AWS DataSync

User Guide

AWS DataSync: User Guide

Table of Contents

What is AWS DataSync?

AWS DataSync is an online data transfer service that simplifies, automates, and accelerates moving data between storage systems and services.

DataSync can copy data to and from:

- Network File System (NFS) file servers
- Server Message Block (SMB) file servers
- Hadoop Distributed File System (HDFS)
- Object storage systems
- Amazon Simple Storage Service (Amazon S3) buckets
- Amazon EFS file systems
- Amazon FSx for Windows File Server file systems
- Amazon FSx for Lustre file systems
- Amazon FSx for OpenZFS file systems
- AWS Snowcone devices

Use cases

These are some of the main use cases for DataSync:

- **Data migration** – Move active datasets rapidly over the network into Amazon S3, Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS. DataSync includes automatic encryption and data integrity validation to help make sure that your data arrives securely, intact, and ready to use.
- **Archiving cold data** – Move cold data stored in on-premises storage directly to durable and secure long-term storage classes such as S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive. Doing so can free up on-premises storage capacity and shut down legacy systems.
- **Data protection** – Move data into any Amazon S3 storage class, choosing the most cost-effective storage class for your needs. You can also send data to Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS for a standby file system.
- **Data movement for timely in-cloud processing** – Move data in or out of AWS for processing. This approach can speed up critical hybrid cloud workflows across many industries. These include machine learning in the life-sciences industry, video production in media and entertainment, big-data analytics in financial services, and seismic research in the oil and gas industry.

Benefits

By using AWS DataSync, you can get the following benefits:

- **Simplify and automate data movement** – DataSync makes it easier to move data over the network between storage systems and services. DataSync automates both the management of data-transfer processes and the infrastructure required for high performance and secure data transfer.
- **Transfer data securely** – DataSync provides end-to-end security, including encryption and integrity validation, to help ensure that your data arrives securely, intact, and ready to use. DataSync accesses your AWS storage through built-in AWS security mechanisms, such as AWS Identity and Access Management (IAM) roles. It also supports virtual private cloud (VPC) endpoints, giving you the option

to transfer data without traversing the public internet and further increasing the security of data copied online.

- **Move data faster** – Transfer data rapidly over the network into AWS. DataSync uses a purpose-built network protocol and a parallel, multi-threaded architecture to accelerate your transfers. This approach speeds up migrations, recurring data-processing workflows for analytics and machine learning, and data-protection processes.
- **Reduce operational costs** – Move data cost-effectively with the flat, per-gigabyte pricing of DataSync. Save on script development and deployment and maintenance costs. Avoid the need for costly commercial transfer tools.

Additional resources

We recommend that you read the following:

- DataSync resources – Includes blogs, videos, and other training materials
- AWS re:Post – See the latest discussion around DataSync
- AWS DataSync pricing – DataSync pricing information

DataSync also supports Terraform. To learn more about DataSync deployment automation with Terraform, see the Terraform documentation.

How AWS DataSync works

Get an overview of how DataSync works.

DataSync architecture

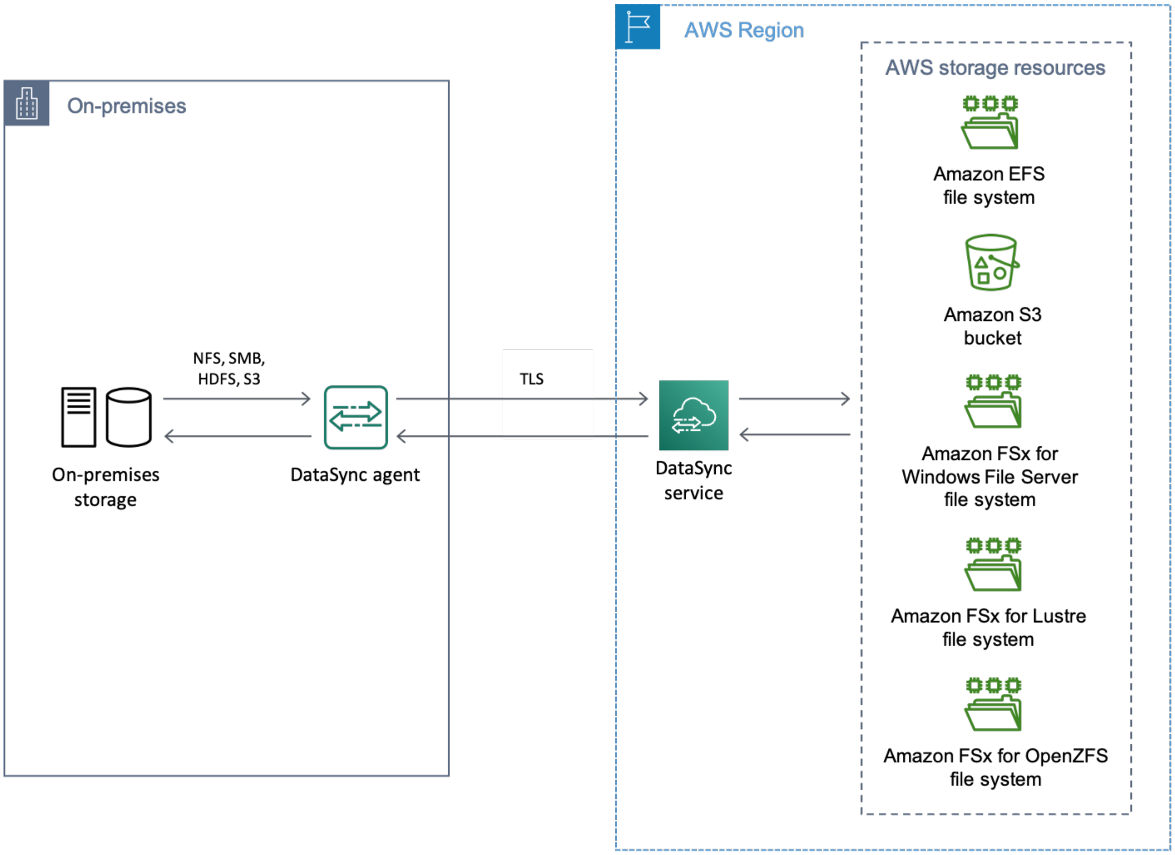
Topics

The following diagrams show how and where DataSync commonly transfers storage data.

For a full list of DataSync supported storage systems and services, see Working with AWS DataSync locations (p. 72).

Transferring between on-premises storage and AWS

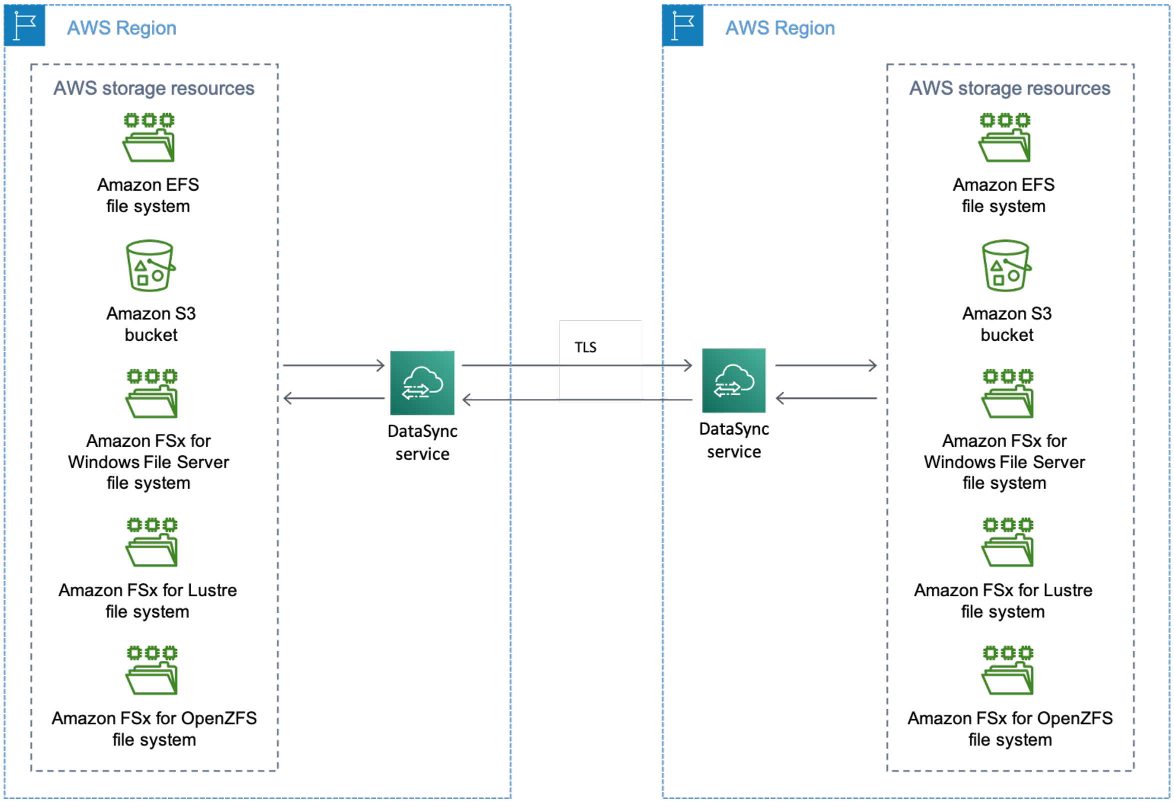
The following diagram shows a high-level view of the DataSync architecture for transferring files between self-managed, on-premises storage systems and AWS services.



Transferring between AWS storage services

The following diagram provides a high-level view of the DataSync architecture for transferring files between AWS services within the same AWS account. This architecture applies to transfers in the same AWS Region and across Regions.

With these kinds of transfers, traffic remains in the AWS network and doesn't traverse the public internet.


AWS Region
AWS storage resources
Amazon EFS file system
Amazon S3 bucket
Amazon FSx for Windows File Server file system
Amazon FSx for Lustre file system
Amazon FSx for OpenZFS file system
DataSync service
TLS
DataSync service
AWS Region
AWS storage resources
Amazon EFS file system
Amazon S3 bucket
Amazon FSx for Windows File Server file system
Amazon FSx for Lustre file system
Amazon FSx for OpenZFS file system

> **Important**
> When you use DataSync to copy files or objects between AWS Regions, you pay for data transfer between Regions. This is billed as data transfer OUT from your source Region to your destination Region. For more information, see Data transfer pricing.

Transferring between cloud storage systems and AWS storage services

With DataSync, you can transfer data between cloud storage systems and AWS services. In this context, cloud storage systems can include:

- Self-managed storage systems hosted by AWS (for example, an NFS share in your virtual private cloud within AWS).
- Storage systems or services hosted by another cloud provider.

For more information, see the following topics:

- Deploying your DataSync agent in AWS Regions (p. 59)
- Tutorial: Transferring data from Google Cloud Storage to Amazon S3 (p. 158)

Components and terminology

Familiarize yourself with DataSync features and concepts.

Agent

An *agent* is a VM that you own that's used to read or write data from storage systems. The agent can be deployed on VMware ESXi, Linux Kernel-based Virtual Machine (KVM), Microsoft Hyper-V hypervisors, or it can be launched as an Amazon EC2 instance. You use the DataSync console, AWS CLI, or DataSync API to set up and activate your agent. The activation process associates your agent VM with your AWS account. For information about agents, see Working with agents (p. 55).

Location

A *location* identifies where you're copying data from or to. Each DataSync transfer (also known as a *task*) has a source and destination location.

For a list of supported locations, see Working with AWS DataSync locations (p. 72).

