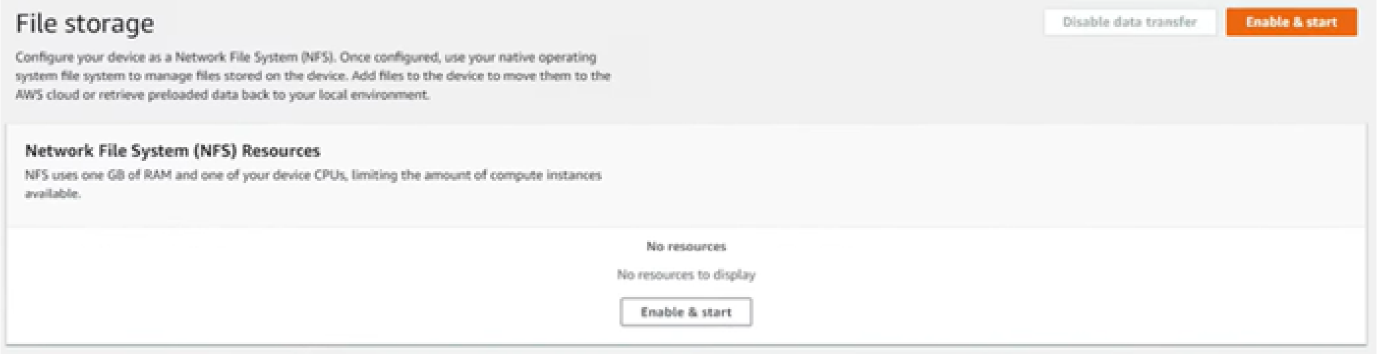
   > **Note**
   >
   > To turn Windows features on, you may need to provide an admin user name and password for your computer.

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

Configuring the NFS interface automatically

The NFS interface is not running on the Snow Family 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 quickly and automatically configure the NFS interface for you. You can also configure the NFS interface yourself. For more information, see Configuring the NFS interface manually.

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



File storage

Configure your device as a Network File System (NFS). Once configured, use your native operating system file system to manage files stored on the device. Add files to the device to move them to the AWS cloud or retrieve preloaded data back to your local environment.

Network File System (NFS) Resources

NFS uses one GB of RAM and one of your device CPUs, limiting the amount of compute instances available.

No resources

No resources to display

2. When the NFS service is started, the IP address of the NFS interface is shown on the dashboard and the **Transfer data** section indicates that the service is active.
3. Choose **Open in Explorer** (if using a Windows or a Linux operating system) to open the file share in your operating system's file browser and start transferring files to the Snow Family device. You can copy and paste or drag and drop files from your client computer into the file share. In Windows operating system, your file share looks like the following `buckets(\\12.123.45.679)(Z:)`.

> **Note**
>
> In Linux operating systems, mounting NFS endpoints requires root permissions.

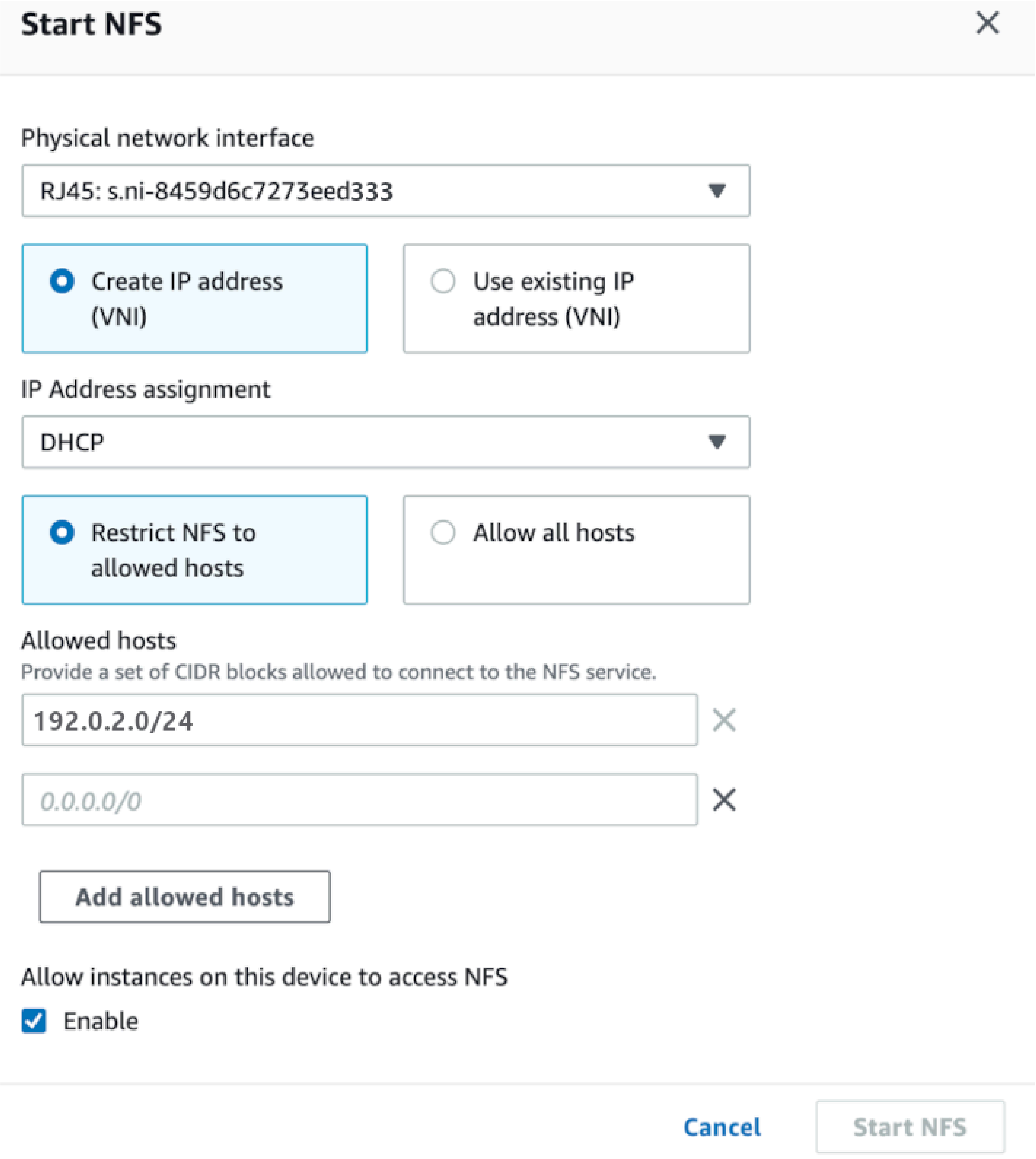
Configuring the NFS interface manually

The NFS interface is not running on the Snow Family device by default, so you need to start it to enable data transfer on the device. You can manually configure the NFS interface by providing the IP address of a Virtual Network Interface (VNI) running on the Snow Family device and restricting access to your file share, if required. Before configuring the NFS interface manually, set up a virtual network interface (VNI) on your Snow Family device. For more information, see Network Configuration for Compute Instances.

You can also have the Snow Family device configure the NFS interface automatically. For more information, see Configuring the NFS interface automatically.

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.



Start NFS

Physical network interface

RJ45: s.ni-8459d6c7273eed333

● Create IP address (VNI)

○ Use existing IP address (VNI)

IP Address assignment

DHCP

● Restrict NFS to allowed hosts

○ Allow all hosts

Allowed hosts

Provide a set of CIDR blocks allowed to connect to the NFS service.

192.0.2.0/24

0.0.0.0/0

Allow instances on this device to access NFS

☑ Enable

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

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

5. Choose to restrict access to the NFS interface and provide a block of allowed network addresses, or allow any devices on the network to access the NFS interface on the Snow Family device.

   - To restrict access to the NFS interface on the Snow Family device, choose **Restrict NFS to allowed hosts**. In **Allowed hosts** enter a set of CIDR blocks. If you want to allow access to more than one CIDR block, enter another set of blocks. To remove a set of blocks, choose **X** next to the field containing the blocks. Choose **Add allowed hosts**.

     > ⓘ **Note**
     >
     > If you choose **Restrict NFS to allowed hosts** and do not provide allowed CIDR blocks, the Snow Family device will deny all requests to mount the NFS interface.

   - To allow any device on the network to access the NFS interface, choose **Allow all hosts**.

6. To allow EC2-compatible instances running on the Snow Family device to access the NFS adapter, choose **Enable**.

7. Choose **Start NFS**. It could take about a minute or two to start.

   > ⚠ **Important**
   >
   > Don't turn off the Snow Family device while the NFS interface is starting.

From the **Network File System (NFS) Resources** section, the **State** of the NFS interface shows as **Active**. You will need the IP address listed to mount the interface as local storage on client computers.

Managing NFS endpoints on the Snow Family device

Each S3 bucket on the Snow Family device is represented as an endpoint and listed in **Mount paths**. After the NFS interface is started, mount an endpoint to transfer files to or from that endpoint. Only one endpoint can be mounted at a time. To mount a different endpoint, unmount the current endpoint first.

To mount an endpoint

1. In the **Mount paths** section, do one of the following to select an endpoint:
   - In the **Filter endpoints** field, enter all or part a bucket name to filter the list of available endpoints on your entry, then choose the endpoint.
   - Choose the endpoint to mount in the **Mount paths** list.
2. Choose **Mount NFS endpoint**. The Snow Family device mounts the endpoint for use.

To unmount an endpoint

1. In the **Mount paths** section, choose the endpoint to unmount.
2. Choose **Unmount endpoint**. The Snow Family device unmounts the endpoint and it is no longer available for use.

> **Note**
>
> Before unmounting an endpoint, ensure no data is being copied from or to it.

Mounting NFS endpoints on client computers

After the NFS interface is started and an endpoint mounted, mount the endpoint as local storage on client computers.

1. In **Mount paths**, choose the copy icon of the endpoint to mount. Paste it in your operating system when mounting the endpoint.
2. The following are the default mount commands for Windows, Linux, and macOS operating systems.

   - Windows:

     ```
     mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/
     buckets/BucketName *
     ```

   - Linux:

     ```
     mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
     ```

   - macOS:

     ```
     mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-
     interface-ip-address:/buckets/$bucketname mount_point
     ```

Stopping the NFS interface

Stop the NFS interface on the Snow Family device when you are done transferring files to or from it.

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. Before starting the DataSync agent, enable NFS on your Snowcone device. See Configuring the NFS interface automatically and Starting NFS and Restricting Access.
2. 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.
3. Choose **Create IP address (VNI)** to create a virtual IP address 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.

   If you choose **Use existing IP address**, then choose a virtual interface from the **Virtual network interface** list box.
5. 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.
6. 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*.

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
- Shutting down your device
- Editing Your Device Alias
- Getting Updates for Your Device and the AWS OpsHub Application
- Managing profiles

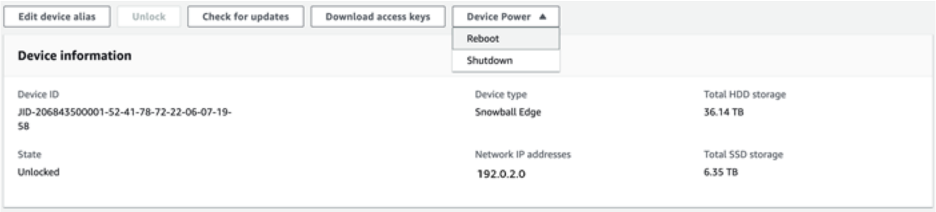
Rebooting your device

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 and interrupts any writing to Amazon S3 buckets on the device.

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.



Device information

Device ID
JID-206843500001-52-41-78-72-22-06-07-19-58

Device type
Snowball Edge

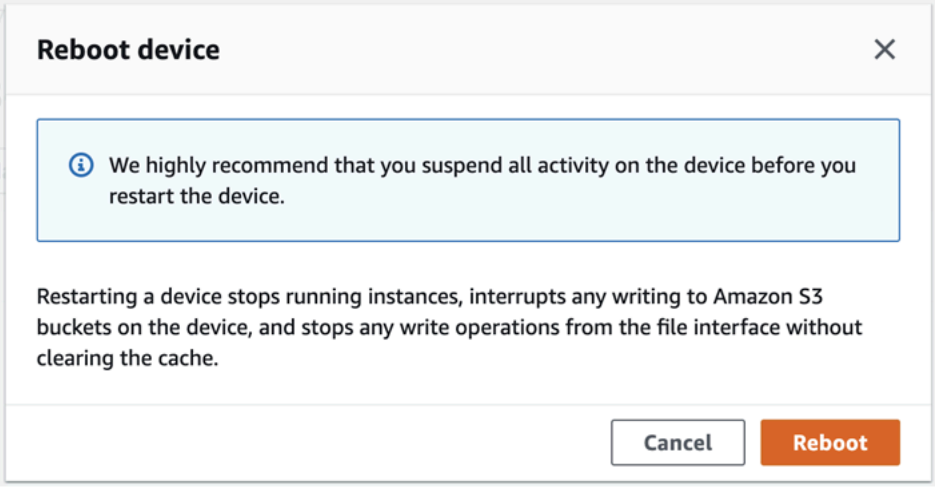
Total HDD storage
36.14 TB

State
Unlocked

Network IP addresses
192.0.2.0

Total SSD storage
6.35 TB

3. In the dialog box, choose **Reboot**. Your device starts to reboot.



Reboot device

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.

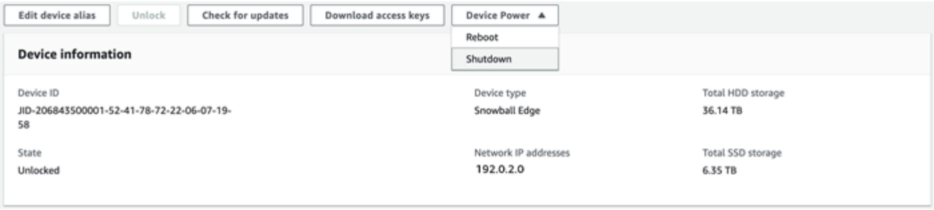
Shutting down your device

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 and interrupts any writing to Amazon S3 buckets on the device.

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.



Device information

Device ID
JID-206843500001-52-41-78-72-22-06-07-19-58

Device type
Snowball Edge

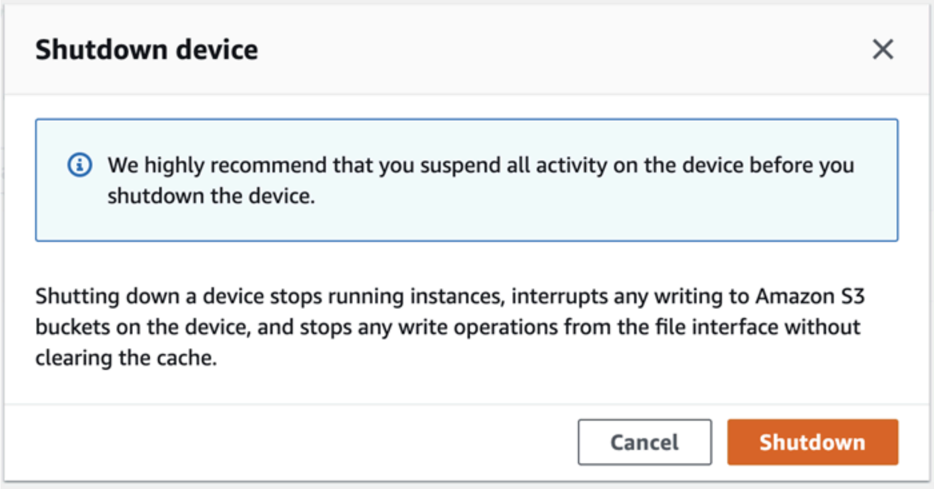
Total HDD storage
36.14 TB

State
Unlocked

Network IP addresses
192.0.2.0

Total SSD storage
6.35 TB

3. In the dialog box, choose **Shutdown**. Your device starts to shut down.



Shutdown device

We highly recommend that you suspend all activity on the device before you shutdown the device.

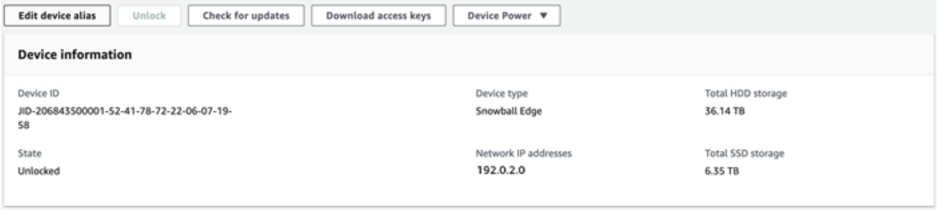
Shutting down a device stops running instances, interrupts any writing to Amazon S3 buckets on the device, and stops any write operations from the file interface without clearing the cache.

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.



Device information

Device ID
JID-206843500001-52-41-78-72-22-06-07-19-58

Device type
Snowball Edge

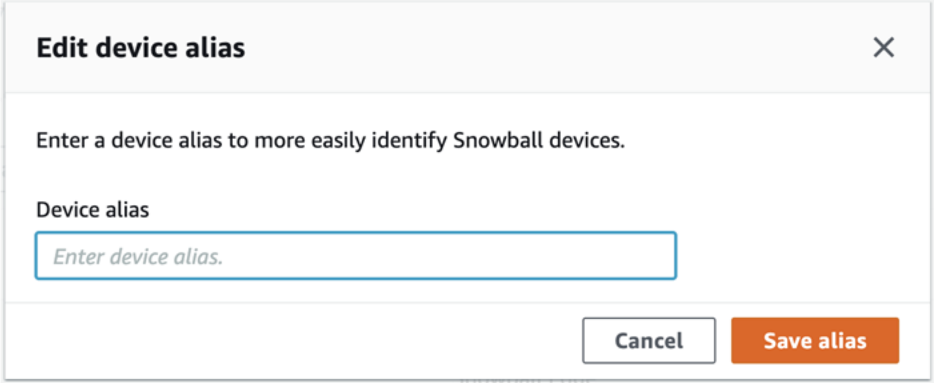
Total HDD storage
36.14 TB

State
Unlocked

Network IP addresses
192.0.2.0

Total SSD storage
6.35 TB

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



Edit device alias

Enter a device alias to more easily identify Snowball devices.

Device alias
Enter device 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.

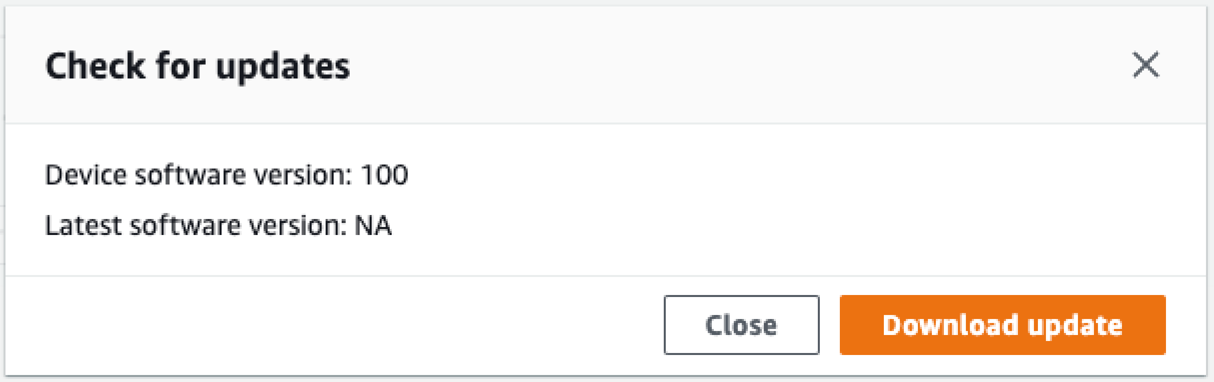
Updating your device

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.

   Check for updates

   Device software version: 100

   Latest software version: NA

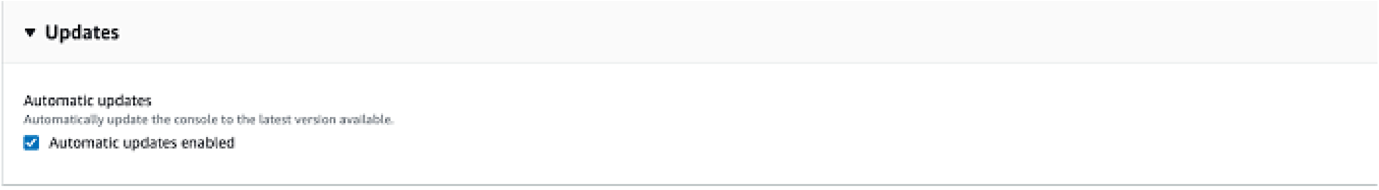
3. If there is an update, choose **Download update**. Otherwise, choose **Close**.

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.



▾ Updates

Automatic updates

Automatically update the console to the latest version available.

☑ Automatic updates enabled

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

Managing profiles

You can create a *profile* for persistent storage of your credentials on your local file system. Using AWS OpsHub, you have the option to create a new profile any time you unlock the device using the device IP address, unlock code, and manifest file.

You can also use the Snowball Edge Client to create a profile at any time. See Configuring a Profile for the Snowball Edge Client.

To edit or delete profiles, edit the profile file in a text editor.

Example Example `snowball-edge.config` file

This example shows a profile file containing three profiles—`SnowDevice1profile`, `SnowDevice2profile`, and `SnowDevice3profile`.

```
{"version":1,"profiles":
    {
    "SnowDevice1profile":
        {
            "name":"SnowDevice1profile",
            "jobId":"JID12345678-136f-45b4-b5c2-847db8adc749",
            "unlockCode":"db223-12345-dbe46-44557-c7cc2",
            "manifestPath":"C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\
\JID12345678-136f-45b4-b5c2-847db8adc749_manifest-1670622989203.bin",
            "defaultEndpoint":"https://10.16.0.1",
            "isCluster":false,
            "deviceIps":[]
        },
    },
    "SnowDevice2profile":
    {
        "name":"SnowDevice2profile",
        "jobId":"JID12345678-fdb2-436a-a4ff-7c510dec1bae",
        "unlockCode":"b893b-54321-0f65c-6c5e1-7f748",
        "manifestPath":"C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\\JID12345678-
fdb2-436a-a4ff-7c510dec1bae_manifest-1670623746908.bin",
        "defaultEndpoint":"https://10.16.0.2",
        "isCluster":false,
        "deviceIps":[]
    },
    "SnowDevice3profile":
    {
        "name":"SnowDevice3profile",
        "jobId":"JID12345678-c384-4a5e-becd-ab5f38888463",
        "unlockCode":"64c89-13524-4d054-13d93-c1b80",
        "manifestPath":"C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\\JID12345678-
c384-4a5e-becd-ab5f38888463_manifest-1670623999136.bin",
        "defaultEndpoint":"https://10.16.0.3",
        "isCluster":false,
        "deviceIps":[]
    }
}
```

To create a profile

1. Unlock your device locally and sign in according to the instructions in Unlocking a device.
2. Name the profile and choose **Save profile name**.

To edit a profile

1. In a text editor, open `snowball-edge.config` from `home directory\.aws\snowball\config`.
2. Edit the file as necessary. For example, to change the IP address of a device in the profile, change the `defaultEndpoint` entry.
3. Save and close the file.

To delete a profile

1. Using a text editor, open `snowball-edge.config` from `home directory\.aws\snowball\config`.

2. Delete the line that contains the profile name, the curly brackets `{ }`that follow the profile name, and the contents within the those brackets.
3. Save and close the file.

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

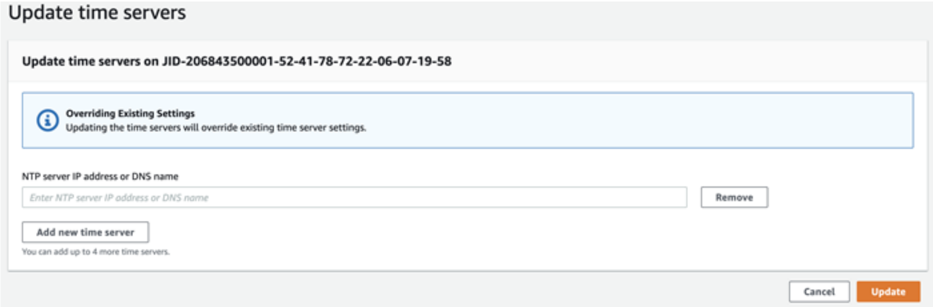
1. On the AWS OpsHub dashboard, find your device under **Devices**. Choose the device to open the device details page.
2. You will see a list of time sources that your device is synchronizing time with in the **Time sources** table.

   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**.



Update time servers

Update time servers on JID-206843500001-52-41-78-72-22-06-07-19-58

Overriding Existing Settings
Updating the time servers will override existing time server settings.

NTP server IP address or DNS name

Enter NTP server IP address or DNS name

You can add up to 4 more time servers.

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 NFS 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.

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

- Setting Time Servers
- Checking Time Sources

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.

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.

> **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.

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 .

`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 and interrupts any writing to Amazon S3 buckets on the device. 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 and interrupts any writing to Amazon S3 buckets on the device. 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.

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 interface or AWS DataSync. If you unlock your Snowcone device with a different IP address, a new certificate is generated, and the old certificate is no longer valid to use with the endpoint. You can get the new, updated certificate from the Snowcone device again using the `get-certificate` command.

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

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

  Usage (configured Snowball Edge client)

  ```
  snowballEdge list-certificates
  ```

Example Example Output

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

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

Usage (configured Snowball Edge client)

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

Example Example Output

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

AWS Snowcone Logs

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

There are three commands related to logs:

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

Usage (configured Snowball Edge client)

```
snowballEdge list-logs
```

Example Example Output

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

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

Usage (configured Snowball Edge client)

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

Example Example Output

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

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

Usage (configured Snowball Edge client)

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

Example Example Output

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

Important

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

Getting Device Status

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

- `describe-device`

Usage (configured Snowball Edge client)

```
snowballEdge describe-device
```

Example Example Output

```
{
  "DeviceId" : "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**
>
> Available storage space on the Snowcone device is not accurate until the NFS service is started.
> 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. 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 installed on new devices in most AWS Regions where Snow Family devices are available.

With Snow Device Management, you can perform the following tasks:

- Create a task
- Check task status
- Check task metadata
- Cancel a task
- Check device info
- Check Amazon EC2-compatible instance state
- List commands and syntax
- List remote-manageable devices
- List task status across devices
- List available resources
- List tasks by status
- List device or task tags
- Apply tags
- Remove tags

Topics

- Choosing the Snow Device Management state when ordering a Snow Family device
- Activating Snow Device Management
- Adding permissions for Snow Device Management to an IAM role
- Snow Device Management CLI commands

Choosing the Snow Device Management state when ordering a Snow Family device

When you create a job to order a Snow device, you can choose which state Snow Device Management will be in when you receive the device: installed but not activated or installed and activated. If it is installed but not activated, you will need to use AWS OpsHub or the Snowball Edge client to activate it before using it. If it is installed and activated, you can use Snow Device Management after receiving the device and connecting it to your local network. You can choose the Snow Device Management state when creating a job to order a device through the AWS Snow Family Management Console, the Snowball Edge client, the AWS CLI, or the Snow job management API.

To choose the Snow Device Management state from the AWS Snow Family Management Console

1. To choose for Snow Device Management to be installed and activated, choose **Manage your Snow device remotely with AWS OpsHub or Snowball client**.
2. To choose for Snow Device Management to be installed but not activated, do not select **Manage your Snow device remotely with AWS OpsHub or Snowball client**.

For more information, see Step 3: Choose your features and options in this guide.

To choose the Snow Device Management state from the AWS CLI, Snowball Edge client, or Snow job management API:

- Use the `remote-management` parameter to specify the Snow Device Management state. The `INSTALLED_ONLY` value of the parameter means Snow Device Management is installed but not activated. The `INSTALLED_AUTOSTART` value of the parameter means Snow Device Management is installed and activated. If you don't specify a value for this parameter, `INSTALLED_ONLY` is the default value.

Example of the syntax of the `remote-management` parameter of the `create-job` command

```
aws snowball create-job \
    --job-type IMPORT \
    --remote-management INSTALLED_AUTOSTART
```

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

For more information, see Job Management API Reference in the AWS Snowball API Reference.

Activating Snow Device Management

Follow this procedure to activate Snow Device Management using the Snowball Edge client.

Before using this procedure, do the following:

- Download and install the latest version of the Snowball Edge client. For more information, see Downloading and Installing the Snowball Client.
- Download the manifest file and get the unlock code for the Snow Family device. For more information, see Getting Your Credentials and Tools.
- Connect the Snow Family device to your local network. For more information, see AWS Snowcone Device Specifications.
- Unlock the Snow Family device. For more information, see Unlocking a device locally.