Task

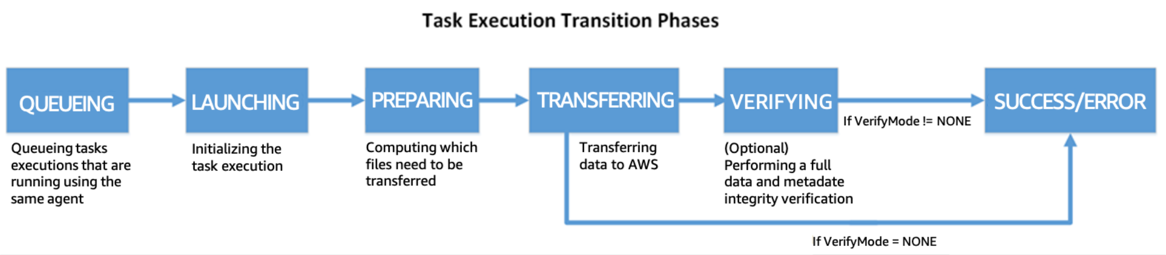
A *task* describes a DataSync transfer. It identifies a source and destination location along with details about how to copy data between those locations. You also can specify how a task treats metadata, deleted files, and permissions.

Task execution

A *task execution* is an individual run of a task, which shows information such as the start time, end time, number of transferred files, and status.

A task execution has five transition phases and two terminal statuses, as shown in the following diagram. These phases and statuses are:

- **QUEUEING** – This phase consists of queuing the task executions that are running using the same agent.
- **LAUNCHING** – During this phase, the task execution is initialized.
- **PREPARING** – During this phase, DataSync computes which files need to be transferred.
- **TRANSFERRING** – During this phase, DataSync transfers data to AWS.
- **VERIFYING** – During this optional phase, DataSync performs a full data and metadata integrity verification. This phase occurs only if the `VerifyMode` option is enabled during configuration.
- **SUCCESS** or **ERROR** – When the task is finished, DataSync sets the task to one of these terminal statuses, depending on whether it was successful.


Task Execution Transition Phases
QUEUEING
Queueing tasks executions that are running using the same agent
LAUNCHING
Initializing the task execution
PREPARING
Computing which files need to be transferred
TRANSFERRING
Transferring data to AWS
VERIFYING
(Optional) Performing a full data and metadate integrity verification
If VerifyMode != NONE
SUCCESS/ERROR
If VerifyMode = NONE

If the `VerifyMode` option isn't enabled in the task configuration, the terminal status is set after the **TRANSFERRING** phase. Otherwise, it is set after the **VERIFYING** phase. The two terminal statuses are these:

- **SUCCESS**
- **ERROR**

For more information, see Task execution statuses (p. 110).

How DataSync transfers files

Topics

When a task starts, it goes through different phases: **LAUNCHING**, **PREPARING**, **TRANSFERRING**, and **VERIFYING**. In the **LAUNCHING** phase, DataSync initializes the task execution. In the **PREPARING** phase, DataSync examines the source and destination file systems to determine which files to sync. It does so by recursively scanning the contents and metadata of files on the source and destination file systems for differences.

The time that DataSync spends in the **PREPARING** phase depends on the number of files in both the source and destination file systems. It also depends on the performance of these file systems. The **PREPARING** phase can therefore take a few minutes to a few hours. For more information, see Starting your DataSync task (p. 109).

After the scanning is done and the differences are calculated, DataSync transitions to the **TRANSFERRING** phase. At this point, DataSync starts transferring files and metadata from the source file system to the destination. DataSync copies changes to files with contents or metadata that are different between the source and the destination. You can narrow down the copied files by filtering the data or by configuring DataSync to not overwrite files that are already present in the destination.

> **Note**
> By default, any changes to metadata on the source storage result in this metadata being copied to the destination storage.

After the **TRANSFERRING** phase is done, DataSync verifies consistency between the source and destination file systems. This is the **VERIFYING** phase.

When DataSync transfers data, it always performs data-integrity checks during the transfer. You can enable additional verification to compare the source and destination at the end of a transfer. This additional check can verify the entire dataset or only the files that were transferred as part of the task execution. For most use cases, we recommend verifying only the files that were transferred.

How AWS DataSync verifies data integrity

AWS DataSync locally calculates the checksum of every file in the source file system and the destination and compares them. Additionally, DataSync compares the metadata of every file in the source and destination and compares them. If there are differences in either one, verification fails with an error code that specifies precisely what failed. For example, you might see error codes such as `Checksum failure`, `Metadata failure`, `Files were added`, `Files were removed`, and so on.

For more information, see DataSync task creation statuses (p. 109) and **Enable verification** in the Configuring task settings (p. 103) section.

How DataSync handles open and locked files

In general, DataSync can transfer open files without any limitations.

If a file is open and it's being written to during the transfer, DataSync detects data inconsistency during the **VERIFYING** phase. This phase is when DataSync detects whether the file on the source is different from the file on the destination.

If a file is locked and the server prevents DataSync from opening it, DataSync skips transferring it. DataSync logs an error during the **TRANSFERRING** phase and sends a verification error.

Setting up

To get started, you first sign up for AWS. If you are a first-time user, we recommend that you read the Regions and requirements section.

Topics

Signing up for an AWS account

To use AWS DataSync, you need an AWS account that gives you access to all AWS resources, forums, support, and usage reports. You aren't charged for any of the services unless you use them. If you already have an AWS account, you can skip this step.

To sign up for 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.

Where can I use DataSync?

For a list of AWS Regions and endpoints that DataSync supports, see AWS DataSync endpoints and quotas in the *AWS General Reference*.

How can I use DataSync?

There are several ways to use DataSync:

- DataSync console, which is part of the AWS Management Console.
- DataSync API (p. 166) or the AWS CLI to programmatically configure and manage DataSync.
- AWS SDKs to build applications that use DataSync.

Paying for DataSync

On the AWS DataSync pricing page, create a custom estimate using the amount of data that you plan to copy.

Requirements for AWS DataSync

AWS DataSync agent and network requirements vary based on where and how you plan to transfer data.

Topics

Agent requirements

Your AWS DataSync agent must adhere to the requirements that apply to your scenario.

Topics

Supported hypervisors

DataSync supports the following hypervisor versions and hosts:

- **VMware ESXi Hypervisor (version 6.5, 6.7, or 7.0)**: A free version of VMware is available on the VMware website. You also need a VMware vSphere client to connect to the host.

  Note
  When VMware ends general support for an ESXi hypervisor version, DataSync also ends support for that version. For information about VMware's supported hypervisor versions, see VMware lifecycle policy on the VMware website.

- **Microsoft Hyper-V Hypervisor (version 2012 R2, 2016, or 2019)**: A free, standalone version of Hyper-V is available at the Microsoft Download Center. For this setup, you need a Microsoft Hyper-V Manager on a Microsoft Windows client computer to connect to the host.

  Note
  The DataSync VM is a generation 1 virtual machine. For more information about the differences between generation 1 and generation 2 VMs, see Should I create a generation 1 or 2 virtual machine in Hyper-V?

- **Linux Kernel-based Virtual Machine (KVM)**: A free, open-source virtualization technology. KVM is included in Linux versions 2.6.20 and newer. AWS DataSync is tested and supported for the CentOS/RHEL 7.8, Ubuntu 16.04 LTS, and Ubuntu 18.04 LTS distributions. Any other modern Linux distribution might work, but function or performance is not guaranteed. We recommend this option if you already have a KVM environment up and running and you're already familiar with how KVM works.

  Note
  Running KVM on Amazon EC2 isn't supported, and cannot be used for DataSync agents. To run the agent on Amazon EC2, deploy an agent Amazon Machine Image (AMI). For more information about deploying an agent AMI on Amazon EC2, see Deploy your agent as an Amazon EC2 instance (p. 23).

- **Amazon EC2 instance**: DataSync provides an Amazon Machine Image (AMI) that contains the DataSync VM image. For the recommended instance types, see Amazon EC2 instance requirements (p. 10).

Virtual machine requirements

When deploying AWS DataSync on-premises, make sure that the underlying hardware where you deploy the DataSync VM can dedicate the following minimum resources:

- **Virtual processors**: Four virtual processors assigned to the VM.
- **Disk space**: 80 GB of disk space for installation of VM image and system data.
- **RAM**: Depending on your configuration, one of the following:
  - 32 GB of RAM assigned to the VM, for tasks that transfer up to 20 million files.
  - 64 GB of RAM assigned to the VM, for tasks that transfer more than 20 million files.

Amazon EC2 instance requirements

When deploying a DataSync agent with Amazon EC2, the instance size must be at least 2xlarge.

We recommend using one of the following instance sizes:

- **m5.2xlarge**: For tasks to transfer up to 20 million files.
- **m5.4xlarge**: For tasks to transfer more than 20 million files.

> **Note**
> An exception to this is if you're running DataSync on an AWS Snowcone device. Use the default instance snc1.medium, which provides 2 CPU cores and 4 GiB of memory.

To connect to an Amazon EC2 agent using SSH, you must use the following cryptographic algorithms:

- **SSH cipher**: aes128-ctr
- **Key exchange**: diffie-hellman-group14-sha1

Network requirements

DataSync network requirements depend on how you plan to transfer data (for example, over the public internet or using a more private connection).

Use the following tables to help you configure network access for DataSync agents that transfer data from your self-managed storage system and through virtual private cloud (VPC), public service, Federal Information Processing Standard (FIPS) endpoints.

Topics

Network requirements to connect to your self-managed storage

To minimize network latency, deploy the DataSync agent close to your self-managed storage. This ensures that files travel over the network between the DataSync agent and the DataSync service using our purpose-built, accelerated protocol that significantly speeds up transfers.

The following ports are required for communication between the DataSync agent and your Network File System (NFS) server, Hadoop Distributed File System (HDFS) cluster, Server Message Block (SMB) server, or Amazon S3 API compatible storage.

From	To	Protocol	Port	How used by the DataSync agent
Agent	NFS server	TCP/UDP	2049 (NFS)	To mount a source NFS file system. Supports NFS v3.x, NFS v4.0, and NFS v4.1.
Agent	SMB server	TCP/UDP	139 (SMB) or 445 (SMB)	To mount a source SMB file share. Supports SMB 2.1 and SMB 3 versions.
Agent	Self-managed object storage	TCP	443 (HTTPS) or 80 (HTTP)	To access your self-managed object storage.
Agent	Hadoop cluster	TCP	NameNode port (default is 8020) In most clusters, you can find this port number in the `core-site.xml` file under the `fs.default` or `fs.default.name` property (depending on the Hadoop distribution).	To access the NameNodes in your Hadoop cluster. Specify the port used when creating an HDFS location.
Agent	Hadoop cluster	TCP	DataNode port (default is 50010) In most clusters, you can find this port number in the `hdfs-site.xml` file under the `dfs.datanode.address` property.	To access the DataNodes in your Hadoop cluster. The DataSync agent automatically determines the port to use.

From	To	Protocol	Port	How used by the DataSync agent
Agent	Hadoop Key Management Server (KMS)	TCP	KMS port (default 9600)	To access the KMS for your Hadoop cluster.
Agent	Kerberos Key Distribution Center (KDC) server	TCP	KDC port (default 88)	When authenticating to the Kerberos realm. This port is used only with HDFS.