```
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://192.0.2.0:9091
```

The Snowball Edge client returns the following when the command is successful.

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

Adding permissions for Snow Device Management to an IAM role

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 Managing AWS services on your device.

> **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
- Check task status
- Check device info
- Check Amazon EC2-compatible instance state
- Check task metadata
- Cancel a task
- List commands and syntax
- List remote-manageable devices
- List task status across devices
- List available resources
- List device or task tags
- List tasks by status
- Apply tags
- Remove tags

Create a task

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

Supported commands:

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

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

Command

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

Exceptions

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

Output

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

Check task status

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

A task can have one of the following states:

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

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

Command

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

Output

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

Check device info

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

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

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

Command

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

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

```
{
    "associatedWithJob": "JID2bf11d5a-ea1e-414a-b5b1-3bf7e6a6e111",
    "deviceCapacities": [
        {
            "available": 158892032000,
            "name": "HDD Storage",
            "total": 158892032000,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 0,
            "name": "SSD Storage",
            "total": 0,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 3,
            "name": "vCPU",
            "total": 3,
            "unit": "Number",
            "used": 0
        },
        {
            "available": 5368709120,
            "name": "Memory",
            "total": 5368709120,
            "unit": "Byte",
            "used": 0
        },
        {
            "available": 0,
            "name": "GPU",
            "total": 0,
            "unit": "Number",
            "used": 0
        }
    ],
    "deviceState": "UNLOCKED",
    "deviceType": "SNC1_HDD",
    "lastReachedOutAt": "2021-07-23T21:21:56.120000+00:00",
    "lastUpdatedAt": "2021-07-23T21:21:56.120000+00:00",
    "managedDeviceId": "smd-fictqic6gcldf111",
    "managedDeviceArn": "arn:aws:snow-device-management:us-west-2:000000000000:managed-
device/smd-fictqic6gcldf111"
    "physicalNetworkInterfaces": [
        {
            "defaultGateway": "10.0.0.1",
            "ipAddress": "10.0.0.2",
            "ipAddressAssignment": "DHCP",
            "macAddress": "ab:cd:ef:12:34:56",
            "netmask": "255.255.252.0",
            "physicalConnectorType": "RJ45",
            "physicalNetworkInterfaceId": "s.ni-530f866d526d4b111"
        },
        {
            "defaultGateway": "10.0.0.1",
            "ipAddress": "0.0.0.0",
            "ipAddressAssignment": "STATIC",
            "macAddress": "ab:cd:ef:12:34:57",
            "netmask": "0.0.0.0",
            "physicalConnectorType": "RJ45",
            "physicalNetworkInterfaceId": "s.ni-8abc787f0a6750111"
        }
    ],
    "software": {
        "installState": "NA",
        "installedVersion": "122",
        "installingVersion": "NA"
    },
    "tags": {
        "Project": "PrototypeA"
    }
}
```

Check Amazon EC2-compatible 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-compatible instance, use the following command. Replace each *`user input placeholder`* with your own information.

Command

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

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

Check task metadata

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

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

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

Command

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

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

Cancel a task

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

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

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

Command

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

Exceptions

```
ValidationException
ResourceNotFoundException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

List commands and syntax

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

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

Command

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

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

Command

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

List remote-manageable devices

To return a list of all devices on your account that have Snow Device Management enabled in the AWS Region where the command is run, use the `list-devices` command. `--max-results` and `--next-token` are optional. For more information, see Using AWS CLI pagination options in the "AWS Command Line Interface User Guide".

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

Command

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

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

List task status across devices

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

A task can have one of the following states:

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

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

Command

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

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

List available resources

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

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

Command

```
aws snow-device-management list-device-resources \
--managed-device-id smd-fictbgr3rbcje111 \
--type AWS::EC2::Instance
```

```
--next-
token YAQGPwAT9l3wVKaGYjt4yS34MiQLWvzcShe9oIeDJr05AT4rXSprqcqQhhBEYRfcerAp0YYbJmRT=
--max-results 10
```

Exceptions

```
ValidationException
InternalServerException
ThrottlingException
AccessDeniedException
```

Output

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

List device or task tags

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

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

Command

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

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
ThrottlingException
```

Output

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

List tasks by status

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

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

Command

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

Exceptions

```
ValidationException
InternalServerException
```

```
ThrottlingException
AccessDeniedException
```

Output

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

        }
    ]
}
```

Apply tags

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

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

Command

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

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
```

```
ThrottlingException
```

Remove tags

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

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

Command

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

Exceptions

```
AccessDeniedException
InternalServerException
ResourceNotFoundException
ThrottlingException
```

Using AWS Services on AWS Snowcone

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

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

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-compatible 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-compatible compute instances hosted on a Snowcone using the supported EC2-compatible 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 .

If the job type is local compute, you might create a total of 8 TiB local EBS volumes and attach them to Amazon EC2-compatible instances. This allows local EC2-compatible 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 to order a Snow Family device 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.

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 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 these operating systems:
  - Amazon Linux 2

    > **Note**
    >
    > The latest version of this AMI will be provided at the time your Snow Family device is being prepared to ship by AWS. To determine the version of this AMI on the device when you receive it, see Determining the version of the Amazon Linux 2 AMI for Snow Family.

  - CentOS 7 (x86_64) - with Updates HVM
  - Ubuntu 16.04 LTS - Xenial (HVM), Ubuntu 20.04 LTS - Focal, or Ubuntu 22.04 LTS - Jammy

    > **Note**
    >
    > Ubuntu 16.04 LTS - Xenial (HVM) images are no longer supported in the AWS Marketplace, but still supported for use on Snow Family device through Amazon EC2 VM Import/Export and running locally in AMIs.

  You can get these images from the AWS Marketplace.

  Before you add any AMIs to your job creation request, make sure that you have one or more supported AMIs in your AWS account.
- 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:
>
> - If you're going to use an AMI from the AWS Marketplace, make sure it has a supported product code and usage operation code. For more information, see Checking product and usage option codes for AWS Marketplace AMIs.
> - 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.
> - Using encrypted AMIs or encrypted Amazon EBS volumes on AWS Snowcone devices is not supported.

Checking product and usage option codes for AWS Marketplace AMIs

Before you begin the process to add an AMI from AWS Marketplace to your Snow Family device, ensure the product and usage codes of the AMI are supported in your AWS Region.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances and from which you will create the job to order the Snow Family device. You can select any Region that's available to you, regardless of your location.
3. In the navigation pane, choose **AMIs**.
4. Use the filter and search options to scope the list of displayed AMIs to see only the AMIs that match your criteria. For example, to list all Linux AMIs provided by AWS, choose **Public images**. Then use the search options to further scope the list of displayed AMIs.

(New console) Choose the **Search** bar and, from the menu, choose **Owner alias**, then the = operator, and then the value **amazon**. Choose the **Search** bar again to choose **Platform**, then the = operator, and then the operating system from the list provided.

(Old console) Choose the **Search** bar and, from the menu, choose **Owner** and then the value **Amazon images**. Choose the **Search** bar again to choose **Platform** and then the operating system from the list provided.

> **Note**
>
> AMIs from AWS Marketplace include **aws-marketplace** in the **Source** column.

5. In the **AMI ID** column, choose the AMI ID of the AMI.
6. In the **Image summary** of the AMI, ensure the **Product codes** are supported by your Region. For more information, see the table below.

> **Note**
>
> The product code avyfzznywektkgl5qv5f57ska is supported in all Regions.

Supported AWS Marketplace AMI product codes

AMI operating system	Product code
Ubuntu Server 14.04 LTS	b3dl4415quatdndl4qa6kcu45
CentOS 7 (x86_64)	aw0evgkw8e5c1q413zgy5pjce
Ubuntu 16.04 LTS	csv6h7oyg29b7epjzg7qdr7no
Amazon Linux 2	avyfzznywektkgl5qv5f57ska
Ubuntu 20.04 LTS	a8jyynf4hjutohctm41o2z18m
Ubuntu 22.04 LTS	47xbqns9xujfkkjt189a13aqe

7. For all Regions, ensure the **Usage operation** code is **RunInstances**.

Determining the version of the Amazon Linux 2 AMI for Snow Family

Use the following procedure to determine the version of the Amazon Linux 2 AMI for Snow Family on the Snow Family device. Install the latest version of the AWS CLI before continuing. For more information, see Install or update to the latest version of the AWS CLI in the AWS Command Line Interface User Guide.

- Use the `describe-images` AWS CLI command to see the description of the AMI. The version is contained in the description. Provide the public key certificate from the previous step. For more information, see describe-images in the AWS CLI Command Reference.

  ```
  aws ec2 describe-images --endpoint http://snow-device-ip:8008 --region snow
  ```

  Example of output of the `describe-images` command

  ```
  {
      "Images": [
          {
              "CreationDate": "2024-02-12T23:24:45.705Z",
              "ImageId": "s.ami-02ba84cb87224e16e",
              "Public": false,
              "ProductCodes": [
                  {
                      "ProductCodeId": "avyfzznywektkgl5qv5f57ska",
                      "ProductCodeType": "marketplace"
                  }
              ],
              "State": "AVAILABLE",
              "BlockDeviceMappings": [
                  {
                      "DeviceName": "/dev/xvda",
                      "Ebs": {
                          "DeleteOnTermination": true,
                          "Iops": 0,
                          "SnapshotId": "s.snap-0efb49f2f726fde63",
                          "VolumeSize": 8,
                          "VolumeType": "sbp1"
                      }
  ```

```
                }
            ],
            "Description": "Snow Family Amazon Linux 2 AMI 2.0.20240131.0 x86_64
 HVM gp2",
            "EnaSupport": false,
            "Name": "amzn2-ami-snow-family-hvm-2.0.20240131.0-x86_64-gp2-
b7e7f8d2-1b9e-4774-a374-120e0cd85d5a",
            "RootDeviceName": "/dev/xvda"
        }
    ]
}
```

In this example, the version of the Amazon Linux 2 AMI for Snow Family is **`2.0.20240131.0`**. It is found in the value of the `Description` name.

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.

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 .
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.*
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 to order a Snow Family device. 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.

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.

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-compatible instance on your Snow Family device. You must create a VNI for each of your EC2-compatible 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 will use the same physical interface (RJ45, SFP+, or QSFP) that is used to managed the Snow Family device. Creating a VNI on a different physical interface than the one being used for device management could lead to unexpected results.

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-compatible instance, with a maximum of 8 per device.

Topics

- Prerequisites
- Setting Up a Virtual Network Interface (VNI)
- Setting Up a Direct Network Interface (DNI)

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-compatible 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 http://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.
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).

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-compatible 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-compatible 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 command.

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

   - The first is to change ensure that packets meant for the VNI associated with the EC2-compatible instance are sent through eth0.
   - The second change configures your direct network interface to use either DCHP 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
MAINROUTETABLE=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=$(LANG=C ip -o link | awk -F ': ' -vIGNORECASE=1 '!/link\/
ieee802\.11/ && /'"$DNI_MAC"'/ { print $2 }')
ip link set $CURRENT_DEVICE_NAME name $DNI
```

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

Additional commands

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

Connecting to Your Compute Instance on a Snowcone Using SSH

To use SSH to connect to your compute instances on Snowcone devices, you must first provide the SSH key to the AMI before creating your job. For more information on that procedure, see Configure an AMI to Use SSH to Connect to Compute Instances Launched on the Device. 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.
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.