Network requirements when using VPC endpoints

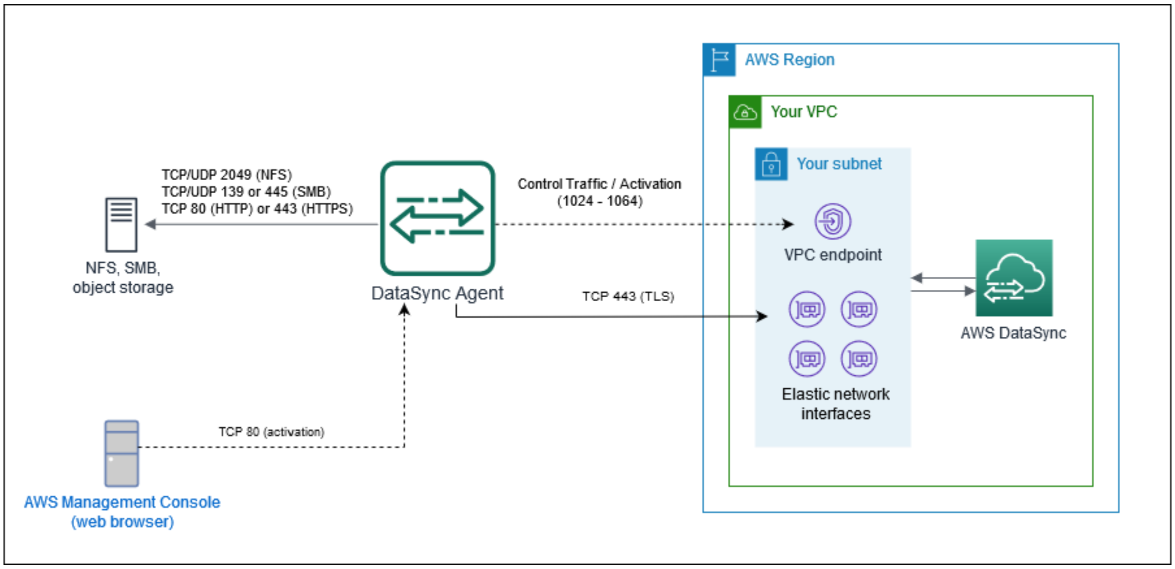
If you use only private IP addresses, you can ensure that your VPC can't be reached over the internet, and you can prevent any packets from entering or exiting the network. By using private IP addresses, you can eliminate all internet access from your self-managed systems, and still use DataSync for data transfers to and from AWS.

DataSync requires the following ports for its operation when your agent is using private endpoints.

From	To	Protocol	Port	How used
Your web browser	Your DataSync agent	TCP	80 (HTTP)	By your computer to obtain the agent activation key. After successful activation, DataSync closes the agent's port 80. The DataSync agent doesn't require port 80 to be publicly accessible. The required level of access to port 80 depends on your network configuration. **Note** Alternatively, you can obtain the activation key from the agent's local console. This method does not require connectivity between the browser and your agent. For more information about using the local console to get the activation key, see Obtaining an activation key using the local console (p. 64).

From	To	Protocol	Port	How used
Agent	Your DataSync VPC endpoint To find the correct IP address, open the Amazon VPC console, and choose **Endpoints** from the left navigation pane. Choose the DataSync endpoint, and check the **Subnets** list to find the private IP address that corresponds to the subnet that you chose for your VPC endpoint setup. For more information, see step 5 in Configuring DataSync to use private IP addresses for data transfer (p. 56).	TCP	1024–1064	For control traffic between the DataSync agent and the AWS service.
Agent	Your task's elastic network interfaces To find the related IP addresses, open the Amazon EC2 console and choose **Network Interfaces** from the left navigation pane. To see the four network interfaces for the task, enter your task ID in the search filter. For more information, see step 9 in Configuring DataSync to use private IP addresses for data transfer (p. 56).	TCP	443 (HTTPS)	For data transfer from the DataSync VM to the AWS service.
Agent	Your DataSync VPC endpoint	TCP	22 (Support channel)	To allow AWS Support to access your DataSync to help you with troubleshooting DataSync issues. You don't need this port open for normal operation, but it's required for troubleshooting.

Following is an illustration of the ports required by DataSync when using private endpoints.



Network requirements when using public service endpoints or FIPS endpoints

Your agent VM requires access to the following endpoints to communicate with AWS when using public service endpoints, or when using FIPS endpoints. Enabling this access is not necessary when using DataSync with VPC endpoints.

If you use a firewall or router to filter or limit network traffic, configure your firewall or router to allow these service endpoints. They're required to enable outbound communication between your network and AWS.

From	To	Protocol	Port	How used	Endpoints accessed by the agent
Your web browser	DataSync agent	TCP	80 (HTTP)	Used by your computer to obtain the agent activation key. After successful activation, DataSync closes the agent's port 80. The DataSync agent doesn't require port 80 to be publicly accessible. The required level of access to port 80 depends on your network configuration.	N/A

From	To	Protocol	Port	How used	Endpoints accessed by the agent
				Note Alternatively, you can obtain the activation key from the agent's local console. This method does not require connectivity between the browser and your agent. For more information about using the local console to get the activation key, see Obtaining an activation key using the local console (p. 64).	

Agent	AWS	TCP	443 (HTTPS)	Used by the DataSync agent to activate with your AWS account. You can block the public endpoints after activation.	For public endpoint activation:
					`activation.datasync.us-east-2.amazonaws.com`
					For FIPS endpoint activation:
					`activation.datasync-fips.us-east-2.amazonaws.com`

From	To	Protocol	Port	How used	Endpoints accessed by the agent

Agent	AWS	TCP	443 (HTTPS)	For communication between the DataSync agent and the AWS service endpoint.
				For information about Regions and service endpoints, see Choose a service endpoint for AWS DataSync (p. 25).
					API endpoints:
					`datasync.us-east-2.amazonaws.com`
					Data transfer endpoints:
					`yourTaskId.datasync-dp.us-east-2.amazonaws.com`
					`cp.datasync.us-east-2.amazonaws.com`
					Data transfer endpoints for FIPS:
					`cp.datasync-fips.us-east-2.amazonaws.com`

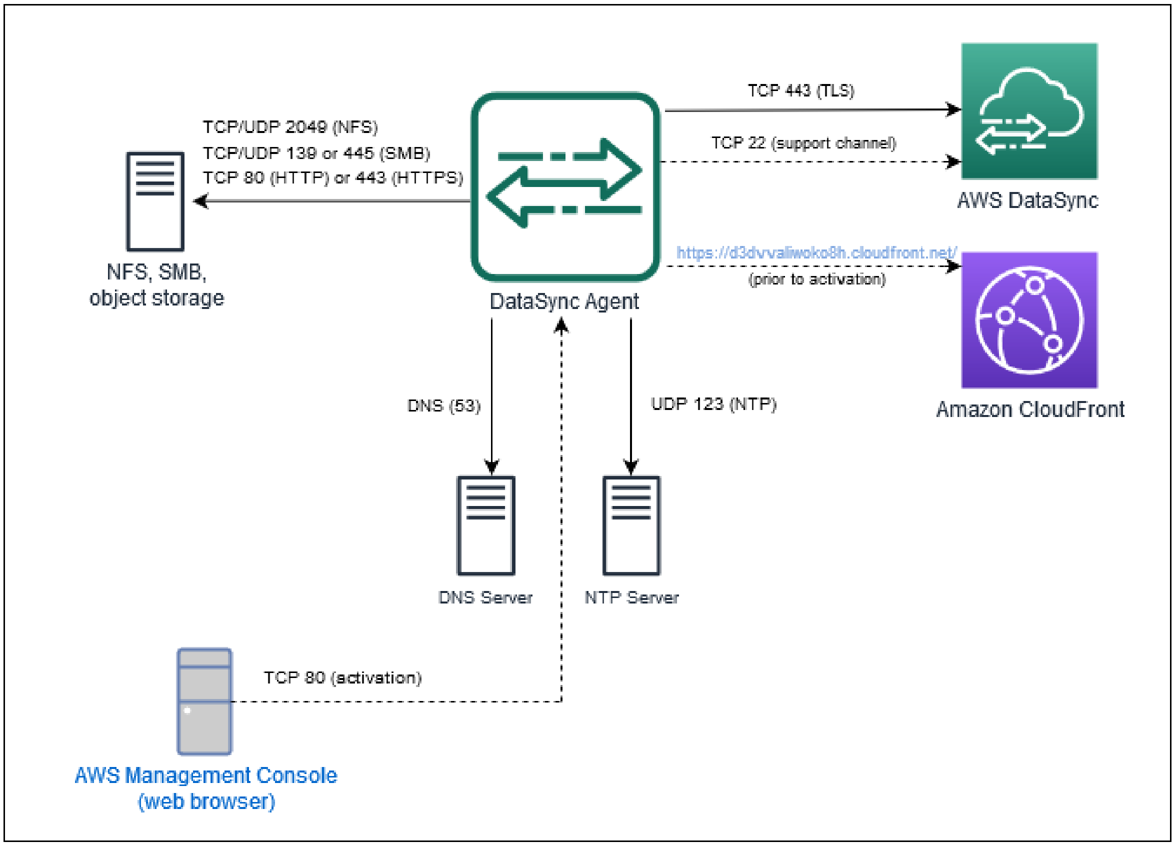
Agent	AWS	TCP	80 (HTTP)	Allows the DataSync agent to get updates from AWS.	The *activation_region* variable is the AWS Region you used to activate your DataSync agent.
					`repo.default.amazonaws.com`
					`packages.us-west-1.amazonaws.com`
					`packages.sa-east-1.amazonaws.com`
					`repo.$activation_region.amazonaws.com`
					`packages.$activation_region.amazonaws.com`

Agent	AWS	TCP	443 (HTTPS)	Allows the DataSync agent to get updates from AWS.	The *activation_region* variable is the AWS Region you used to activate your DataSync agent.
					`amazonlinux.default.amazonaws.com`
					`cdn.amazonlinux.com`
					`amazonlinux-2-repos-$activation_region.s3.dualstack.$activation_region.amazonaws.com`
					`amazonlinux-2-repos-$activation_region.s3.$activation_region.amazonaws.com`
					`*.s3.$activation_region.amazonaws.com`

From	To	Protocol	Port	How used	Endpoints accessed by the agent
Agent	Domain Name Service (DNS) server	TCP/UDP	53 (DNS)	For communication between DataSync agent and the DNS server.	N/A
Agent	AWS	TCP	22 (Support channel)	Allows AWS Support to access your DataSync to help you with troubleshooting DataSync issues. You don't need this port open for normal operation, but it's required for troubleshooting.	AWS support channel: `54.201.223.107`

Agent	Network Time Protocol (NTP) server	UDP	123 (NTP)	Used by local systems to synchronize the VM time to the host time.	NTP:
					`0.amazon.pool.ntp.org`
					`1.amazon.pool.ntp.org`
					`2.amazon.pool.ntp.org`
					`3.amazon.pool.ntp.org`
					Note
					If you want to change the default NTP configuration of your VM agent to use a different NTP server using the local console, see Configuring a Network Time Protocol (NTP) server for VMware agents (p. 69).

The following diagram shows the ports required by DataSync when using public service endpoints or FIPS endpoints.



Required network interfaces for data transfers

For every task you run, DataSync automatically creates and manages elastic network interfaces for data transfer traffic. How many network interfaces DataSync creates and where they're created depends on the following details about your task:

- Whether your task requires a DataSync agent.
- Your source and destination locations (where you're copying data from and to).
- The type of endpoint used to activate your agent.

Each network interface uses a single IP address in your subnet (the more network interfaces there are, the more IP addresses you need). Use the following tables to make sure your subnet has enough IP addresses for your task.

Transfers with agents

You need a DataSync agent when copying data between a self-managed storage system and an AWS storage service.

Location	Network interfaces created by default	Where network interfaces are created when using a public or FIPS endpoint	Where network interfaces are created when using a private (VPC) endpoint
Amazon S3	4	N/A (network interfaces aren't	The subnet you specified when

Location	Network interfaces created by default	Where network interfaces are created when using a public or FIPS endpoint	Where network interfaces are created when using a private (VPC) endpoint
		needed since DataSync communicates directly with the S3 bucket)	activating your DataSync agent.
Amazon EFS	4	The subnet you specify when creating the Amazon EFS location.	
Amazon FSx for Windows File Server	4	The same subnet as the preferred file server for the file system.	
Amazon FSx for Lustre	4	The same subnet as the file system.	
Amazon FSx for OpenZFS	4	The same subnet as the file system.	

Transfers without agents

You don't need a DataSync agent when copying data between AWS services.

> **Note**
> The total number of network interfaces depends on your DataSync task locations. For example, transferring from an Amazon EFS location to FSx for Lustre requires four network interfaces. Meanwhile, transferring from an FSx for Windows File Server to an Amazon S3 bucket requires two network interfaces.