   ```
   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.*

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. 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.

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-compatible commands you can use on your AWS Snowcone device, see Supported Amazon EC2-compatible AWS CLI Commands on a Snowcone .

Note

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

Creating a Launch Configuration to Autostart Amazon EC2-compatible Instances

To automatically start Amazon EC2-compatible 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-compatible 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-compatible 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-compatible 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 NFS 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 Instance Metadata Service for Snow with Amazon EC2-compatible instances

IMDS for Snow provides Instance Metadata Service (IMDS) for Amazon EC2-compatible instances on Snow. Instance metadata is categories of information about instances. It includes categories such as *host name*, *events*, and *security groups*. Using IMDS for Snow, you can use instance metadata to access user data that you specified when launching your Amazon EC2-compatible instance. For example, you can use IMDS for Snow to specify parameters for configuring your instance, or include these parameters in a simple script. You can build generic AMIs and use user data to modify the configuration files supplied at launch time.

To learn about instance metadata and user data and Snow EC2-compatible instances, see Supported Instance Metadata and User Data in this guide.

> ⚠ **Important**
>
> Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who

has direct access to the instance, and potentially any software running on the instance, can view its metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

> **Note**
>
> The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. We do not support the retrieval of instance metadata using the link-local IPv6 address.

Topics

IMDS versions

You can access instance metadata from a running instance using IMDS version 2 or IMDS version 1:

- Instance Metadata Service version 2 (IMDSv2), a session-oriented method
- Instance Metadata Service version 1 (IMDSv1), a request-response method

Depending on the version of your Snow software, you can use IMDSv1, IMDSv2, or both. This also depends on the type of AMI running in the EC2-compatible instance. Some AMIs, such as those running Ubuntu 20.04, require IMDSv2. The instance metadata service distinguishes between IMDSv1 and IMDSv2 requests based on the presence of PUT or GET headers. IMDSv2 uses both of these headers. IMDSv1 uses only the GET header.

AWS encourages the use of IMDSv2 rather than IMDSv1 because IMDSv2 includes higher security. For more information, see Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service.

IMDSv2

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration. Session duration can be a minimum of one second and a maximum of six hours. During this duration, you can use the same session token for subsequent requests. After this duration expires, you must create a new session token for future requests.

The following example uses a Linux shell script and IMDSv2 to retrieve the top-level instance metadata items. This example:

1. Creates a session token lasting six hours (21,600 seconds) using the PUT request.
2. Stores the session token header in a variable named TOKEN.
3. Requests the top-level metadata items using the token.

You can run two commands separately, or combine them.

Separate commands

First, generate a token using the following command.

> **Note**
>
> X-aws-ec2-metadata-token-ttl-seconds is a required header. If this header is not included, you will receive an **400 - Missing or Invalid Parameters** error code.

```
    [ec2-user ~]$ TOKEN=curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"
```

Then, use the token to generate top-level metadata items using the following command.

```
    [ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/
```

Combined commands

You can store the token and combine the commands. The following example combines the above two commands and stores the session token header in a variable named TOKEN.

> **Note**
>
> If there is an error in creating the token, an error message is stored in the variable instead of a valid token and the command will not work.

Example of combined commands

```
    [ec2-user ~]$ TOKEN=curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600" \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
meta-data/
```

After you've created a token, you can reuse it until it expires. The following example command gets the ID of the AMI used to launch the instance and stores it in the `$TOKEN` created in the previous example.

Example of reusing a token

```
    [ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id
```

When you use IMDSv2 to request instance metadata, the request must follow these rules:

1. Use a PUT request to initiate a session to the instance metadata service. The PUT request returns a token that must be included in subsequent `GET` requests to the instance metadata service. The token is required to access metadata using IMDSv2.
2. Include the token in all `GET` requests to the instance metadata service.
    a. The token is an instance-specific key. The token is not valid on other EC2-compatible instances and will be rejected if you attempt to use it outside of the instance on which it was generated.
    b. The PUT request must include a header that specifies the time to live (TTL) for the token, in seconds, up to a maximum of six hours (21,600 seconds). The token represents a logical session. The TTL specifies the length of time that the token is valid and, therefore, the duration of the session.

c. After a token expires, to continue accessing instance metadata, you must create a new session using another PUT request.

d. You can choose to reuse a token or create a new token with every request. For a small number of requests, it might be easier to generate and immediately use a token each time you need to access the instance metadata service. But for efficiency, you can specify a longer duration for the token and reuse it rather than having to write a PUT request every time you need to request instance metadata. There is no practical limit on the number of concurrent tokens, each representing its own session.

HTTP `GET` and `HEAD` methods are allowed in IMDSv2 instance metadata requests. PUT requests are rejected if they contain an `X-Forwarded-For` header.

By default, the response to PUT requests has a response hop limit (time to live) of 1 at the IP protocol level. IMDS for Snow does not have ability to modify the hop limit on PUT responses.

IMDSv1

IMDSv1 uses the request-response model. To request instance metadata, you send a `GET` request to the instance metadata service.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

Retrieve instance metadata

Your instance metadata is available from your running instance, so you do not need to use Amazon EC2 console or the AWS CLI to access it. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application. Instance metadata is divided into categories. For a description of each instance metadata category, see Supported Instance Metadata and User Data in this guide.

To view all categories of instance metadata from within a running instance, use the following IPv4 URI:

```
http://169.254.169.254/latest/meta-data/
```

The IP addresses are link-local addresses and are valid only from the instance. For more information, see Link-local address on Wikipedia.

Responses and error messages

All instance metadata is returned as text (HTTP content type `text/plain`).

A request for a specific metadata resource returns the appropriate value, or an **404 - Not Found** HTTP error code, if the resource is not available.

A request for a general metadata resource (when the URI ends with a / character) returns a list of available resources, or an **404 - Not Found** HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII character code 10).

For requests made using IMDSv1, the following HTTP error codes can be returned:

- **400 - Missing or Invalid Parameters** — The PUT request is not valid.
- **401 - Unauthorized** — The GET request uses an invalid token. The recommended action is to generate a new token.
- **403 - Forbidden** — The request is not allowed or the instance metadata service is turned off.

Examples of retrieving instance metadata using IMDSv1 and IMDSv2

The following examples provide commands that you can use on a Linux instance.

Example of getting the available versions of the instance metadata

This example gets the available versions of the instance metadata. Each version refers to an instance metadata build when new instance metadata categories were released. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

IMDSv2

```
    [ec2-user ~]$ TOKEN=`curl -X PUT "http://192.0.2.0/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
 http://192.0.2.0/
    % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
 Dload  Upload   Total   Spent    Left  Speed
```

```
   100        56         100      56      0       0       3733     0    --:--:--
--:--:-- --:--:--  3733
   *   Trying 192.0.2.0...
   * TCP_NODELAY set
   * Connected to 192.0.2.0 (192.0.2.0) port 80 (#0)
   > GET / HTTP/1.1
   > Host: 192.0.2.0
   > User-Agent: curl/7.61.1
   > Accept: */*
   > X-aws-ec2-metadata-token:
MDAXcxNFLbAwJIYx8KzgNckcHTdxT4Tt69TzpKExlXKTULHIQnjEtXvD
   >
   * HTTP 1.0, assume close after body
   < HTTP/1.0 200 OK
   < Date: Mon, 12 Sep 2022 21:58:03 GMT
   < Content-Length: 274
   < Content-Type: text/plain
   < Server: EC2ws
   <
   1.0
   2007-01-19
   2007-03-01
   2007-08-29
   2007-10-10
   2007-12-15
   2008-02-01
   2008-09-01
   2009-04-04
   2011-01-01
   2011-05-01
   2012-01-12
   2014-02-25
   2014-11-05
   2015-10-20
   2016-04-19
   2016-06-30
   2016-09-02
   2018-03-28
   2018-08-17
   2018-09-24
   2019-10-01
   2020-10-27
   2021-01-03
   2021-03-23
```

```
* Closing connection 0
```

IMDSv1

```
[ec2-user ~]$ curl http://192.0.2.0/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
2018-08-17
2018-09-24
2019-10-01
2020-10-27
2021-01-03
2021-03-23
latest
```

Example of getting the top-level metadata items

This example gets the top-level metadata items. For information on top-level metadata items, see Supported Instance Metadata and User Data in this guide.

IMDSv2

```
    [ec2-user ~]$ TOKEN=`curl -X PUT "http://192.0.2.0/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
 http://192.0.2.0/latest/meta-data/
    ami-id
    hostname
    instance-id
    instance-type
    local-hostname
    local-ipv4
    mac
    network/
    reservation-id
    security-groups
```

IMDSv1

```
    [ec2-user ~]$ curl http://192.0.2.0/latest/meta-data/
    ami-id
    hostname
    instance-id
    instance-type
    local-hostname
    local-ipv4
    mac
    network/
    reservation-id
    security-groups
```

Example of getting values of top-level metadata

The following examples get the values of some of the top-level metadata items that were obtained in the preceding example. The IMDSv2 requests use the stored token that was created in the preceding example command, assuming it has not expired.

`ami-id` IMDSv2

```
    curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://192.0.2.0/latest/meta-data/
ami-id ami-0abcdef1234567890
```

`ami-id` IMDSv1

```
    curl http://192.0.2.0/latest/meta-data/ami-id ami-0abcdef1234567890
```

`reservation-id` IMDSv2

```
    [ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://192.0.2.0/
latest/meta-data/reservation-id r-0efghijk987654321
```

`reservation-id` IMDSv1

```
    [ec2-user ~]$ curl http://192.0.2.0/latest/meta-data/reservation-id \
    r-0efghijk987654321
```

`local-hostname` IMDSv2

```
    [ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://192.0.2.0/
latest/meta-data/local-hostname ip-00-000-00-00
```

`local-hostname` IMDSv1

```
    [ec2-user ~]$ curl http://192.0.2.0/latest/meta-data/local-hostname ip-00-000-00-00
```

Using the Amazon EC2-compatible Endpoint

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

Topics

Specifying the Amazon EC2-compatible 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-compatible 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-compatible 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.
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-compatible endpoint, you can programmatically manage your AMIs and compute instances on a Snowcone with Amazon EC2-compatible 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 Amazon EC2-compatible AWS CLI Commands on a Snowcone

You can manage your compute instances on a Snow Family device through an Amazon EC2-compatible endpoint. This type of endpoint supports many of the Amazon EC2 CLI commands and actions of the AWS SDKs. For information about installing and setting up the AWS CLI, including specifying which AWS Regions you want to make AWS CLI calls against, see the AWS Command Line Interface User Guide.

List of Supported Amazon EC2-compatible 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.
  - --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-compatible Endpoint as the AWS CLI Endpoint.

- 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-compatible Endpoint as the AWS CLI Endpoint.
- 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-compatible Endpoint as the AWS CLI Endpoint.

- 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-compatible Endpoint as the AWS CLI Endpoint.
- 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-compatible Endpoint as the AWS CLI Endpoint.
- 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-compatible Endpoint as the AWS CLI Endpoint.
- 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-compatible Endpoint as the AWS CLI Endpoint.
- modify-instance-attribute – Modifies an attribute of the specified instance. The following attributes are supported:
  - `instanceInitiatedShutdownBehavior`
  - `userData`
- revoke-security-group-egress – Removes one or more egress rules from a security group:
  - [--group-id `value`] – The ID of the security group
  - [--ip-permissions `value`] – One or more sets of IP permissions.
- revoke-security-group-ingress – Revokes one or more ingress rules to a security group. When calling `revoke-security-group-ingress`, you must specify a value for either `group-name` or `group-id`.
  - [--group-name `value`] – The name of the security group.

  - [--group-id `value`] – The ID of the security group.
  - [--ip-permissions `value`] – One or more sets of IP permissions.
  - [--protocol `value`] The IP protocol. Possible values are `tcp`, `udp`, and `icmp`. The `--port` argument is required unless the "all protocols" value is specified (-1).
  - [--port `value`] – For TCP or UDP, the range of ports to allow. A single integer or a range (minimum–maximum).

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

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

  - [--block-device-mappings `(list)`] – The block device mapping entries. The parameters `DeleteOnTermination`, `VolumeSize`, and `VolumeType` are supported. Boot volumes must be type `sbg1`.

    The JSON syntax for this command is as follows.

    ```
    {
       "DeviceName": "/dev/sdh",
       "Ebs":
       {
          "DeleteOnTermination": true|false,
          "VolumeSize": 100,
          "VolumeType": "sbp1"|"sbg1"
       }
    }
    ```
  - --count – Number of instances to launch. If a single number is provided, it is assumed to be the minimum to launch (defaults to 1). If a range is provided in the form `min:max`, then the

  first number is interpreted as the minimum number of instances to launch and the second is interpreted as the maximum number of instances to launch.

  - --image-id – The Snowcone AMI ID of the AMI, which you can get by calling `describe-images`. An AMI is required to launch an instance.
  - --InstanceInitiatedShutdownBehavior – By default, when you initiate a shutdown from your instance (using a command such as shutdown or poweroff), the instance stops. You can change this behavior so that it terminates instead. The parameters `stop` and `terminate` are supported. The default is `stop`. For more information, see Changing the instance initiated shutdown behavior in the *Amazon EC2 User Guide for Linux Instances*.
  - --instance-type – The `sbe` instance type.
  - --launch-template `structure` – The launch template to use to launch the instances. Any parameters that you specify in the `run-instances` command override the same parameters in the launch template. You can specify either the name or ID of a launch template, but not both.

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

  - --security-group-ids – One or more security group IDs. You can create a security group using CreateSecurityGroup. If no value is provided, the ID for the default security group is assigned to created instances.
  - --tag-specifications – The tags to apply to the resources during launch. You can only tag instances on launch. The specified tags are applied to all instances that are created during launch. To tag a resource after it has been created, use `create-tags`.
  - --user-data – The user data to make available to the instance. If you are using the AWS CLI, base64-encoding is performed for you, and you can load the text from a file. Otherwise, you must provide base64-encoded text.

- start-instances – Starts an `sbe` instance that you've previously stopped. All resources attached to the instance persist through starts and stops, but are erased if the instance is terminated.
  - --instance-ids – The IDs of one or more `sbe` instances that were stopped on the device.
- stop-instances – Stops an `sbe` instance that is running. All resources attached to the instance persist through starts and stops, but are erased if the instance is terminated.

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

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

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

Supported Amazon EC2 API Operations

Following, you can find Amazon EC2 API operations that you can use with a Snowcone device, with links to their descriptions in the *Amazon EC2 API Reference.* Amazon EC2 API calls require Signature Version 4 (SigV4) signing. If you're using the AWS CLI or an AWS SDK to make these API calls, the SigV4 signing is handled for you. Otherwise, you need to implement your own SigV4 signing solution.

- AssociateAddress – Associates an Elastic IP address with an instance or a network interface.
- AttachVolume – The following request parameters are supported:
  - `Device`
  - `InstanceId`
  - `VolumeId`
- AuthorizeSecurityGroupEgress – Adds one or more egress rules to a security group for use with a Snowcone device. Specifically, this action permits instances to send traffic to one or more destination IPv4 CIDR address ranges.

- AuthorizeSecurityGroupIngress – Adds one or more ingress rules to a security group. When calling AuthorizeSecurityGroupIngress, you must specify a value either for `GroupName` or `GroupId`.
- CreateVolume – The following request parameters are supported:
  - `SnapshotId`
  - `Size`
  - `VolumeType`
  - `TagSpecification.N`
- CreateLaunchTemplate – The following request parameters are supported:
  - `ImageId`
  - `InstanceType`
  - `SecurityGroupIds`
  - `TagSpecifications`
  - `UserData`
- CreateLaunchTemplateVersion
- CreateTags – The following request parameters are supported:
  - `AMI`
  - `Instance`
  - `Launch template`
  - `Security group`
- CreateSecurityGroup – Creates a security group on your Snowcone. You can create up to 50 security groups. When you create a security group, you specify a friendly name of your choice.
- DeleteLaunchTemplate
- DeleteLaunchTemplateVersions
- DeleteSecurityGroup – Deletes a security group. If you attempt to delete a security group that is associated with an instance, or is referenced by another security group, the operation fails with `DependencyViolation`.
- DeleteTags – Deletes the specified set of tags from the specified set of resources.
- DeleteVolume – The following request parameters are supported:
  - `VolumeId`
- DescribeAddresses

- DescribeImages
- DescribeInstanceAttribute – The following attributes are supported:
  - `instanceType`
  - `userData`
- DescribeLaunchTemplates
- DescribeLaunchTemplateVersions
- DescribeInstances
- DescribeSecurityGroups – Describes one or more of your security groups. `DescribeSecurityGroups` is a paginated operation. You can issue multiple API calls to retrieve the entire dataset of results.
- DescribeTags – With this command, the following filters are supported:
  - `resource-id`
  - `resource-type` – AMI or compute instance only
  - `key`
  - `value`
- DescribeVolume – The following request parameters are supported:
  - `MaxResults`
  - `NextToken`
  - `VolumeId.N`
- DetachVolume – The following request parameters are supported:
  - `Device`
  - `InstanceId`
  - `VolumeId`
- DisassociateAddress
- GetLaunchTemplateData
- ModifyLaunchTemplate
- ModifyInstanceAttribute – Only the `userData` attribute is supported.
- RevokeSecurityGroupEgress – Removes one or more egress rules from a security group.
- RevokeSecurityGroupIngress – Revokes one or more ingress rules to a security group. When calling RevokeSecurityGroupIngress, you must specify a value either for `group-name` or `group-id`.

- RunInstances –

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

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

Autostarting Amazon EC2-compatible instances with launch templates

You can automatically start your Amazon EC2-compatible 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-compatible 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-compatible instance on the Snowcone.

When you use autostart configurations on your Snowcone, you configure the parameters that you want your Amazon EC2-compatible 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-compatible instance when you launch that instance.

To automatically launch EC2-compatible instances on the Snowcone, take the following steps:

1. When you order your AWS Snowcone device, create a job to order a Snow Family device with compute instances. For more information, see Creating a Job with Compute Instances.
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-compatible AWS CLI Commands on a Snowcone .

   > **Note**
   >
   > The Amazon EC2 endpoint is the device endpoint.

4. Use the Snowball Edge client command `snowballEdge create-autostart-configuration` to bind your EC2-compatible launch template to your network configuration. For more information, see Creating a Launch Configuration to Autostart Amazon EC2-compatible Instances.
5. Reboot, then unlock your AWS Snowcone device. Your EC2-compatible 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-compatible instances on your AWS Snowcone device.

Using Block Storage with your Amazon EC2-compatible 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-compatible 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-compatible AWS CLI Commands on a Snowcone .

⚠ **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-compatible instances running on the AWS Snowcone device hosting the volumes.
- Volume types are limited to either capacity-optimized HDD (`sbg1`) or >performance-optimized SSD (`sbp1`). The default volume type is `sbg1`.
- Amazon EC2 root volumes always use the IDE driver. Additional Amazon EBS volumes preferentially use the Virtio driver if available. If the Virtio driver isn't available, SBE defaults to the IDE driver. The Virtio driver allows for better performance and is recommended.
- When creating Amazon EBS volumes, the `encrypted` parameter isn't supported. However, all data on your device is encrypted by default.
- Volumes can be from 1 GB to 8 TB in size.
- Up to 10 Amazon EBS volumes can be attached to a single EC2-compatible 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.

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-compatible AWS CLI Commands on a Snowcone . For more information about supported API operations, see Supported Amazon EC2 API Operations.

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.*

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.

Data	Description
`hostname`	The private IPv4 DNS hostname of the instance.
`instance-id`	The ID of this instance.
`instance-type`	The type of instance.
`local-hostname`	The private IPv4 DNS hostname of the instance.
`local-ipv4`	The private IPv4 address of the instance.
`mac`	The instance's media access control (MAC) address.
`network/interfaces/macs/`*`mac`*`/local-hostname`	The interface's local hostname.
`network/interfaces/macs/`*`mac`*`/local-ipv4s`	The private IPv4 addresses associated with the interface.
`network/interfaces/macs/`*`mac`*`/mac`	The instance's MAC address.
`network/interfaces/macs/`*`mac`*`/public-ipv4s`	The Elastic IP addresses associated with the interface.
`public-ipv4`	The public IPv4 address.
`public-keys/0/openssh-key`	Public key. Only available if supplied at instance launch time.
`reservation-id`	The ID of the reservation.
userData	Shell scripts to send instructions to an instance at launch.

Supported Instance Dynamic Data Categories on a Snowcone

Data	Description
instance-identity/document	JSON containing instance attributes. Only `instanceId` , `imageId`, `privateIp` , and `instanceType` have values, and the other returned attributes are null. For more information, see [Instance identity documents](#) in the *Amazon EC2 User Guide*.

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

- Unprotected Private Key File Error

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.
- Update an existing VNIC. For more information, see Updating a Virtual Network Interface.

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.

Before starting the DataSync agent, enable Network File System (NFS) on your Snowcone device. See Configuring the NFS interface automatically and Starting NFS and Restricting Access.

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 inAWS 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 AWS DataSync task quotas for the maximum number of files per task for a Snowcone device. 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.

Managing the NFS interface

Use the Network File System (NFS) interface to upload files to the Snow Family device as if the device is local storage to your operating system. This allows for a more user-friendly approach to transferring data because you can use features of your operating system, like copying files, dragging and dropping them, or other graphical user interface features. Each S3 bucket on the device is available as an NFS interface endpoint and can be mounted to copy data to. The NFS interface is available for import jobs.

When started, the NFS interface uses 1 GB of memory and 1 CPU. This may limit the number of other services running on the Snow Family device or the number of EC2-compatible instances that can run.

Data transferred through the NFS interface is not encrypted in transit. When configuring the NFS interface, you can provide CIDR blocks and the Snow Family device will restrict access to the NFS interface from client computers with addresses in those blocks.

Files on the device will be transferred to Amazon S3 when it is returned to AWS. For more information, see How AWS Snowcone Works.

For more information about using NFS with your computer operating system, see the documentation for your operating system.

Keep the following details in mind when using the NFS interface.

- File names are object keys in your local S3 bucket on the Snow Family device. The key name is a sequence of Unicode characters whose UTF-8 encoding is at most 1,024 bytes long. We recommend using NFSv4.1 where possible and encode file names with Unicode UTF-8 to ensure a successful data import. File names that are not encoded with UTF-8 might not be uploaded to S3 or might be uploaded to S3 with a different file name depending on the NFS encoding you use.
- Ensure that the maximum length of your file path is less than 1024 characters. Snow Family devices do not support file paths that are greater that 1024 characters. Exceeding this file path length will result in file import errors.
- For more information, see Object keys in the Amazon Simple Storage Service User Guide.
- For NFS based transfers, standard POSIX style meta-data will be added to your objects as they get imported to Amazon S3 from Snow Family devices. In addition, you will 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 Snow Family device import with NFS option.

> **Note**
>
> Available storage space on the Snowcone device is not accurate until the NFS service is started.

You can also configure and manage the NFS interface with AWS OpsHub, a GUI tool. For more information, see Using NFS for Offline File Transfer.

NFS configuration for Snow Family devices

The NFS interface is not running on the Snow Family device by default, so you need to start it to enable data transfer to the device. You can configure the NFS interface by providing the IP address of a Virtual Network Interface (VNI) running on the Snow Family device and restricting access to your file share, if required. Before configuring the NFS interface, set up a virtual network interface (VNI) on your Snow Family device. For more information, see Network Configuration for Compute Instances.

Configure Snow Family devices for the NFS interface

- Use the `describe-service` command to determine if the NFS interface is active.

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

The command will return the state of the NFS service, ACTIVE or INACTIVE.

```
{
  "ServiceId" : "nfs",
  "Status" : {
  "State" : "ACTIVE"
  }
}
```

If the value of the State name is ACTIVE, the NFS interface service is active and you can mount the Snow Family device NFS volume. For more information, see

After the NFS interface is started, mount the endpoint as local storage on client computers.

The following are the default mount commands for Windows, Linux, and macOS operating systems.

- Windows:

```
mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/
buckets/BucketName *
```

- Linux:

```
mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
```

- macOS:

```
mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-
interface-ip-address:/buckets/$bucketname mount_point
```

. If the value is `INACTIVE`, you have to start the service.

Starting the NFS service on the Snow Family device

Start a virtual network interface (VNI), if necessary, then start the NFS service on the Snow Family device. If necessary, when starting the NFS service, provide a block of allowed network addresses. If you don't provide any addresses, access to the NFS endpoints will be unrestricted.

1. Use the `describe-virtual-network-interface` command to see the VNIs available on the Snow Family device.

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

   If one or more VNIs are active on the Snow Family device, the command returns the following.

   ```
   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"
   ```

```
    }
  ]
```

Note the value of the `VirtualNetworkInterfaceArn` name of the VNI to use with the NFS interface.

2. If no VNIs are available, use the `create-virtual-network-interface` command to create a VNI for the NFS interface. For more information, see Setting up a Virtual Network Interface (VNI).
3. Use the `start-service` command to start the NFS service and associate it with the VNI. To restrict access to the NFS interface, include the `service-configuration` and `AllowedHosts` parameters in the command.

```
snowballEdge start-service --virtual-network-interface-arns arn-of-vni --service-id
 nfs --service-configuration AllowedHosts=CIDR-address-range
```

4. Use the `describe-service` command to check the service status. It is running when the value of the `State` name is `ACTIVE`.

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

The command returns the service state, as well as the IP address and port number of the NFS endpoint and the CIDR ranges allowed to access the endpoint.

```
{
 "ServiceId" : "nfs",
 "Status" : {
 "State" : "ACTIVE"
 },
 "Endpoints" : [ {
 "Protocol" : "nfs",
 "Port" : 2049,
 "Host" : "192.0.2.0"
 } ],
 "ServiceConfiguration" : {
 "AllowedHosts" : [ "10.24.34.0/23", "198.51.100.0/24" ]
 }
}
```

Mounting NFS endpoints on client computers

After the NFS interface is started, mount the endpoint as local storage on client computers.

The following are the default mount commands for Windows, Linux, and macOS operating systems.

- Windows:

```
mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/
buckets/BucketName *
```

- Linux:

```
mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
```

- macOS:

```
mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-
interface-ip-address:/buckets/$bucketname mount_point
```

Stopping the NFS interface

When you are finished transferring files through the NFS interface and before powering off the Snow Family device, use the `stop-service` command to stop the NFS service.

```
snowballEdge stop-service --service-id nfs
```

Using AWS IoT Greengrass to run pre-installed software on Amazon EC2-compatible 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-compatible 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-compatible instance for AWS IoT Greengrass V2

1. Launch the AWS IoT Greengrass validated AMI with a public IP Address and an SSH key:

   a. Using the AWS CLI: run-instances.

   b. Using AWS OpsHub: Launching an Amazon EC2-compatible instance.

   > **Note**
   >
   > Take note of the public IP address and SSH key name that are associated with the instance.

2. Connect to the EC2-compatible 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-compatible instance. Replace *public-ip-address* with the public IP address of the EC2-compatible 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-compatible instance. To connect to your EC2-compatible 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 Convert your private key using PuTTYgen in the Amazon EC2 User Guide.

Installing AWS IoT Greengrass

Next, you set up your EC2-compatible 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. Use the following commands to provide credentials to allow you to install AWS IoT Greengrass Core software. Replace the example values with your credentials:

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

> **Note**
>
> These are credentials from the IAM user in the AWS Region, not the Snow Family device.

5. Use the following command to install the AWS IoT Greengrass Core software. The command creates AWS resources that the core software requires to operate and sets up the core software as a system service that runs when the AMI boots up.

   Replace the following parameters in the command:

   - `region`: The AWS Region in which to find or create resources.
   - `MyGreengrassCore`: The name of the AWS IoT thing for your AWS IoT Greengrass core device.
   - `MyGreengrassCoreGroup`: The name of the AWS IoT thing group for your AWS IoT Greengrass core device.

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE \
    -jar ./GreengrassInstaller/lib/Greengrass.jar \
    --aws-region region \
    --thing-name MyGreengrassCore \
    --thing-group-name MyGreengrassCoreGroup \
    --thing-policy-name GreengrassV2IoTThingPolicy \
    --tes-role-name GreengrassV2TokenExchangeRole \
    --tes-role-alias-name GreengrassCoreTokenExchangeRoleAlias \
    --component-default-user ggc_user:ggc_group \
    --provision true \
    --setup-system-service true \
    --deploy-dev-tools true
```

Note

This command is for an Amazon EC2-compatible instance running an Amazon Linux 2 AMI. For a Windows AMI, see Install the AWS IoT Greengrass Core software.