Location	Network interfaces created by default	Where network interfaces are created
Amazon S3	N/A (network interfaces aren't needed since DataSync communicates directly with the S3 bucket)	
Amazon EFS	2	The subnet you specify when creating the Amazon EFS location.
Amazon FSx for Windows File Server	2	The same subnet as the preferred file server for the file system.
Amazon FSx for Lustre	2	The same subnet as the file system.
Amazon FSx for OpenZFS	2	The same subnet as the file system.

To see the network interfaces allocated for your DataSync task, use the DescribeTask operation.

Getting started with AWS DataSync

In this topic, you can find step-by-step instructions on how to get started using AWS DataSync on the AWS Management Console.

Before you begin, we recommend reading How AWS DataSync works (p. 3) to understand the components and terms used in DataSync and how DataSync works. We also recommend reading the Using identity-based policies (IAM policies) for DataSync (p. 125) section to understand the AWS Identity and Access Management (IAM) permissions that DataSync requires.

To use AWS DataSync

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the upper-right corner, select the AWS Region where you want to run DataSync.

   We recommend choosing the AWS Region where the Amazon S3 bucket, Amazon EFS file system, Amazon FSx for Windows File Server file system, Amazon FSx for Lustre file system, or Amazon FSx for OpenZFS file system involved in your transfer resides.

   If you haven't created DataSync resources in this AWS Region before, the DataSync home page appears.
3. On the DataSync home page, select whether to create the data transfer task **Between on-premises storage and AWS** or **Between AWS Storage services**.
4. Choose **Get started** to begin using DataSync.

   If this is your first time using DataSync in this AWS Region, the **Create agent** page appears. From this page, you can download your virtual machine (VM) or create an Amazon EC2 instance.

   If you have used DataSync in this AWS Region, the **Agents** page appears and you can see your agents listed.

Next, take the following steps.

Topics

Create an AWS DataSync agent

For AWS DataSync to access your self-managed storage (whether on-premises or in the cloud), you need a DataSync agent associated with your AWS account.

Tip
An agent isn't required when transferring between AWS storage services in the same AWS account. To set up a data transfer between two AWS services, see Configure a source location (p. 28).

Topics

- Deploy your DataSync agent (p. 21)
- Choose a service endpoint for AWS DataSync (p. 25)
- Activate your AWS DataSync agent (p. 27)

Deploy your DataSync agent

Where you deploy your AWS DataSync agent depends on where you're copying data to and from and whether you're working with on-premises or in-cloud storage systems.

Topics

- Deploy your agent on VMware (p. 21)
- Deploy your agent on KVM (p. 22)
- Deploy your agent on Hyper-V (p. 22)
- Deploy your agent as an Amazon EC2 instance (p. 23)
- Deploy your agent on AWS Snowcone (p. 25)
- Deploy your agent on AWS Outposts (p. 25)

Deploy your agent on VMware

You can download and deploy an AWS DataSync agent in your VMware environment and then activate it. You can also use an existing agent instead of deploying a new one. You can use a previously created agent if it can access your self-managed storage and if it's activated in the same AWS Region.

To deploy an agent on VMware

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. If you don't have an agent, on the **Create agent** page in the console, choose **Download image** in the **Deploy agent** section. Doing this downloads the agent and deploys it in your VMware ESXi hypervisor. The agent is available as a VM. If you want to deploy the agent as an Amazon EC2 instance, see Deploy your agent as an Amazon EC2 instance (p. 23).

   AWS DataSync currently supports the VMware ESXi hypervisor. For information about hardware requirements for the VM, see Virtual machine requirements (p. 10). For information about how to deploy an `.ova` file in a VMware host, see the documentation for your hypervisor.

   If you have previously activated an agent in this AWS Region and want to use that agent, choose that agent and choose **Create agent**. The Configure a source location (p. 28) page appears.
3. Power on your hypervisor, log in to your VM, and get the IP address of the agent. You need this IP address to activate the agent.

   > **Note**
   > The VM's default credentials are the login **`admin`** and the password **`password`**.
   > You can change the password on the local console. You don't need to log in to the VM for DataSync functionality. Login is mainly required for troubleshooting, such as running a connectivity test or opening a support channel with AWS. It's also required for network-specific settings, such as setting up a static IP address.

After you have deployed an agent, you choose a service endpoint (p. 25).

Deploy your agent on KVM

You can download and deploy an AWS DataSync agent in your KVM environment and then activate it. You can also use an existing agent instead of deploying a new one. You can use a previously created agent if it can access your self-managed storage and if it's activated in the same AWS Region.

To deploy an agent on KVM

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. If you don't have an agent, on the **Create agent** page in the console, choose **Download image** in the **Deploy agent** section. Doing this downloads the agent in a `.zip` file that contains a `.qcow2` image file that can you can deploy in your KVM hypervisor.

   The agent is available as a VM. If you want to deploy the agent as an Amazon EC2 instance, see Deploy your agent as an Amazon EC2 instance (p. 23).

   AWS DataSync currently supports the KVM hypervisor. For information about hardware requirements for the VM, see Virtual machine requirements (p. 10).

   To get started installing your `.qcow2` image for use in KVM, use the following command.

   ```
   virt-install \
   --name "datasync" \
   --description "AWS DataSync agent" \
   --os-type=generic \
   --ram=32768 \
   --vcpus=4 \
   --disk path=datasync-yyyymmdd-x86_64.qcow2,bus=virtio,size=80 \
   --network default,model=virtio \
   --graphics none \
   --import
   ```

   For information about how to manage this VM, and your KVM host, see the documentation for your hypervisor.

   If you previously activated an agent in this AWS Region and want to use that agent, choose that agent, and then choose **Create agent**. The Configure a source location (p. 28) page appears.
3. Power on your hypervisor, log in to your VM, and get the IP address of the agent. You need this IP address to activate the agent.

   > **Note**
   > The VM's default credentials are the login **`admin`** and the password **`password`**.
   > You can change the password on the local console. You don't need to log in to the VM for DataSync functionality. Login is mainly required for troubleshooting, such as running a connectivity test or opening a support channel with AWS. It's also required for network-specific settings, such as setting up a static IP address.

After you deploy an agent, you choose a service endpoint (p. 25).

Deploy your agent on Hyper-V

You can download and deploy an AWS DataSync agent in your Hyper-V environment and then activate it. You can also use an existing agent instead of deploying a new one. You can use a previously created agent if it can access your self-managed storage and if it's activated in the same AWS Region.

To deploy an agent on Hyper-V

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. If you don't have an agent, on the **Create agent** page in the console, choose **Download image** in the **Deploy agent** section. Doing this downloads the agent in a `.zip` file that contains a `.vhdx` image file that can you can deploy in your Hyper-V hypervisor.

   The agent is available as a VM. If you want to deploy the agent as an Amazon EC2 instance, see Deploy your agent as an Amazon EC2 instance (p. 23).

   AWS DataSync currently supports the Hyper-V hypervisor. For information about hardware requirements for the VM, see Virtual machine requirements (p. 10). For information about how to deploy a `.vhdx` file in a Hyper-V host, see the documentation for your hypervisor.

   If you previously activated an agent in this AWS Region and want to use that agent, choose that agent, and then choose **Create agent**. The Configure a source location (p. 28) page appears.
3. Power on your hypervisor, log in to your VM, and get the IP address of the agent. You need this IP address to activate the agent.

   > **Note**
   > The VM's default credentials are the login **`admin`** and the password **`password`**.
   > You can change the password on the local console. You don't need to log in to the VM for DataSync functionality. Login is mainly required for troubleshooting, such as running a connectivity test or opening a support channel with AWS. It's also required for network-specific settings, such as setting up a static IP address.

After you deploy an agent, you choose a service endpoint (p. 25).

Deploy your agent as an Amazon EC2 instance

You deploy a DataSync agent as an Amazon EC2 instance when copying data between:

- A self-managed, in-cloud storage system and an AWS storage service.

  For more information about these use cases, including high-level architecture diagrams, see Deploying your DataSync agent in AWS Regions (p. 59).
- Amazon S3 on AWS Outposts (p. 25) and an AWS storage service.

> **Warning**
> We don't recommend using an Amazon EC2 agent to access your on-premises storage because of increased network latency. Instead, deploy the agent as a VMware, KVM, or Hyper-V virtual machine in your data center as close to your on-premises storage as possible.

To choose the agent AMI for your AWS Region

- Use the following CLI command to get the latest DataSync Amazon Machine Image (AMI) ID for the specified AWS Region.

  ```
  aws ssm get-parameter --name /aws/service/datasync/ami --region region
  ```

  #### Example Example command and output

  ```
  aws ssm get-parameter --name /aws/service/datasync/ami --region us-east-1
  ```

```
{
    "Parameter": {
        "Name": "/aws/service/datasync/ami",
        "Type": "String",
        "Value": "ami-id",
        "Version": 6,
        "LastModifiedDate": 1569946277.996,
        "ARN": "arn:aws:ssm:us-east-1::parameter/aws/service/datasync/ami"
    }
}
```

To deploy your DataSync agent as an Amazon EC2 instance

> **Important**
> To avoid charges, deploy your agent in a way that it doesn't require network traffic between Availability Zones. For example, deploy your agent in the Availability Zone where your self-managed file system resides.
> To learn more about data transfer prices for all AWS Regions, see Amazon EC2 On-Demand pricing.

1. From the AWS account where the source file system resides, launch the agent using your AMI from the Amazon EC2 launch wizard. Use the following URL to launch the AMI.

   ```
   https://console.aws.amazon.com/ec2/v2/home?region=source-file-system-
   region#LaunchInstanceWizard:ami=ami-id
   ```

   In the URL, replace the *`source-file-system-region`* and *`ami-id`* with your own source AWS Region and AMI ID. The **Choose an Instance Type** page appears on the Amazon EC2 console.
2. Choose one of the recommended instance types for your use case, and choose **Next: Configure Instance Details**. For the recommended instance types, see Amazon EC2 instance requirements (p. 10).
3. On the **Configure Instance Details** page, do the following:

   a. For **Network**, choose the virtual private cloud (VPC) where your source Amazon EFS or NFS file system is located.

   b. For **Auto-assign Public IP**, choose a value. For your instance to be accessible from the public internet, set **Auto-assign Public IP** to **Enable**. Otherwise, set **Auto-assign Public IP** to **Disable**. If a public IP address isn't assigned, activate the agent in your VPC using its private IP address.

      When you transfer files from an in-cloud file system, to increase performance we recommend that you choose a **Placement Group** value where your NFS server resides.
4. Choose **Next: Add Storage**. The agent doesn't require additional storage, so you can skip this step and choose **Next: Add tags**.
5. (Optional) On the **Add Tags** page, you can add tags to your Amazon EC2 instance. When you're finished on the page, choose **Next: Configure Security Group**.
6. On the **Configure Security Group** page, do the following:

   a. Make sure that the selected security group allows inbound access to HTTP port 80 from the web browser that you plan to use to activate the agent.

   b. Make sure that the security group of the source file system allows inbound traffic from the agent. In addition, make sure that the agent allows outbound traffic to the source file system. If you deploy your agent using a VPC endpoint, you need to allow additional ports. For more information, see How DataSync works with VPC endpoints (p. 56).

   For the complete set of network requirements for DataSync, see Network requirements (p. 10).

7. Choose **Review and Launch** to review your configuration, then choose **Launch** to launch your instance. Remember to use a key pair that's accessible to you. A confirmation page appears and indicates that your instance is launching.
8. Choose **View Instances** to close the confirmation page and return to the Amazon EC2 instances screen. When you launch an instance, its initial state is **pending**. After the instance starts, its state changes to **running**. At this point, it's assigned a public Domain Name System (DNS) name and IP address, you can find these in the **Descriptions** tab.
9. If you set **Auto-assign Public IP** to **Enable**, choose your instance and note the public IP address in the **Description** tab. You use this IP address later to connect to your sync agent.