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

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
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.

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.

> ⓘ **Note**
>
> Once you return the Snow device for import into Amazon S3, AWS will start ingestion of the data after ensuring the device has not been tampered with and that the device is healthy. In case you do not want the data on the device to be ingested to your destination S3 bucket, you can request to cancel the Snow job. If you cancel the job, we will skip the data transfer and securely erase the device following the established processes. We are not able to hold a device containing your data at our facilities due to our strict chain of custody and operating procedures.

For information about to ship the device, see Shipping Considerations for AWS Snowcone.

Protecting Data on Your Device

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

Topics

Securing Your AWS Snowcone

Following are some security points that we recommend you consider when using Snowcone, in addition to some high-level information on other security precautions that we take when a device arrives at AWS for processing.

We recommend the following security approaches:

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

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

We perform the following security steps:

- When transferring data with the file interface, object metadata is persisted.

- When a device arrives at AWS, we inspect it for any signs of tampering and to verify that no changes were detected by the Trusted Platform Module (TPM). Snowcone uses multiple layers of security designed to protect your data, including tamper-resistant enclosures, 256-bit encryption, and an industry-standard TPM designed to provide both security and full chain of custody for your data.
- After the data transfer job has been processed and verified, AWS performs a software erasure of the Snowcone device following the National Institute of Standards and Technology (NIST) guidelines for media sanitization.

Validating NFC Tags

AWS Snowcone devices have NFC tags built into them. You can scan these tags with the Snowcone Verification App, available on Android. Scanning and validating these NFC tags can help you verify that your device has not been tampered with before you use it.

Validating NFC tags includes using the Snowball Edge client to generate a device-specific QR code to verify that the tags you're scanning are for the right device. For information, see Getting Your QR Code for NFC Validation.

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 a job to order a Snow Family device.
2. Receive the device.
3. Connect to your local network.
4. Get your credentials and tools. For more information, see Getting Credentials.
5. Download and install the Snowball Edge client. For more information, see Using the AWS Snowball Edge Client.

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.

   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.

> **Note**
>
> If we are unable to import data to our data centers from the Snow device due to any issue with access permissions you have configured, we will attempt to notify you and you will have 30 days from the date we provide the notification to resolve the issue. If the issue is not resolved, we may cancel your AWS Snow Family job and delete data from the device.

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.

Job Status	Meaning
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.

Notifications for Snow Family devices

How Snow uses Amazon SNS

The Snow service is designed to take advantage of the robust notifications delivered by Amazon Simple Notification Service (Amazon SNS). While creating a job to order a Snow device, you can provide email addresses to receive notifications for your job status changes. When you do this, you choose an existing SNS topic or create a new one. If the SNS topic is encrypted, you need to enable customer-managed KMS encryption for the topic and set up customer-managed KMS key policy. See Choose your notification preferences.

After you create your job, every email address that you specified to get Amazon SNS notifications receives an email message from AWS notifications asking for confirmation to the topic subscription. A user of the email account must confirm the subscription by choosing **Confirm subscription**. The Amazon SNS notification emails are tailored for each job status, and include a link to the AWS Snow Family Management Console.

You can also configure Amazon SNS to send text messages for status change notifications from the Amazon SNS console. For more information, see Mobile text messaging (SMS) in the *Amazon Simple Notification Service Developer Guide*.

Encrypting SNS topics for Snow job status changes

Enable customer-managed KMS encryption for the SNS topic for Snow job status change notifications. SNS topics encrypted with AWS-managed encryption cannot receive Snow job status changes because the Snow import IAM role does not have access to the AWS-managed KMS key to perform `Decrypt` and `GenerateDataKey` actions. Additionally, policies of AWS-managed KMS keys cannot be edited.

To enable server-side encryption for an SNS topic using the Amazon SNS management console

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Topics**.
3. In the Topics page, choose the topic used for job status change notifications, then choose **Edit**.
4. Expand the **Encryption** section and do the following:

   a. Choose **Enable encryption**.

   b. Specify the AWS KMS key. See

   c. For each KMS type, the description, account, and KMS ARN are displayed.

5. To use a custom key from your AWS account, choose the **AWS KMS key** field and then choose the custom KMS kms from the list. For instructions on creating custom KMS, see [Creating keys](https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html) in the AWS Key Management Service Developer Guide.

   To use a custom KMS ARN from your AWS account or from another AWS account, enter the KMS key ARN in the **AWS KMS key** field.

6. Choose **Save changes**. Server side encryption is enabled for your topic and the topic page is displayed.

Setting up a customer-managed KMS key policy

After enabling encryption for SNS topics that will receive notifications for Snow job status changes, update the KMS policy for the SNS topic encryption and allow the Snow service principal `"importexport.amazonaws.com"` for `"mks:Decrypt"` and `"mks:GenerateDataKey*"` actions.

To allow the import export service role in the KMS key policy

1. Sign in to the AWS Management Console and open the AWS Key Management Service (AWS KMS) console at https://console.aws.amazon.com/kms.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. At the top-right corner of the console, change the AWS Region of the console to the same region as the Snow device was ordered from.
4. In the navigation pane, choose **Customer managed keys**.
5. IN the list of KMS keys, choose the alias or key ID of the KMS key to update.
6. Choose the **Key policy** tab, in the key policy statements, you can see the principals that have been given access to the KMS key by the key policy, and you can see the actions they can perform.
7. For the Snow service principal `"importexport.amazonaws.com"`, add the following policy statement for `"kms:Decrypt"` and `"kms:GenerateDataKey*"` actions:

```
  {

    "Effect": "Allow",
    "Principal": {
    "Service": "service.amazonaws.com"
  },
  "Action": [
  "kms:Decrypt",
  "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
    "ArnLike": {
    "aws:SourceArn": "arn:aws:service:region:customer-account-id:resource-type/customer-resource-id"
  },
  "StringEquals": {
  "kms:EncryptionContext:aws:sns:topicArn": "arn:aws:sns:your_region:customer-account-id:your_sns_topic_name"
  }
  }
  }
```

8. Choose **Save Changes** to apply the changes and exit the policy editor.

SNS notification examples

Amazon SNS notifications produce the following email messages when your job status changes. These messages are examples of the `Email-JSON` SNS topic protocol.

Job status	SNS notification JSON

Job created	`{`
	`"Type" : "Notification",`
	`"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",`
	`"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",`

<table>
<tr><th>Job status</th><th>SNS notification JSON</th></tr>
<tr><td></td><td><pre>
  "Message" : "Your job Job-name
 (JID8bca334a-6c2f-4cd0-97e2
-3f5a4dc9bd6d) has been created. More
 info - https://console.aws.amazon.
com/importexport",
  "Timestamp" : "2023-02-23T00:27:
58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5tlZhJNHLHUXvZ
gtZzlk24FzVa7oX0T4P03neeXw8
ZEXZx6z35j2FOTuNYShn2h0bKNC/
zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZ
F+4uWHEE73yDVR4SyYAikP9jstZzDRm
+bcVs8+T0yaLiEGLrIIIL4esi1llhIkg
ErCuy5btPcWXBdio2fpCRD5x9oR
6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/
iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7
TalMD0lzmJuOrExtnSIbZew3foxgx8GT
+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/
LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://
sns.us-east-1.amazonaws.com/
SimpleNotificationService-010a507c1
833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://
sns.us-east-2.amazonaws.com/?
Action=Unsubscribe&SubscriptionArn
=arn:aws:sns:us-east-2:1111
22223333:ExampleTopic1:e103
9402-24e7-40a3-a0d4-797da162b297"
  }
</pre></td></tr>
</table>

Job status	SNS notification JSON

Preparing appliance	{
	  "Type" : "Notification",
	  "MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
	  "TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
	  "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) is being prepared. More info - https://console.aws.amazon.com/importexport",
	  "Timestamp" : "2023-02-23T00:27:58.831Z",
	  "SignatureVersion" : "1",
	  "Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
	  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
	  "UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
	  }

Job status	SNS notification JSON

Exporting	{
	"Type" : "Notification",
	"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
	"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
	"Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) is being Exported. More info - https://console.aws.amazon.com/importexport",
	"Timestamp" : "2023-02-23T00:27:58.831Z",
	"SignatureVersion" : "1",
	"Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
	"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
	"UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
	}

<table>
<tr><th>Job status</th><th>SNS notification JSON</th></tr>
<tr><td>In transit to you</td><td>

```
{
  "Type" : "Notification",
  "MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
  "TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
  "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) is in transit to you. More info - https://console.aws.amazon.com/importexport",
  "Timestamp" : "2023-02-23T00:27:58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
  }
```

</td></tr>
</table>

Job status	SNS notification JSON

Delivered to you	{
	"Type" : "Notification",
	"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
	"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
	"Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) was delivered to you. More info - https://console.aws.amazon.com/importexport",
	"Timestamp" : "2023-02-23T00:27:58.831Z",
	"SignatureVersion" : "1",
	"Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
	"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
	"UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
	}

Job status	SNS notification JSON

In transit to AWS	{
	"Type" : "Notification",
	"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
	"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
	"Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) is in transit to AWS. More info - https://console.aws.amazon.com/importexport",
	"Timestamp" : "2023-02-23T00:27:58.831Z",
	"SignatureVersion" : "1",
	"Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
	"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
	"UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
	}

<table>
<tr><th>Job status</th><th>SNS notification JSON</th></tr>
<tr><td>At sorting facility</td><td><pre>
  {
  "Type" : "Notification",
  "MessageId" : "dc1e94d9-56c5-5e9
6-808d-cc7f68faa162",
  "TopicArn" : "arn:aws:sns:us-ea
st-2:111122223333:ExampleTopic1",
  "Message" : "Your job Job-name
 (JID8bca334a-6c2f-4cd0-97e2
-3f5a4dc9bd6d) is at AWS sorting
 facility. More info - https://
console.aws.amazon.com/impor
texport",
  "Timestamp" : "2023-02-23T00:27:
58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5tlZhJNHLHUXvZ
gtZzlk24FzVa7oX0T4P03neeXw8
ZEXZx6z35j2FOTuNYShn2h0bKNC/
zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZ
F+4uWHEE73yDVR4SyYAikP9jstZzDRm
+bcVs8+T0yaLiEGLrIIIL4esi1llhIkg
ErCuy5btPcWXBdio2fpCRD5x9oR
6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/
iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7
TalMD0lzmJuOrExtnSIbZew3foxgx8GT
+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/
LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://
sns.us-east-1.amazonaws.com/
SimpleNotificationService-010a507c1
833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://
sns.us-east-2.amazonaws.com/?
Action=Unsubscribe&SubscriptionArn
=arn:aws:sns:us-east-2:1111
22223333:ExampleTopic1:e103
9402-24e7-40a3-a0d4-797da162b297"
  }
</pre></td></tr>
</table>

Job status	SNS notification JSON

At AWS	`{`
	`"Type" : "Notification",`
	`"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",`
	`"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",`
	`"Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) is at AWS. More info - https://console.aws.amazon.com/importexport",`
	`"Timestamp" : "2023-02-23T00:27:58.831Z",`
	`"SignatureVersion" : "1",`
	`"Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",`
	`"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",`
	`"UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"`
	`}`

<table>
<tr><th>Job status</th><th>SNS notification JSON</th></tr>
<tr><td>Importing</td><td>

```
  {
  "Type" : "Notification",
  "MessageId" : "dc1e94d9-56c5-5e9
6-808d-cc7f68faa162",
  "TopicArn" : "arn:aws:sns:us-ea
st-2:111122223333:ExampleTopic1",
  "Message" : "Your job Job-name
 (JID8bca334a-6c2f-4cd0-97e2
-3f5a4dc9bd6d) is being imported.
 More info - https://console.aw
s.amazon.com/importexport",
  "Timestamp" : "2023-02-23T00:27:
58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5tlZhJNHLHUXvZ
gtZzlk24FzVa7oX0T4P03neeXw8
ZEXZx6z35j2FOTuNYShn2h0bKNC/
zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZ
F+4uWHEE73yDVR4SyYAikP9jstZzDRm
+bcVs8+T0yaLiEGLrIIIL4esi1llhIkg
ErCuy5btPcWXBdio2fpCRD5x9oR
6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/
iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7
TalMD0lzmJuOrExtnSIbZew3foxgx8GT
+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/
LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://
sns.us-east-1.amazonaws.com/
SimpleNotificationService-010a507c1
833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://
sns.us-east-2.amazonaws.com/?
Action=Unsubscribe&SubscriptionArn
=arn:aws:sns:us-east-2:1111
22223333:ExampleTopic1:e103
9402-24e7-40a3-a0d4-797da162b297"
  }
```

</td></tr>
</table>

Job status	SNS notification JSON

Completed	{
	"Type" : "Notification",
	"MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
	"TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
	"Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) complete.\nThanks for using AWS Snow Family.\nCan you take a quick survey on your experience? Survey here: http://bit.ly/1pLQJMY. More info - https://console.aws.amazon.com/importexport",
	"Timestamp" : "2023-02-23T00:27:58.831Z",
	"SignatureVersion" : "1",
	"Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
	"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
	"UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
	}

Job status	SNS notification JSON

<table>
<tr><th>Job status</th><th>SNS notification JSON</th></tr>
<tr><td>Cancelled</td><td>

```
{
  "Type" : "Notification",
  "MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",
  "TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",
  "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) was canceled. More info - https://console.aws.amazon.com/importexport",
  "Timestamp" : "2023-02-23T00:27:58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5tlZhJNHLHUXvZgtZzlk24FzVa7oX0T4P03neeXw8ZEXZx6z35j2FOTuNYShn2h0bKNC/zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZF+4uWHEE73yDVR4SyYAikP9jstZzDRm+bcVs8+T0yaLiEGLrIIIL4esi1llhIkgErCuy5btPcWXBdio2fpCRD5x9oR6gmE/rd5O7lX1c1uvnv4r1Lkk4pqP2/iUfxFZva1xLSRvgyfm6D9hNklVyPfy+7TalMD0lzmJuOrExtnSIbZew3foxgx8GT+lbZkLd0ZdtdRJlIyPRP44eyq78sU0Eo/LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-010a507c1833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://sns.us-east-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-2:111122223333:ExampleTopic1:e1039402-24e7-40a3-a0d4-797da162b297"
  }
```

</td></tr>
</table>

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.

Returning a Snowcone Device

The prepaid shipping information on the E Ink display contains the correct address to return the device. For information about how to return your Snowcone device, see Shipping Carriers. 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.

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
- JMAPI Required Strings
- JMAPI Endpoints
- JMAPI CLI Commands
- Examples

Common Uses of JMAPI

- Automating ordering of Snowcone devices
- Downloading the manifest file
- Downloading the unlock file
- Listing out the current Snowcone jobs
- Downloading the Snowcone job completion report

JMAPI Required Strings

When placing an order through the job management API, you use the following required parameters, which are shown with examples.

```
--job-type
--resources
```

```
--address-id
--region
--role-arn
--kms-key-arn
--shipping-option
--device-type
--description
```

JMAPI Endpoints

API Endpoint

To make calls to each endpoint, the format is `snowballEdge.`*`region`*`.amazonaws.com`. Following are some examples to help you understand the breakdown of the endpoint.

Example

Region	Endpoint
US East (N. Virginia)	snowball.us-east-1.amazonaws.com
US West (Oregon)	snowball.us-west-2.amazonaws.com

JMAPI CLI Commands

Job Management CLI

The following are the CLI calls that you can make against the job management API.

Command	Example
Listing Jobs	`aws snowball list-jobs`
Describe Job	`aws snowball describe-job --job-id` *`[JOB ID]`*
Describe Address	`aws snowball describe-address --address-id`

Command	Example
Create Address	`aws snowball create-address --cli-input-json file://create-address.json`
Create Job	`aws snowball create-job --cli-input-json file://create-job.json`
Cancel Job	`aws snowball cancel-job --job-id` *[JOB ID]*

Examples

The following are examples of commands using the job management API.

KMS JSON Example

The following JSON example is a properly formatted JSON file for using the AWS KMS policy file.

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333"
    }
}
```

Create Address Example

The following examples show you how you would format the command to create your address and what the response is when it is successful.

```
aws snowball create-address --address "Name=Bob,Company=AWS,Street1=1234 Fake
 St.,City=All,StateOrProvince=Any,Country=US,PostalCode=12345,PhoneNumber=1234567890"
```

Example Output

```
{
    "AddressId": "ADID3be640c8-1111-1111-1111-917f201ffa42"
}
```

Create Job Example

The following command shows you an example command for running the `create-job` command.

```
aws snowball create-job --job-type 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-compatible compute instances on an AWS Snowcone device.

Topics

- Configuring an AMI to Use SSH to Connect to Compute Instances Launched on the Device
- Creating Your Job Using the Console
- Creating Your Job Using the AWS CLI

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 to order a Snow Family device. 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 a job to order a Snow Family device. 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 20 total AMIs to your job, depending on device type.
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.

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

When you create a job to order a Snow Family device, 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

Snow Family devices can only be returned to the same AWS Region where the devices were ordered. Some AWS Regions support sending Snow Family devices to a different country than from which the device was ordered. See Region-Based Shipping Restrictions for more information.

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.

   If you've powered off and unplugged your Snowcone device and the shipping information doesn't appear on the E Ink screen after about a minute, see Troubleshooting problems returning Snow Family devices.

Region-Based Shipping Restrictions

Before you create a job to order a Snow Family device, you should sign in to the console from the AWS Region that your data resides. Snow Family devices are not shipped between international countries—for example, from Asia Pacific (India) to Asia Pacific (Australia).

An exception to shipping between countries is among European Union (EU) member countries. For data transfers in the European AWS Regions, we only ship 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 Snow Family devices to post office boxes.

Shipping an AWS Snowcone Device

The prepaid shipping information on the E Ink display 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”. 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.

If you've powered off and unplugged your Snowcone device and the shipping information doesn't appear on the E Ink screen after about a minute, see Troubleshooting problems returning Snow Family devices.

Shipping Carriers

When you create a job to order a Snow Family device, you provide the address where you want the AWS Snowcone device shipped. 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 South Korea, Japan, Australia, Indonesia, Israel, and Singapore, 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 Canada, the EU, South Africa, and the US

In Canada, the EU, South Africa, and the US, keep the following information in mind for UPS to pick up an AWS Snowcone device:

- 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 provides a tracking number for your shipment.

UPS services for Snow Family devices is domestic only within a country.

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 Israel

In Israel, arrange pick up by contacting AWS by email at snowball-shipping@amazon.com. Enter *Snowcone Pickup Request* in the subject and include this information:

- Job ID – The job ID associated with the Snow device you are returning. You can find the job ID in the AWS Snow Family Management Console.
- Pickup address – The address where the device will be picked up.
- Pickup date – The soonest date you would like the device to be picked up.

- Point of contact details – The name, email address, and local phone number that the shipping service, Kuehne + Nagel, can use if necessary for information about the pickup.

Soon, you will get a follow-up email from AWS Support with information regarding the pickup of your device at the address you provided. Prepare the device for shipment and be ready for pickup, usually before 12:00 local time. See Preparing an AWS Snowcone Device for Shipping.

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.

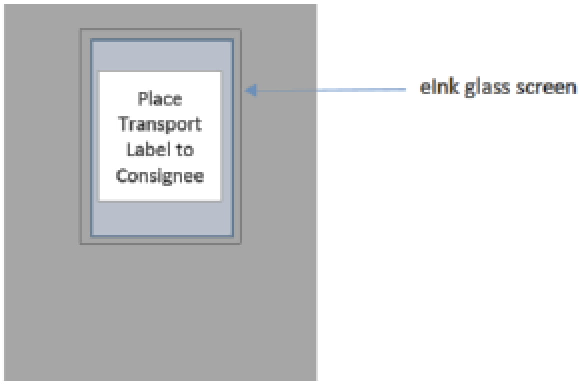
> ⚠ **Important**
>
> Unless personally instructed otherwise by AWS, never affix a separate shipping label to the AWS Snowcone device . Always use the shipping label that is displayed on the device's E Ink display.

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.

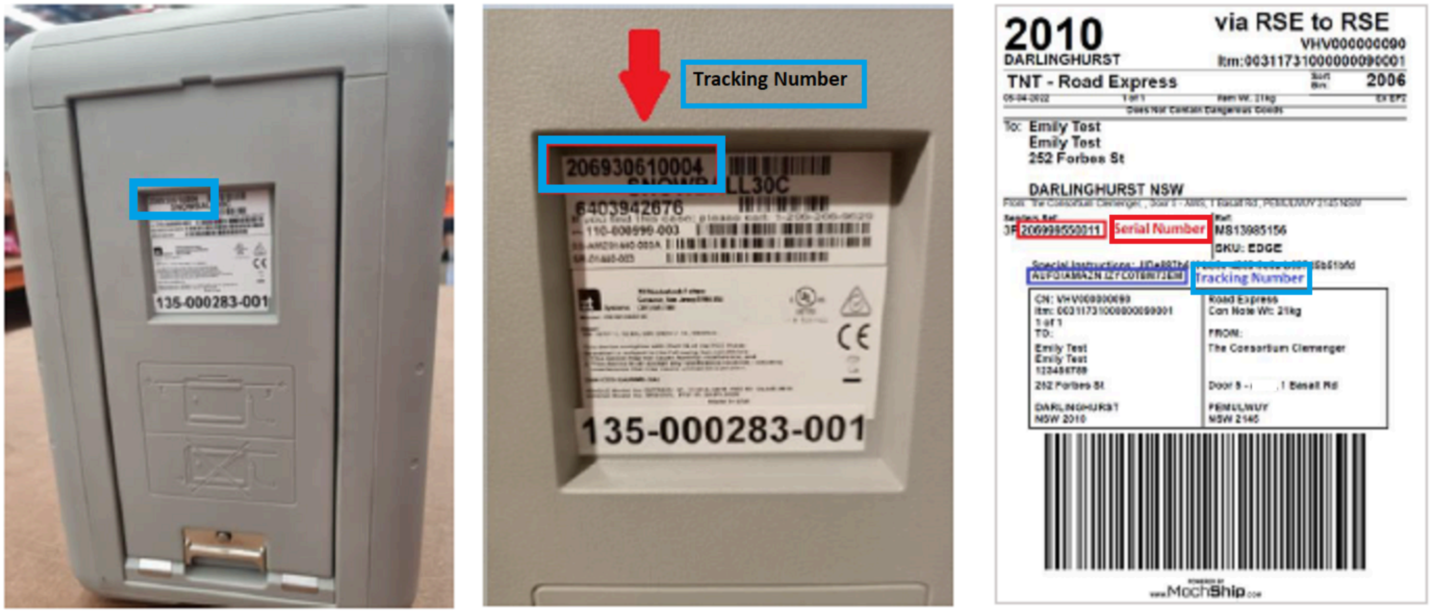
AWS Snowcone Pickups in Australia

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 Snowcone Pickups in South Korea

In South Korea, Kuehne + Nagel handles your pickups. When you are ready to return your device, send an email to snowball-shipping@amazon.com with *Snowcone Pickup Request* in the subject line so we can schedule the pickup for you. In the body of the email, include the following information:

- Job ID – The job ID associated with the Snowcone that you want returned to AWS.
- Pickup address – The address where the device will be picked up.
- Pickup date – The soonest day you would like the device picked up.
- Point of contact details – the name, email address, and local phone number that Kuehne + Nagel can use to get in touch with you if needed.

Soon, you will get a follow-up email from the AWS Support with information regarding the pickup of your device at the address you provided. Prepare the device for shipment (see Preparing an AWS Snowcone Device for Shipping) and be ready for pickup usually between 1300 and 1500 hours local time.

AWS Snowcone Pickups in Hong Kong

In Hong Kong, S.F. Express handles your pickups. When you are ready to return your device, send an email to snowball-shipping-ap-east-1@amazon.com with *Snowcone Pickup Request* in the subject line so we can schedule the pickup for you. In the body of the email, include the following information:

- Job ID
- AWS account ID
- Contact name
- Contact phone number
- Contact email address
- The day you want the device picked up.

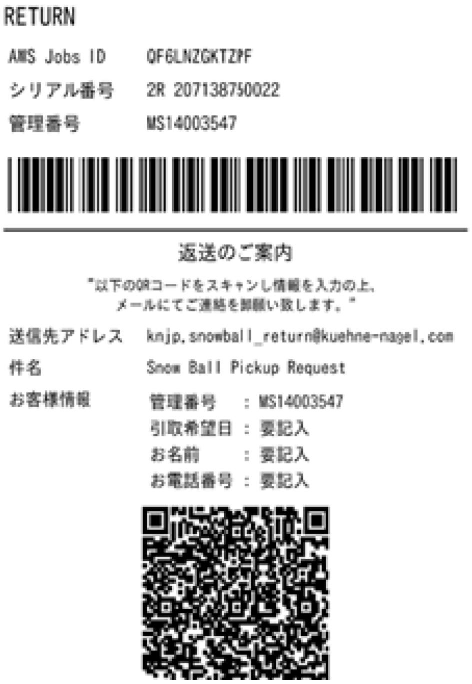
- Earliest preferred pickup time
- Latest preferred pickup time
- Pickup address

> **Note**
>
> Once you arrange a pickup date with S.F. Express, it can't be rescheduled.

The device will be delivered to AWS by S.F. Express. Use the S.F. Express tracking number of the return shipment to learn when the delivery was completed.

AWS Snowcone Pickups in Indonesia, Japan, and Singapore

In Indonesia, Japan, and Singapore, 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.



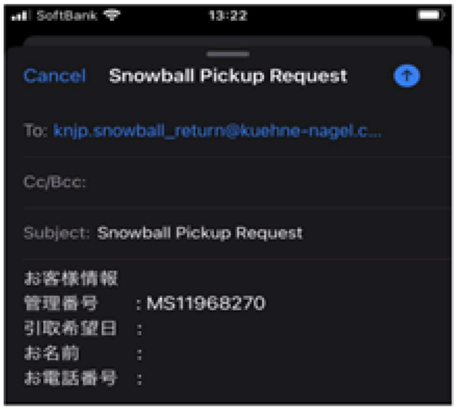
RETURN

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御願い致します。"

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



To: knjp.snowball_return@kuehne-nagel.c...

Cc/Bcc:

Subject: Snowball Pickup Request

お客様情報
管理番号　　：MS11968270
引取希望日　：
お名前　　　：
お電話番号　：

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:

- **Australia, Indonesia, Japan, Singapore, South Korea –** When shipping within these countries, you have access to the standard shipping speed of 1 to 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.
- **European Union (EU) –** When shipping to any of the countries within the EU, you have access to express shipping. Typically, Snowcone 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.
- **India –** When shipping within India, AWS Snowcone devices are sent out within 7 working days of AWS receiving all related tax documents.
- **Israel –** When shipping in Israel, you have access to same-day shipping.
- **United States of America (US) and Canada –** When shipping in the US and Canada, you have access to one-day shipping and two-day shipping.
- **United Kingdom (UK) –** When shipping within the UK, you have access to express shipping. Typically, Snowcone 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.

Updating software on Snowcone devices

AWS will notify you when new software is available for Snow Family devices you have. The notification is provided through email, AWS Health Dashboard, and as a CloudWatch event. The email notification is sent from Amazon Web Services, Inc. to the email address attached to the AWS account used to order the Snow Family device. When you receive the notification, follow the instructions in this topic and download and install the update as soon as possible to avoid interruption of your use of the device. For more information about AWS Health Dashboard, see AWS Health User Guide. For more information about CloudWatch Events, see Amazon CloudWatch Events User Guide.