   If you set **Auto-assign Public IP** to **Disable**, launch or use an existing instance in your VPC to activate the agent. In this case, you use the private IP address of the sync agent to activate the agent from this instance in the VPC.

Deploy your agent on AWS Snowcone

The DataSync agent AMI is pre-installed on your Snowcone device. Launch the agent with one of the following tools:

- AWS OpsHub
- Snowball Edge client

Deploy your agent on AWS Outposts

You can launch a DataSync Amazon EC2 instance on your AWS Outpost. To learn more about launching an AMI on AWS Outposts, see Launch an instance on your Outpost in the *AWS Outposts User Guide*.

When using DataSync to access Amazon S3 on Outposts, you must launch the agent in a VPC that's allowed to access your Amazon S3 access point, and activate the agent in the Outpost's parent Region. The agent must also be able to route to the Amazon S3 on Outposts endpoint for the bucket. To learn more about working with Amazon S3 on Outposts endpoints, see Working with Amazon S3 on Outposts in the *Amazon S3 User Guide*.

Choose a service endpoint for AWS DataSync

You must specify an endpoint that your AWS DataSync agent uses to communicate with AWS. The agent can connect to the following types of endpoints:

- **Public endpoints**: If you use public endpoints, all communication from your DataSync agent to AWS occurs over the public internet. For instructions, see Choose a public service endpoint (p. 26).
- **Federal Information Processing Standard (FIPS) endpoints**: If you need FIPS 140-2 validated cryptographic modules when accessing the AWS GovCloud (US-East) or AWS GovCloud (US-West) Region, use this endpoint to activate your agent. You use the AWS CLI or API to access this endpoint. For more information, see Federal Information Processing Standard (FIPS) 140-2.
- **Virtual private cloud (VPC) endpoints**: If you use a VPC endpoint, all communication from DataSync to AWS occurs through the endpoint in your VPC. This establishes a private connection between your self-managed storage system, your VPC, and AWS services, providing extra security as your data is copied over the network. For instructions, see Using AWS DataSync in a virtual private cloud (p. 56).

> **Note**
> After you choose a service endpoint type and activate your agent, you can't change it to use a different service endpoint type later. If you need to transfer data to multiple endpoint types, create a DataSync agent for each endpoint type that you use.

For more information about service endpoints, see AWS DataSync in the *AWS General Reference*.

Topics

- Choose a public service endpoint (p. 26)
- Choose a FIPS service endpoint (p. 26)
- Choose a VPC endpoint (p. 26)

Choose a public service endpoint

If you use a public endpoint, all communication from your DataSync agent to AWS occurs over the public internet.

To choose a public service endpoint

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. Go to the **Agents** page and choose **Create agent**.
3. In the **Service endpoint** section, choose **Public service endpoints in *`AWS Region name`***. For a list of supported AWS Regions, see AWS DataSync in the *AWS General Reference*.

Next Step: the section called "Activate your agent" (p. 27)

Choose a FIPS service endpoint

If you use a FIPS service endpoint, DataSync communicates with the AWS GovCloud (US) or Canada (Central) Region.

To choose a FIPS service endpoint

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. For **Hypervisor**, choose the type of agent you deployed.
3. In the **Service endpoint** section, choose the FIPS endpoint that you want. For information about supported FIPS endpoint, see AWS DataSync in the *AWS General Reference*.

Next step: the section called "Activate your agent" (p. 27)

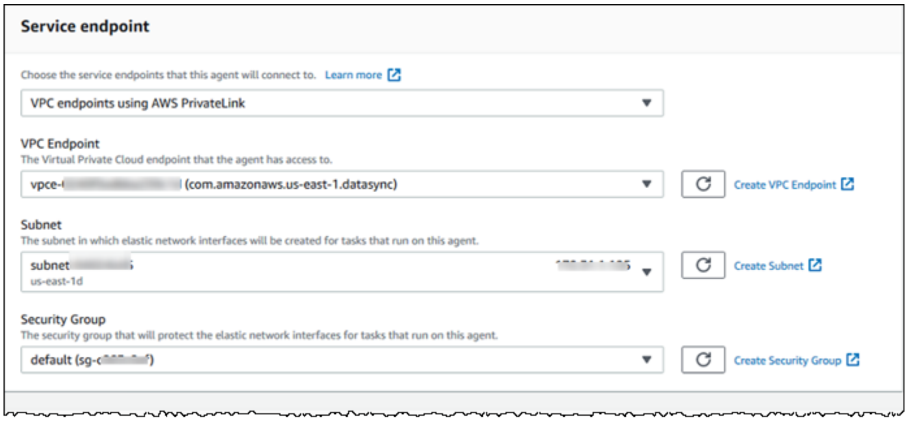
Choose a VPC endpoint

If you use a VPC endpoint, all communication from DataSync to AWS services occurs through the VPC endpoint in your VPC in AWS. This approach provides a private connection between your self-managed storage, your VPC, and AWS services.

You can also use a VPC endpoint outside your VPC to connect your data center directly to AWS resources. In this case, you use a virtual private network (VPN) or AWS Direct Connect. You set up a VPC route table to use the endpoint to access the service. For detailed information, see Routing for gateway endpoints.

To choose a VPC endpoint

1. Create a VPC endpoint. If you already have a VPC endpoint in the AWS Region, you can use it.
2. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
3. Go to the **Agents** page and choose **Create agent**.
4. For **Hypervisor**, choose **Amazon EC2**.
5. In the **Service endpoint** section, choose **VPC endpoints using AWS PrivateLink**. This is the VPC endpoint that the agent has access to.



Service endpoint

Choose the service endpoints that this agent will connect to. Learn more

VPC endpoints using AWS PrivateLink

VPC Endpoint

The Virtual Private Cloud endpoint that the agent has access to.

vpce-[redacted] (com.amazonaws.us-east-1.datasync)

Subnet

The subnet in which elastic network interfaces will be created for tasks that run on this agent.

subnet[redacted]
us-east-1d [redacted]

Security Group

The security group that will protect the elastic network interfaces for tasks that run on this agent.

default (sg-[redacted])

6. For **VPC Endpoint**, choose the private VPC endpoint that you want your agent to connect to.

   You noted the endpoint ID when you created the VPC endpoint.

   > **Important**
   > You must choose a VPC endpoint that includes the DataSync service name (for example, `com.amazonaws.us-east-2.datasync`).

7. For **Subnet**, choose the subnet in which you want to run your task.

   This is the subnet where the elastic network interfaces (p. 18) are created.

8. For **Security Group**, choose a security group for your task.

   This is the security group that protects your network interface for tasks that run on your agent.

For additional information about using DataSync in a VPC, see Using AWS DataSync in a virtual private cloud (p. 56).

Next step: the section called “Activate your agent” (p. 27)

Activate your AWS DataSync agent

After you deploy your AWS DataSync agent and specify a service endpoint, you must activate the agent to associate it with your AWS account.

> **Note**
> An agent can be associated with only one AWS account at a time.

To activate your agent

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. Go to the **Agents** page and choose **Create agent**.
3. In the **Activation key** section, select **Automatically get the activation key from your agent**.

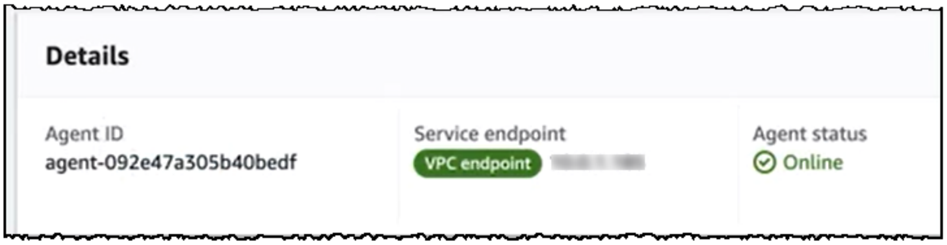
   This option requires that your browser access the agent using port 80. Once activated, the agent closes the port. For more information, see Network requirements (p. 10).

   Alternatively, select **Manually enter your agent's activation key** if you don't want a connection between your browser and agent. For more information, see Obtaining an activation key using the local console (p. 64).

4. For the **Agent address**, enter the agent's IP address or domain name and select **Get key**. Your browser connects to the IP address and gets a unique activation key from your agent.

If activation succeeds, the activation key is displayed. If the activation fails, make sure that your security group is configured properly and verify that your firewall allows the required ports.

5. (Optional) For **Agent name**, enter a name for your agent.
6. (Optional) For **Tags**, enter a key and value to add a tag to your agent. A *tag* is a key-value pair that helps you manage, filter, and search for your agents.
7. Choose **Create agent**. Your agent is listed on the **Agents** page. In the **Service endpoint** column, verify that your service endpoint is correct.



Details

Agent ID
agent-092e47a305b40bedf

Service endpoint
VPC endpoint [redacted]

Agent status
✓ Online

8. In the **Tasks** section of the page, choose **Create task**. The **Configure source location** page appears.

Configure a source location

A *task* consists of a pair of locations that data will be transferred between. The *source location* defines the storage system or service that you want to read data from. The *destination location* defines the storage system or service that you want to write data to.

For a list of all DataSync supported source and destination endpoints, see Working with AWS DataSync locations (p. 72).

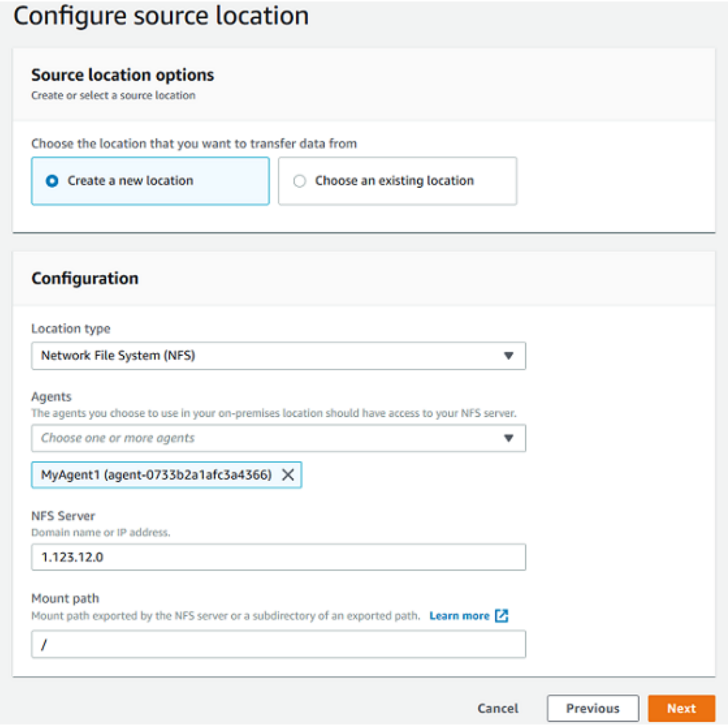
In the following walkthrough, we give an example of configuring a Network File System (NFS) file system as the source location.

To configure a different location type as your source location, see the following topics:

- Creating a location for NFS (p. 74)
- Creating a location for SMB (p. 75)
- Creating a location for HDFS (p. 77)
- Creating a location for object storage (p. 79)
- Creating a location for Amazon EFS (p. 81)
- Creating a location for FSx for Windows File Server (p. 84)
- Creating a location for FSx for Lustre (p. 86)
- Creating a location for FSx for OpenZFS (p. 87)
- Creating a location for Amazon S3 (p. 88)

To create an NFS location

1. On the **Configure source location** page, choose **Create a new location** or **Choose existing location**. **Create a new location** enables you to define a new location and **Choose existing location** enables you to choose from locations that you have previously created in this AWS Region.



Configure source location

Source location options
Create or select a source location

Choose the location that you want to transfer data from

● Create a new location ○ Choose an existing location

Configuration

Location type
Network File System (NFS)

Agents
The agents you choose to use in your on-premises location should have access to your NFS server.
Choose one or more agents

MyAgent1 (agent-0733b2a1afc3a4366)

NFS Server
Domain name or IP address.
1.123.12.0

Mount path
Mount path exported by the NFS server or a subdirectory of an exported path. Learn more
/

2. For **Location type** in the **Configuration** section, choose your NFS server from the list.
3. For **Agents**, choose your agent from the list. You can add more than one agent. For this walkthrough, we add only one agent.

   Note
   In many cases, you might be transferring from an in-cloud NFS file system or an Amazon EFS file system. In such cases, make sure that you choose an agent that you created in an Amazon EC2 instance that can access this file system.
   You can't use agents that are created with different endpoint types for the same task.
4. For **NFS server**, enter the IP address or domain name of your NFS server. An agent that's installed on-premises uses this hostname to mount the NFS server in a network. The NFS server should allow full access to all files.
5. For **Mount path**, enter a path that's exported by the NFS server, or a subdirectory that can be mounted by other NFS clients in your network. The path is used to read data from or write data to your NFS server.
6. Choose **Next** to open the **Configure destination location** page.

Configure a destination location

A *task* consists of a pair of locations that data will be transferred between. The *source location* defines the storage system or service that you want to read data from. The *destination location* defines the storage system or service that you want to write data to.

For a list of all DataSync supported source and destination endpoints, see Working with AWS DataSync locations (p. 72).

To configure a different location type, see the following topics:

- Creating a location for NFS (p. 74)
- Creating a location for SMB (p. 75)

- Creating a location for HDFS (p. 77)
- Creating a location for object storage (p. 79)
- Creating a location for Amazon EFS (p. 81)
- Creating a location for FSx for Windows File Server (p. 84)
- Creating a location for FSx for Lustre (p. 86)
- Creating a location for FSx for OpenZFS (p. 87)
- Creating a location for Amazon S3 (p. 88)

Configure task settings

After you have created an AWS DataSync agent and configured the source and destination locations, you can configure the settings for a new task. A task is a set of two locations (source and destination) and a set of options that you use to control the behavior of the task.

You configure task settings when creating a new task in the AWS DataSync console. You can also edit task settings by opening the AWS DataSync console at https://console.aws.amazon.com/datasync/, selecting the task you want to edit, and choosing **Edit**.

On the **Configure settings** page, for **Task name - *optional***, enter a name for your task. **Task name** is an optional setting.

The **Options** section contains configuration options for running your task. The following sections provide more details about these options.

Topics

Data verification options

As DataSync transfers data, it always performs data integrity checks during the transfer. You can enable additional verification to compare source and destination at the end of a transfer. This additional check can verify the entire dataset or only the files that were transferred as part of the task execution. For most use cases, we recommend verifying only the files transferred.

Task data verification options specify how to verify data that's transferred by the task.

Data verification options are as follows:

- **Verify only the data transferred (recommended)** – This option calculates the checksum of transferred files and metadata on the source. It then compares this checksum to the checksum calculated on those files at the destination at the end of the transfer. We recommend this option when transferring to S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes. For more information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).
- **Verify all data in the destination** – This option performs a scan at the end of the transfer of the entire source and entire destination to verify that source and destination are fully synchronized. You can't

use this option when transferring to S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes. For more information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).

- **Check integrity during the transfer** – This option doesn't run additional verification at the end of the transfer. All data transmissions are still integrity-checked with checksum verification during the transfer.

Ownership and permissions-related options

DataSync preserves metadata between storage systems that have similar metadata structures. Different options are used to configure such metadata preservation depending on the storage system type.

When copying data between Network File System (NFS), Hadoop Distributed File System (HDFS), Amazon EFS, Amazon FSx for Lustre, Amazon FSx for OpenZFS, and Amazon S3, choose one of the following (if applicable):

- Choose **Copy ownership** to have DataSync copy POSIX file and folder ownership, such as the group ID of the file's owners and the user ID of the file's owner.
- Choose **Copy permissions** to have DataSync copy POSIX permissions for files and folders from the source to the destination.

When copying between Server Message Block (SMB) and FSx for Windows File Server, or between two FSx for Windows File Server locations, choose one of the following (if applicable):

- Choose **Copy ownership, DACLs, and SACLs** to have DataSync copy the following:
  - The object owner.
  - NTFS discretionary access lists (DACLs), which determine whether to grant access to an object.
  - NTFS system access control lists (SACLs), which are used by administrators to log attempts to access a secured object.

  Copying SACLs requires granting additional permissions to the Windows user that DataSync uses to access your SMB location. See user (p. 76) to learn more about choosing a user name that ensures sufficient permissions to files, folders, and metadata.
- Choose **Copy ownership and DACLs** to have DataSync copy the following:
  - The object owner.
  - NTFS discretionary access lists (DACLs), which determine whether to grant access to an object.

  DataSync won't copy NTFS system access control lists (SACLs) when you choose this option.
- Choose **Do not copy ownership or ACLs** if you want DataSync not to copy any ownership or permissions data. The objects that DataSync writes to your destination location are owned by the user whose credentials are provided for DataSync to access the destination location. Destination object permissions are determined based on the permissions configured on the destination server.

For more information about metadata preservation using DataSync, see How DataSync handles metadata and special files (p. 93).

File metadata and management options

You can configure how you want DataSync to handle aspects of your files and objects during a transfer:

- Choose **Copy timestamps** to have DataSync copy the timestamp metadata from the source to the destination.

- Choose **Keep deleted files** to have DataSync keep files in the destination that don't exist in the source file system.

  If your task deletes objects, from your Amazon S3 bucket, you might incur minimum storage duration charges for certain storage classes. For detailed information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).
- Choose **Overwrite files** if you want files at the destination to be overwritten by files from the source when the source data or metadata is different.

  If you don't choose this option, the destination file isn't replaced by the source file, even if the destination file differs from the source file.

  If your task overwrites objects, you might incur additional charges for certain storage classes (for example, for retrieval or early deletion). For detailed information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).
- Choose **Copy object tags** if you want to preserve the tags associated with your objects when transferring between object storage systems.

Bandwidth options

You can configure a bandwidth limit for DataSync tasks. Bandwidth limit options are as follows:

- Choose **Use available** to have DataSync use all the network bandwidth that is available for the transfer.
- Choose **Set bandwidth limit (MiB/s)** to limit the maximum bandwidth that you want DataSync to use for this task.

  You can change bandwidth limits for an in-progress task execution. For more information, see Adjusting bandwidth throttling for a task execution (p. 110).

Filtering options

When you transfer data from your source to your destination location, you can apply filters to transfer only a subset of the files in your source location. The configuration options for filtering are as follows.

- In the **Data transfer configuration** section, use the **Exclude patterns** section to specify files, folders, and objects to exclude from your transfer. To include specific files, folders, and objects in your transfer, select **Specific files and folders** and then use the **Include patterns** section.
- To add additional patterns to your filters, choose **Add pattern**. For detailed information about filtering and syntax for creating patterns, see Filtering data transferred by AWS DataSync (p. 104).
- You can modify filter patterns when you edit a task. You can also specify different patterns each time that you execute a task.

Scheduling and queueing options

You can schedule a DataSync task to be run at a specific time. If you are using a single agent to run multiple tasks, you can queue those tasks. Configuring options for scheduling are as follows:

- In the **Schedule (optional)** section, configure your task to run on a schedule that you specify, with a minimum interval of 1 hour.
- For **Frequency**, configure how frequently you want the task to run. For frequency configuration options, see Configuring a task schedule (p. 108).

If you are using a single agent to run multiple tasks, choose **Queueing** to make the tasks run in series (first in, first out). For more information, see Queueing task executions (p. 109).

Tags and logging options

You can add one or more tags to a DataSync task. A tag is a key-value pair that is associated with the task. You can also choose logging options to have DataSync publish logs for individual files or objects to the CloudWatch log group that you specify. Tags and logging options are as follows:

- In the **Tags** section, enter **Key** and **Value** to tag your task. A *tag* is a key-value pair that helps you manage, filter, and search for your tasks. We recommend that you create a name tag for your task.
- Choose **Task logging** to have DataSync publish logs for individual files or objects to the CloudWatch log group that you specify.

  To upload logs to your CloudWatch log group, DataSync requires a resource policy that grants sufficient permissions. If you don't have a policy in the current Region, a check box appears so that you can create the required policy automatically. For an example of such a policy, see Allowing DataSync to upload logs to Amazon CloudWatch log groups (p. 115).

  For more information about using log groups and streams, see Working with Log Groups and Log Streams in the *Amazon CloudWatch Logs User Guide*.

  Use the **Log level** option to set the level of detail that is logged to CloudWatch Logs. Log level options include the following:
  - Choose **Log basic information such as transfer errors** to publish only basic information (such as transfer errors) to CloudWatch.
  - Choose **Log all transferred objects, files, and folders** to publish log records to CloudWatch Logs for all files or objects that the task copies and integrity checks.
  - Choose **Do not send logs to CloudWatch** if you don't want DataSync logs to be published to CloudWatch.

Choose **Next** to open the **Review** page.

Review your settings and create your task

Next, you review your settings and create your task.

To review your settings

1. On the **Review** page, review and edit your configuration and settings if necessary. You can edit the settings on the page by choosing **Previous** at the bottom of the page. For more information about task settings, see Creating your DataSync task (p. 98).
2. When you are done reviewing, choose **Create task**. The **Status** value of the task is now **Creating**. During the **Creating** status, AWS DataSync attempts to mount the source NFS location. Wait for the task to transition to the **Available** status before you run the task.

Start your task

Next, you start your task. You can further review your configuration settings before you start the task,

To start your task with the default configuration

1. When the **Status** of the task changes from **Creating** to **Available**, choose **Start**..

2. Choose **Start with defaults**.

   The task starts and you're redirected to the **Task execution** page.

To start your task with a modified configuration

1. When the **Status** of the task changes from **Creating** to **Available**, choose **Start**, and then **Start with overriding options**.
2. Modify the settings you want to change before starting the task.
3. Review your changes and choose **Start**.

   The task starts and you're redirected to the **Task execution** page.

When you create a task, it first enters the **Creating** state. While the task is in the **Creating** state, AWS DataSync performs validation checks on the source and destination locations. After DataSync validates the locations, the task transitions to the **Available** state. If an agent on the source location goes offline, the task transitions to the **Unavailable** state.

For information about how DataSync transfers files, see How DataSync transfers files (p. 6).

Clean up resources

If you used DataSync for a test or don't need the resources you created, delete them so that you don't get charged for resources that you're not using.

To clean up resources

1. Delete tasks that you don't need. For instructions about how to delete a task, see Deleting your DataSync task (p. 112).
2. Delete locations that you don't need. For instructions on how to delete a location, see Deleting a location (p. 97).
3. Delete agents that you don't need. For instructions about how to delete an agent, see Deleting a DataSync agent (p. 63).

Using the AWS Command Line Interface with AWS DataSync

In this section, you can find examples of using the AWS Command Line Interface (AWS CLI) commands for AWS DataSync. You can use these commands to create an agent, create source and destination locations, and run a task.

Before you begin, we recommend reading How AWS DataSync works (p. 3) to understand the components and terms used in DataSync and how the service works. We also recommend reading Using identity-based policies (IAM policies) for DataSync (p. 125) to understand the AWS Identity and Access Management (IAM) permissions that DataSync requires.

Before you use AWS CLI commands, install the AWS CLI. For information about how to install the AWS CLI, see Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*. After you install the AWS CLI, you can use the `help` command to see the DataSync operations and the parameters associated with them.

To see the available operations, enter the following command.

```
aws datasync help
```

To see the parameters associated with a specific operation, enter the following command.

```
aws datasync operation help
```

For more information about the AWS CLI, see What is the AWS Command Line Interface?

Topics

For information about supported AWS Regions and endpoints, see DataSync AWS Regions and endpoints.

For information about DataSync Amazon Resource Name (ARN) values, see DataSync Amazon Resource Names.