You can download software updates from AWS and install them on Snowcone devices in your on-premises environments. These updates happen in the background. You can continue to use your devices as normal while the latest software is downloaded securely from AWS to your device. However, to apply downloaded updates, you must stop services running on the device and restart it after the update is complete.

Software updates provided by AWS for Snowball Edge/Snowcone devices (Appliances) are Appliance Software as per Section 9 of the Service Terms.

The software updates are provided solely for the purpose of installing the software updates on the applicable Appliance on behalf of AWS. You will not (or attempt to), and will not permit or authorize third parties to (or attempt to) (i) make any copies of the software updates other than those necessary to install the software updates on the applicable Appliance, or (ii) circumvent or disable any features or measures in the software updates, including, but not limited to, any encryption applied to the software update. Once the software updates have been installed on the applicable Appliance, you agree to delete the software updates from any and all media utilized in installing the software updates to the Appliance.

> ⚠ **Warning**
>
> We highly recommend that you suspend all activity on your device before installing the update. Updating the device and restarting will stop running instances and interrupt any writes to local Amazon S3 buckets.

Topics

- Downloading updates
- Installing updates
- Updating the SSL certificate
- Updating your Amazon Linux 2 AMIs on Snow Family devices

Prerequisites

Before you can update your device, the following prerequisites must be met:

- You've created your job, have the device on-premises, and you've unlocked it. For more information, see Getting Started.
- Updating Snowcone devices is done through the Snowball Edge client. The latest version of the Snowball Edge client must be downloaded and installed on a computer in your local environment that has a network connection to the device you want to update. For more information, see Using the AWS Snowball Edge Client.
- (Optional) We recommend that you configure a profile for the Snowball Edge client. For more information, see Configuring a Profile for the Snowball Edge Client.

After you complete these tasks, you can download and install updates for Snowcone devices.

Downloading updates

There are two primary ways that you can download an update for Snow Family devices:

- You can trigger manual updates at any time using specific Snowball Edge client commands.
- You can programmatically determine a time to automatically update the device.

The following procedure outlines the process of manually downloading updates. For information about automatically updating your Snowcone device, see `configure-auto-update-strategy` in Updating a Snowcone.

Note

If your device has no access to the internet, you can download an update file using the GetSoftwareUpdates API. Then point to a local file location when you call `download-updates` using the `uri` parameter, as in the following example.

```
snowballEdge download-updates --uri file:///tmp/local-update
```

For Windows operating systems, format the value of the `uri` parameter as follows:

```
snowballEdge download-updates --uri file:/C:/path/to/local-update
```

To check for and download Snowcone software updates

1. Open a terminal window, and ensure that the Snowcone device is unlocked using the `describe-device` command. If the device is locked, use the `unlock-device` command to unlock it. For more information, see Unlocking the Snow Family device
2. When the device is unlocked, run the `snowballEdge check-for-updates` command. This command returns the latest available version of the Snowball Edge software, and also the current version installed on the device.
3. If your device software is out of date, run the `snowballEdge download-updates` command.

   > **Note**
   >
   > If your device is not connected to the internet, first download an update file using the GetSoftwareUpdates API. Then run the `snowballEdge download-updates` command using the `uri` parameter with a local path to the file that you downloaded, as in the following example.
   >
   > ```
   > snowballEdge download-updates --uri file:///tmp/local-update
   > ```
   >
   > For Windows operating systems, format the value of the `uri` parameter as follows:
   >
   > ```
   > snowballEdge download-updates --uri file:/C:/path/to/local-update
   > ```

4. You can check the status of this download with the `snowballEdge describe-device-software` command. While an update is downloading, you display the status using this command.

Example output of `describe-device-software` command

```
Install State: Downloading
```

Installing updates

After downloading updates, you must install them and restart your device for the updates to take effect. The following procedure guides you through manually installing updates.

> **Note**
>
> Suspend all activity on the device before you install software updates. Installing updates stops running instances and interrupts any writes to Amazon S3 buckets on the device. This can result in lost data

To install software updates that were already downloaded to standalone Snow Family devices

1. Open a terminal window, and ensure that the Snowcone device is unlocked using the `describe-device` command. If the device is locked, use the `unlock-device` command to unlock it. For more information, see Unlocking a Snow Family device.
2. Run the `list-services` command to see the services available on the device. The command returns the service IDs of each service available on the device.

   ```
   snowballEdge list-services
   ```

   Example of output of `list-services` command

   ```
   {
     "ServiceIds" : [ "greengrass", "fileinterface", "s3", "ec2", "s3-snow" ]
   }
   ```

3. For each service ID identified by the `list-services` command, run the `describe-service` command to see the status. Use this information to identify services to stop.

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

   Example of output of `describe-service` command

   ```
   {
   "ServiceId" : "s3",
     "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"
     }
    } ]
   }
   ```

   This output shows that the `s3` service is active and must be stopped using the `stop-service` command.

4. Use the `stop-service` command to stop each service where the value of the `State` name is `ACTIVE` in the output of the `list-services` command. If more than one service is running, stop each one before continuing.

Note

The Amazon S3 adapter, Amazon EC2, AWS STS, and IAM services cannot be stopped. If Amazon S3 compatible storage on Snow Family devices is running, stop it before installing updates. Amazon S3 compatible storage on Snow Family devices has `s3-snow` as the `serviceId`.

```
snowballEdge stop-service --service-id service-id --device-ip-addresses snow-
device-1-ip-address snow-device-device-2-ip-address snow-device-3-ip-address --
manifest-file path/to/manifest/file.bin --unlock-code 29-character-unlock-code --
endpoint https://snow-device-ip-address
```

Example of output of the `stop-service` command

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

5. Run the `snowballEdge install-updates` command.
6. You can check the status of this installation with the `snowballEdge describe-device-software` command. While an update is installing, you display the status with this command.

   Example output

   ```
   Install State: Installing //Possible values[NA, Installing, Requires
   Reboot]
   ```

   You've successfully installed a software update for your Snowcone device. Installing an update does not automatically apply the update to the device. To finish installing the update, the device must be restarted.

> ⚠ **Warning**
>
> Restarting your Snow Family device without stopping all activity on the device can result in lost data.

7. When all the services on the device have stopped, reboot the device, unlock the device, and reboot it again. This completes installation of the downloaded software updates. For more information about rebooting the device, see Rebooting the Snow Family device. For more information about unlocking the device, see Unlocking the Snow Family device.
8. When the device powers on after the second reboot, unlock the device.
9. Run the `check-for-updates` command. This command returns the latest available version of the Snowcone software, and also the current version that is installed on the device.

You have now successfully updated the Snow Family device or cluster of devices and confirmed that the update to the latest Snow Family software.

Updating the SSL certificate

If you plan to keep your Snow Family device for more than 360 days, you will need to update the Secure Sockets Layer (SSL) certificate on the device to avoid interruption of your use of the device. If the certificate expires, you will not be able to use the device and will have to return it to AWS.

AWS will notify you 30 days before the SSL certificate expires for Snow Family devices you have. The notification is provided through email, AWS Health Dashboard, and as a CloudWatch event. The email notification is sent from Amazon Web Services, Inc. to the email address attached to the AWS account used to order the Snow Family device. When you receive the notification, follow the instructions in this topic and request an update as soon as possible to avoid interruption of your use of the device. For more information about AWS Health Dashboard, see AWS Health User Guide. For more information about CloudWatch Events, see Amazon CloudWatch Events User Guide.

This topic explains how to determine when the certificate will expire and how to update your device.

1. Use the `snowballEdge describe-device-software` command to determine when the certificate will expire. In the output of the command, the value of `CertificateExpiry` includes the date and time at which the certificate will expire.

Example of `describe-device-software` output

```
Installed version: 101
Installing version: 102
Install State: Downloading
CertificateExpiry : Thur Jan 01 00:00:00 UTC 1970
```

2. Contact AWS Support and request an SSL certificate update.
3. AWS Support will provide an update file. Download and install the update file.
4. Use the new unlock code and manifest file when Unlocking a device.

Updating your Amazon Linux 2 AMIs on Snow Family devices

As a best-practice for security, keep your Amazon Linux 2 AMIs up-to-date on Snow Family devices. Regularly check the Amazon Linux 2 AMI (HVM), SSD Volume Type (64-bit x86) in the AWS Marketplace for updates. When you identify the need to update your AMI, import the latest Amazon Linux 2 image to the Snow device. See Importing an Image into Your Device as an Amazon EC2-compatible AMI.

You can also get the latest Amazon Linux 2 image ID using the `ssm get-parameters` command in the AWS CLI.

```
  aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2 --query 'Parameters[0].[Value]' --region your-region
```

The command returns the latest image ID of the AMI. For example:

```
   ami-0ccb473bada910e74
```

Best Practices for the AWS Snowcone Device

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

Topics

Security

- If you notice anything that looks suspicious about the AWS Snowcone device, don't connect it to your internal network. Instead, contact AWS Support, and a new AWS Snowcone device will be shipped to you.
- We recommend that you don't save a copy of the unlock code in the same location in the workstation as the manifest for that job. Saving these separately helps prevent unauthorized parties from gaining access to the AWS Snowcone device. For example, you can save a copy of the manifest to your local server, and email the code to a user that unlocks the device. This approach limits access to the AWS Snowcone device to individuals who have access to files saved on the server and also that user's email address.
- The credentials displayed when you run the Snowball Edge client command `snowballEdge list-access-keys` followed by `snowballEdge get-secret-access-key` are a pair of keys: an access key and a secret key. These keys are only associated with the job and the local resources on the device. They don't map to your AWS account or any other AWS account. If you try to use these keys to access services and resources in the AWS Cloud, they fail, because they work only for the local resources associated with your job.
- You can restrict access to NFS shares. For details, see Restricting Access to NFS Shares When NFS is Running.
- 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.

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-compatible Instances

To avoid accidentally deleting the Amazon EC2-compatible 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-compatible 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-compatible 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-compatible instances. Using Amazon EBS volumes allows the local Amazon EC2 instances to access more local capacity than the root volume alone. Because this is local storage only, data written to the Amazon EBS volumes is lost when the device is returned to AWS because it can't be imported into Amazon S3.

Topics

Compute resources quotas

Available Compute Resources for Snowcone Devices Quotas for Storage

The following table outlines the available compute resources for Snowcone devices and their quotas for storage.

Instance type	vCPU cores	Memory (GiB)	Quota
`snc1.micro`	1	1	2

Instance type	vCPU cores	Memory (GiB)	Quota
`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-compatible instance capacity quotas for a Snowcone device.

Amazon EC2	Size
Amazon EC2-compatible AMIs	125 GB (the combined size of all the AMIs used in a customer's job)
Amazon EC2-compatible 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.

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 doors on the front and back, don't open the Snowcone device for any reason.
- The device must not be physically damaged. To prevent damage, close the two 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

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-compatible 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-compatible 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.

Troubleshooting problems returning Snow Family devices

Sometimes, after turning off the Snow Family device in preparation to return it, the return shipping information does not appear on the E Ink display.

Action to take

1. Log in to the AWS Snow Family Management Console.
2. View the job for the device.
3. In the **Details** section, in the **Return shipping label** section, choose **To view and print your return label follow this link**.
4. Print the label and attach it to the device.
5. Return the device according to Returning a Snowcone Device.

Job Management API Reference

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:**August 25, 2023

Change	Description	Date
Include custom AMIs when ordering devices	Custom Amazon Machine Images can now be preloaded while ordering AWS Snow Family jobs. For more information, see Using Amazon EC2 on Snowcone.	November 15, 2023
New AWS Region supported	AWS Snowcone SSD devices are now available in Israel (Tel Aviv) AWS Region. Pickup information for this region was added. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference* . For information on shipping, see Shipping Considerations for AWS Snowcone.	August 25, 2023
New AWS Region supported	Snowcone devices are now available in Europe (Paris) AWS Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference* . For information on shipping,	June 29, 2022

	see Shipping Considerations for AWS Snowcone.	
New Snowcone troubleshooting tips	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	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 *AWS General Reference*.	February 23, 2022
New AWS Region supported	AWS Snowcone is now available in the Europe (London) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	January 5, 2022
Support for Network Time Protocol (NTP) server configuration	Snowcone devices now support external Network Time Protocol (NTP) server configuration.	November 16, 2021

New AWS Region supported	AWS Snowcone SSD is now available in the US West (N. California), US East (Ohio), Asia Pacific (Singapore), Asia Pacific (Tokyo), and Asia Pacific (Sydney) Regions. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	November 3, 2021
New AWS Region supported	AWS Snowcone is now available in the US West (N. California), US East (Ohio) and South America (São Paulo) Regions. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	September 29, 2021
New AWS Region supported	AWS Snowcone is now available in the Asia Pacific (Singapore) and Asia Pacific (Tokyo) Regions. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	August 26, 2021
Support for offline data export from Amazon S3 using Snowcone devices	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	Snow Device Management allows you to manage your AWS Snowcone device and local AWS services remotely. All Snowcone devices support Snow Device Managemen t, and it comes preinstalled on new devices in most AWS Regions where Snowcone is available. For more informati on, see Using AWS Snow Device Management to Manage Devices.	August 4, 2021
New AWS Region supported	AWS Snowcone is now available in the Canada (Central) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	April 28, 2021
New AWS Region supported	AWS Snowcone is now available in the Asia Pacific (Sydney) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	March 24, 2021
Support for direct network interface	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 informati on, see Network Configura tion for Compute Instances.	January 12, 2021

New AWS Region supported	AWS Snowcone is now available in the Europe (Frankfurt) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	November 18, 2020
New AWS Region supported	AWS Snowcone is now available in the Europe (Ireland) Region. For more information, see AWS Snow Family endpoints and quotas in the *AWS General Reference*.	September 16, 2020
Introducing AWS Snowcone	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 Glossary Reference*.