Creating an agent

To access your self-managed storage, you first deploy and activate an AWS DataSync agent. The activation process associates your agent with your AWS account. An agent isn't required when transferring between AWS storage services within the same AWS account. To set up a data transfer between two AWS services, see Creating locations (p. 38).

A DataSync agent can transfer data through public service endpoints, Federal Information Processing Standard (FIPS) endpoints, and Amazon VPC endpoints. For more information, see Creating a DataSync agent (p. 55).

Note

When you configure your agent to use Amazon VPC endpoints, the data transferred between your agent and the DataSync service doesn't cross the public internet and doesn't require public IP addresses. For end-to-end instructions for this configuration, see Using AWS DataSync in a virtual private cloud (p. 56).

To create an agent to read from a Network File System (NFS), Server Message Block (SMB), Hadoop Distributed File System (HDFS), or self-managed object storage source location

1. Download the current DataSync `.ova` image or launch the current DataSync Amazon Machine Image (AMI) based on Amazon EC2 from the AWS DataSync console. For information about how to get the `.ova` image or Amazon EC2 AMI, see Create an AWS DataSync agent (p. 20). For information about hardware requirements and recommended Amazon EC2 instance types, see Virtual machine requirements (p. 10).

   Important

   If you are deploying your agent on Amazon EC2, deploy the agent so that it doesn't require network traffic between Availability Zones (to avoid charges for such traffic).

   - To access your Amazon EFS or Amazon FSx for Windows File Server file system, deploy the agent in an Availability Zone that has a mount target to your file system.
   - For self-managed file systems, deploy the agent in the Availability Zone where your file system resides.

   To learn more about data-transfer prices for all AWS Regions, see Amazon EC2 On-Demand pricing.

2. Make sure that you satisfy the network-connectivity requirements for the agent. For information about network requirements, see Network requirements (p. 10).

3. Deploy the `.ova` image in your hypervisor, power on the hypervisor, and note the agent's IP address. Make sure that you can reach the agent on port 80. You can use the following command to check.

```
nc -vz agent-ip-address 80
```

   Note

   The `.ova` default credentials are login **`admin`**, password **`password`**. You can change the password on the virtual machine (VM) local console. You don't need to log in to the VM for basic DataSync functionality. Logging in is required mainly for troubleshooting, network-specific settings, and so on.
   You log in to the agent VM local console by using your VM's hypervisor client. For information about how to use the VM local console, see Working with your DataSync agent's local console (p. 64).

4. Send an HTTP/1.1 GET request to the agent to get the activation key. You can do this by using standard Unix tools:

   - To activate an agent by using a public service endpoint, use the following command.

```
curl "http://agent-ip-address/?gatewayType=SYNC&activationRegion=aws-
region&no_redirect"
```

   - To activate an agent by using a virtual private cloud (VPC) endpoint, use the IP address of the VPC endpoint. Use the following command.

```
curl "http://agent-ip-address/?gatewayType=SYNC&activationRegion=aws-
region&privateLinkEndpoint=IP address of VPC
 endpoint&endpointType=PRIVATE_LINK&no_redirect"
```

To find the correct IP address, open the Amazon VPC console at https://console.aws.amazon.com/vpc/ and choose **Endpoints** from the navigation pane at left. Choose the DataSync endpoint, and check **Subnets list** to find the private IP address that corresponds to the subnet that you chose for your VPC endpoint setup.

For more information about VPC endpoint configuration, see step 5 in Configuring DataSync to use private IP addresses for data transfer (p. 56).

- To activate an agent using a Federal Information Processing Standard (FIPS) endpoint, specify `endpointType=FIPS`. Also, the `activationRegion` value must be set to an AWS Region within the United States. To activate a FIPS endpoint, use the following command.

```
curl "http://agent-IP-address/?gatewayType=SYNC&activationRegion=US-based-aws-
region&endpointType=FIPS&no_redirect"
```

This command returns an activation key similar to the one following.

```
F0EFT-7FPPR-GG7MC-3I9R3-27DOH
```

5. After you have the activation key, do one of the following:

- To activate your agent using a public endpoint or FIPS endpoint, use the following command.

```
aws datasync create-agent \
  --agent-name agent-name \
  --activation-key obtained-activation-key
```

- To activate your agent using a VPC endpoint, use the following command.

```
aws datasync create-agent \
  --agent-name agent-name \
  --vpc-endpoint-id vpc-endpoint-id \
  --subnet-arns subnet-arns \
  --security-group-arns security-group-arns \
  --activation-key obtained-activation-key
```

In this command, use the following arguments:

- *`vpc-endpoint-id`* – The AWS endpoint that the agent connects to. To find the endpoint ID, open the Amazon VPC console at https://console.aws.amazon.com/vpc/, and choose **Endpoints** from the navigation pane on the left. Copy the **Endpoint ID** value of the DataSync endpoint. For more information about VPC endpoint configuration, see step 5 in Configuring DataSync to use private IP addresses for data transfer (p. 56).
- *`security-group-arn`* – The Amazon Resource Names (ARNs) of the security groups to use for the task's endpoint.

  This is the security group that you created in step 3 of Configuring DataSync to use private IP addresses for data transfer (p. 56).
- *`subnet-arns`* – The ARNs of the subnets where the task endpoints for the agent are created.

  This is the subnet that you chose in step 1 of Configuring DataSync to use private IP addresses for data transfer (p. 56).

These commands return the ARN of the agent that you just activated. The ARN is similar to the one following.

```
{
    "AgentArn": "arn:aws:datasync:us-east-1:111222333444:agent/
agent-0b0addbeef44baca3”
}
```

Note
After you choose a service endpoint, you can't change it later.

After you activate the agent, it closes port 80 and the port is no longer accessible. If you can't connect to the agent after you have activated it, verify that the activation was successful by using the following command:

```
aws datasync list-agents
```

Note
Make sure that you are using the same AWS credentials throughout the whole process. Don't switch between multiple terminals where you are authenticated with different AWS credentials.

Creating locations

Each AWS DataSync task is made up of a pair of locations between which data is transferred. The source location defines the storage system or service that you want to read data from. The destination location defines the storage system or service that you want to write data to.

You can work with the following locations:

- Network File System (NFS)
- Server Message Block (SMB)
- Hadoop Distributed File System (HDFS)
- Self-managed object storage source locations
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx for Windows File Server
- Amazon FSx for Lustre
- Amazon FSx for OpenZFS
- Amazon Simple Storage Service (Amazon S3)

For a list of all DataSync supported source and destination endpoints, see Working with AWS DataSync locations (p. 72).

Topics

Creating an NFS location

Use the following procedure to create an NFS location by using the AWS CLI. An NFS location defines a file system on an NFS server that can be read from or written to. You can also create an NFS location by using the AWS Management Console. For more information, see Creating a location for NFS (p. 74).

> **Note**
> If you are using an NFS location on an AWS Snowcone device, see NFS server on AWS Snowcone (p. 75) for more information about transferring data to or from that device.

To create an NFS location by using the CLI

- Use the following command to create an NFS source location.

  ```
  $ aws datasync create-location-nfs \
      --server-hostname nfs-server-address \
      --on-prem-config AgentArns=datasync-agent-arns \
      --subdirectory nfs-export-path
  ```

  For the preceding command, the following applies:

  - The path (*nfs-export-path*) that you provide for the `--subdirectory` parameter must be a path that's exported by the NFS server, or a subdirectory. Other NFS clients in your network must be able to mount this path. To see all the paths exported by your NFS server, run the command `showmount -e` *nfs-server-address* from an NFS client with access to your server. You can specify any directory that appears in the results, and any subdirectory of that directory.
  - To transfer all the data in the folder that you specified, DataSync needs permissions to read all the data. To give DataSync permissions, you can do one of two things. You can configure the NFS export with `no_root_squash`. Or, for the all files that you want DataSync to access, you can make sure that the permissions allow read access for all users. Doing either enables the agent to read the files. For the agent to access directories, you must additionally give all users execute access.
  - Make sure that the NFS export path is accessible without Kerberos authentication.

  DataSync automatically chooses the NFS version that it uses to read from an NFS location. To specify an NFS version, use the optional `Version` parameter in the NfsMountOptions (p. 315) API operation.

This command returns the Amazon Resource Name (ARN) of the NFS location, similar to the ARN shown following.

```
{ "LocationArn": "arn:aws:datasync:us-east-1:111222333444:location/
loc-0f01451b140b2af49" }
```

To make sure that the directory can be mounted, you can connect to any computer that has the same network configuration as your agent and run the following command.

```
mount -t nfs -o nfsvers=<nfs-server-version <nfs-server-address:<nfs-export-path <test-
folder
```

The following is an example of the command.

```
mount -t nfs -o nfsvers=3 198.51.100.123:/path_for_sync_to_read_from /
temp_folder_to_test_mount_on_local_machine
```

Creating an SMB location

Use the following procedure to create an SMB location by using the AWS CLI. An SMB location defines a file system on an SMB server that can be read from or written to. You can also create an SMB location by using the console. For more information, see Creating a location for SMB (p. 75).

To create an SMB location by using the CLI

- Use the following command to create an SMB source location.

  ```
  aws datasync create-location-smb \
      --server-hostname smb-server-address \
      --user user-who-can-mount-share \
      --domain domain-of-the-smb-server \
      --password user-password \
      --agent-arns datasync-agent-arns \
      --subdirectory smb-export-path
  ```

  The *`smb-export-path`* that you provide for the `--subdirectory` parameter should be a path that's exported by the SMB server, or a subdirectory. Specify the path by using forward slashes; for example, `/path/to/folder`. Other SMB clients in your network should be able to access this path.

  DataSync automatically chooses the SMB version that it uses to read from an SMB location. To specify an SMB version, use the optional `Version` parameter in the SmbMountOptions (p. 327) API operation.

This command returns the Amazon Resource Name (ARN) of the SMB location, similar to the ARN shown following.

```
{
    "LocationArn": "arn:aws:datasync:us-east-1:111222333444:location/
loc-0f01451b140b2af49"
}
```

Creating an HDFS location

Use the following procedure to create a Hadoop Distributed File System (HDFS) location by using the AWS CLI. An HDFS location defines a file system on a Hadoop cluster that can be read from or written to. You can also create an HDFS location by using the AWS Management Console. For more information, see Creating a location for HDFS (p. 77).

To create an HDFS location by using the AWS CLI

- Use the following command to create an HDFS location. In the following example, replace each *`user input placeholder`* with your own information.

  ```
  aws datasync create-location-hdfs --name-nodes [{"Hostname":"host1", "Port": 8020}] \
      --authentication-type "SIMPLE|KERBEROS" \
      --agent-arns [arn:aws:datasync:us-east-1:123456789012:agent/
  agent-01234567890example] \
      --subdirectory "/path/to/my/data"
  ```

  The following parameters are required in the `create-location-hdfs` command:

  - `name-nodes` – Specifies the hostname or IP address of the NameNode in the Hadoop cluster and the TCP port that the NameNode is listening on.

- `authentication-type` – The type of authentication to use when connecting to the Hadoop cluster. Specify `SIMPLE` or `KERBEROS`.

  If you use `SIMPLE` authentication, use the `--simple-user` parameter to specify the user name of the user. If you use `KERBEROS` authentication, use the `--kerberos-principal`, `--kerberos-keytab`, and `--kerberos-krb5-conf` parameters. For more information, see create-location-hdfs.
- `agent-arns` – The ARNs of the DataSync agents to use for the HDFS location.

The preceding the command returns the location ARN, similar to the following:

```
{
    "arn:aws:datasync:us-east-1:123456789012:location/loc-01234567890example"
}
```

Creating an object storage location

Use the following procedure to create a self-managed object storage location by using the AWS CLI. An object storage location is the endpoint for an Amazon S3 API-compatible object storage server. An object storage location defines an object storage server that can be read from or written to.

For more information about object storage locations, including compatibility requirements, see Creating a location for object storage (p. 79).

To create a self-managed object storage location by using the CLI

- Use the following command to create a self-managed object storage location.

```
aws datasync create-location-object-storage \
    --server-hostname object-storage-server.example.com \
    --bucket-name myBucket \
    --agent-arns arn:aws:datasync:us-east-1:123456789012:agent/agent-01234567890deadfb
```

The following parameters are required in the `create-location-object-storage` command.

- `server-hostname`: The Domain Name System (DNS) name or IP address of the self-managed object storage server.
- `bucket-name`: The name that identifies the bucket on the self-managed object storage server at the location.
- `agent-arns`: The ARNs of the agents to use for the self-managed object storage location.

If your object storage requires a user name and password to authenticate, use the `--access-key` and `--secret-key` parameters to provide the user name and password, respectively.

The preceding command returns a location ARN similar to the following.

```
{
    "arn:aws:datasync:us-east-1:123456789012:location/loc-01234567890deadfb"
}
```

Creating an Amazon EFS location

Use the following procedure to create an Amazon EFS location by using the AWS CLI. An EFS location is the endpoint for an Amazon EFS file system, which defines an EFS file system that can be read from or written to. You can also create an EFS location by using the console. For more information, see Creating a location for Amazon EFS (p. 81).

To create an Amazon EFS location by using the CLI

1. If you don't have an Amazon EFS file system, create one. For information about how to create an EFS file system, see Getting started with Amazon Elastic File System in the *Amazon Elastic File System User Guide*.
2. Identify a subnet that has at least one mount target for that file system. You can see all the mount targets and the subnets associated with an EFS file system by using the `describe-mount-targets` command.

```
aws efs describe-mount-targets \
    --region aws-region  \
    --file-system-id file-system-id
```

> **Note**
> The AWS Region that you specify is the one where your target S3 bucket or EFS file system is located.

This command returns information about the target similar to the information shown following.

```
{
    "MountTargets": [
        {
            "OwnerId": "111222333444",
            "MountTargetId": "fsmt-22334a10",
            "FileSystemId": "fs-123456ab",
            "SubnetId": "subnet-f12a0e34",
            "LifeCycleState": "available",
            "IpAddress": "11.222.0.123",
            "NetworkInterfaceId": "eni-1234a044"
        }
    ]
}
```

3. Specify an Amazon EC2 security group that can be used to access the mount target. You can run the following command to find out the security group of the mount target.

```
aws efs describe-mount-target-security-groups \
    --region aws-region \
    --mount-target-id mount-target-id
```

The security group that you provide must be able to communicate with the security group on the mount target in the subnet specified.

The relationship between security group M on the mount target and security group S, which you provide for DataSync to use at this stage, is as follows:

- Security group M, which you associate with the mount target, must allow inbound access for the TCP protocol on the NFS port (2049) from security group S.

  You can enable an inbound connection either by its IP address (CIDR range) or its security group.

- Security group S, which you provide to DataSync to access Amazon EFS, should have a rule that enables outbound connections to the NFS port. It enables outbound connections on one of the file system's mount targets.

  You can enable outbound connections either by IP address (CIDR range) or security group.

  For information about security groups and mount targets, see Security groups for Amazon EC2 instances and mount targets in the *Amazon Elastic File System User Guide.*

4. Create the EFS location. To create the EFS location, you need the ARNs for your Amazon EC2 subnet, Amazon EC2 security group, and an EFS file system. Because the DataSync API accepts fully qualified ARNs, you can construct these ARNs. For information about how to construct ARNs for different services, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

   Use the following command to create an EFS location.

```
aws datasync create-location-efs \
    --subdirectory /path/to/your/subdirectory \
    --efs-filesystem-arn 'arn:aws:elasticfilesystem:region:account-id:file-
system/filesystem-id' \
    --ec2-config SecurityGroupArns='arn:aws:ec2:region:account-id:security-
group/security-group-id',SubnetArn='arn:aws:ec2:region:account-id:subnet/subnet-id'
```

   Note
   The AWS Region that you specify is the one where your target S3 bucket or EFS file system is located.

The command returns a location ARN similar to the one shown following.

```
{
    "LocationArn": "arn:aws:datasync:us-west-2:111222333444:location/
loc-07db7abfc326c50fb"
}
```

Creating an Amazon FSx for Windows File Server location

Use the following procedure to create an FSx for Windows File Server location by using the AWS CLI. An Amazon FSx location is the endpoint for an FSx for Windows File Server. This endpoint defines the Amazon FSx file share that you can read from or write to.

You can also create an Amazon FSx location by using the console. For more information, see Creating a location for FSx for Windows File Server (p. 84).

To create an FSx for Windows File Server location by using the AWS CLI

- Use the following command to create an Amazon FSx location.

```
aws datasync create-location-fsx-windows \
    --fsx-filesystem-arn arn:aws:fsx:region:account-id:file-system/filesystem-id \
    --security-group-arns arn:aws:ec2:region:account-id:security-group/group-id \
    --user smb-user --password password
```

  In the `create-location-fsx-windows` command, specify the following:

  - `fsx-filesystem-arn` – The fully qualified Amazon Resource Name (ARN) of the file system that you want to read from or write to.

The DataSync API accepts fully qualified ARNs, and you can construct these ARNs. For information about how to construct ARNs for different services, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

- `security-group-arns` – The ARN of an Amazon EC2 security group that can be applied to the elastic network interfaces (p. 18) of the file system's preferred subnet. For more information, see Creating an Amazon VPC with an instance tenancy of dedicated in the *Amazon EC2 User Guide.*
- The AWS Region – The Region that you specify is the one where your target Amazon FSx file system is located.

The preceding command returns a location ARN similar to the one shown following.

```
{
    "LocationArn": "arn:aws:datasync:us-west-2:111222333444:location/
loc-07db7abfc326c50fb"
}
```

Creating an Amazon FSx for Lustre location

Use the following procedure to create an Amazon FSx for Lustre location by using the AWS CLI. An FSx for Lustre location is an endpoint for an FSx for Lustre file system that you can read or write to.

You can also create an FSx for Lustre location by using the console. For more information, see Creating a location for FSx for Lustre (p. 86).

To create an FSx for Lustre location by using the AWS CLI

- Use the following command to create an FSx for Lustre location.

  ```
  aws datasync create-location-fsx-lustre \
      --fsx-filesystem-arn arn:aws:fsx:region:account-id:file-system:filesystem-id \
      --security-group-arns arn:aws:ec2:region:account-id:security-group/group-id
  ```

  The following parameters are required in the `create-location-fsx-lustre` command.

  - `fsx-filesystem-arn` – The fully qualified Amazon Resource Name (ARN) of the file system that you want to read from or write to.
  - `security-group-arns` – The ARN of an Amazon EC2 security group to apply to the elastic network interfaces of the file system's preferred subnet. For more information, see Dedicated Instances in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

The preceding command returns a location ARN similar to the following.

```
{
    "LocationArn": "arn:aws:datasync:us-west-2:111222333444:location/loc-07sb7abfc326c50fb"
}
```

Creating an Amazon FSx for OpenZFS location

With the AWS CLI, you can create an FSx for OpenZFS location for DataSync to transfer from or to. You also can create an FSx for OpenZFS location in the console (p. 87).

To create an FSx for OpenZFS location by using the AWS CLI

1. Copy the following command:

```
$ aws datasync create-location-fsx-openzfs \
   --fsx-filesystem-arn arn:aws:fsx:region:account-id:file-system/filesystem-id \
   --security-group-arns arn:aws:ec2:region:account-id:security-group/group-id \
   --protocol NFS={}
```

2. Specify the following in the command:

   - For `fsx-filesystem-arn`, specify the location file system's fully qualified Amazon Resource Name (ARN). This includes the AWS Region where your file system resides, your AWS account, and the file system ID.
   - For `security-group-arns`, specify the ARN of the Amazon EC2 security group that provides access to the elastic network interfaces of your FSx for OpenZFS file system's preferred subnet. This includes the AWS Region where your Amazon EC2 instance resides, your AWS account, and the security group ID.

     For more information about security groups, see File System Access Control with Amazon VPC in the *Amazon FSx for OpenZFS User Guide*.
   - For `protocol`, specify the protocol that DataSync uses to access your file system. (DataSync currently supports only NFS.)
3. Run the command. You get a response showing the location that you just created.

```
{
    "LocationArn": "arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890"
}
```

Creating an Amazon S3 location

Use the following procedure to create an Amazon S3 location by using the AWS CLI. An Amazon S3 location requires an Amazon S3 bucket that can be read from or written to. To create an S3 bucket, see Creating a bucket in the *Amazon S3 User Guide*.

For DataSync to access an S3 bucket, DataSync needs an AWS Identity and Access Management (IAM) role that has the required permissions. With the following procedure, you create the IAM role, the required IAM policies, and the S3 location by using the AWS CLI.

For DataSync to assume the IAM role, AWS Security Token Service (AWS STS) must be activated in your account and Region. For more information about temporary security credentials, see Temporary security credentials in IAM in the *IAM User Guide*.

You can also create an S3 location by using the console. For more information, see Creating a location for Amazon S3 (p. 88).

To create an S3 location by using the CLI

1. Create an IAM trust policy that allows DataSync to assume the IAM role required to access your S3 bucket.

   The following is an example of a trust policy.

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

2. Create a temporary file for the IAM policy, as shown in the following example.

```
$ ROLE_FILE=$(mktemp -t sync.iam.role.filename.json)
$ IAM_ROLE_NAME='YourBucketAccessRole'

$ cat<<EOF> ${ROLE_FILE}
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "datasync.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

3. Create an IAM role and attach the IAM policy to it.

   The following command creates an IAM role and attaches the policy to it.

```
$ aws iam create-role --role-name ${IAM_ROLE_NAME} --assume-role-policy-document
 file://${ROLE_FILE}
{
    "Role": {
        "Path": "/",
        "RoleName": "YourBucketAccessRole",
        "RoleId": "role-id",
        "Arn": "arn:aws:iam::account-id:role/YourBucketAccessRole",
        "CreateDate": "2018-07-27T02:49:23.117Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "datasync.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

4. Allow the IAM role that you created to write to your S3 bucket.

   Attach to the IAM role an IAM policy that has sufficient permissions to access your S3 bucket. The following example shows the minimum permissions needed for DataSync to read and write to an S3 bucket in an AWS Region.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn"
        },
        {
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListMultipartUploadParts",
                "s3:PutObjectTagging",
                "s3:GetObjectTagging",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn/*"
        }
    ]
}
```

To attach the policy to your IAM role, run the following command.

```
$ aws iam attach-role-policy \
    --role-name role-name  \
    --policy-arn 'arn:aws:iam::aws:policy/YourPolicyName'
```

For Amazon S3 buckets on AWS Outposts, use the following policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3-outposts:ListBucket",
                "s3-outposts:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": [
                "s3OutpostsBucketArn",
                "s3OutpostsAccessPointArn"
            ],
            "Condition": {
                "StringLike": {
                    "s3-outposts:DataAccessPointArn": "s3OutpostsAccessPointArn"
                }
            }
        },
        {
            "Action": [
                "s3-outposts:AbortMultipartUpload",
                "s3-outposts:DeleteObject",
                "s3-outposts:GetObject",
                "s3-outposts:ListMultipartUploadParts",
                "s3-outposts:PutObjectTagging",
                "s3-outposts:GetObjectTagging",
                "s3-outposts:PutObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "s3OutpostsBucketArn/*",
                "s3OutpostsAccessPointArn"
            ],
            "Condition": {
                "StringLike": {
                    "s3-outposts:DataAccessPointArn": "s3OutpostsAccessPointArn"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3-outposts:GetAccessPoint"
            ],
            "Resource": "s3OutpostsAccessPointArn"
        }
    ]
}
```

5. Create the S3 location.

   Use the following command to create your Amazon S3 location.

```
$ aws datasync create-location-s3 \
    --s3-bucket-arn 'arn:aws:s3:::bucket' \
    --s3-storage-class 'your-S3-storage-class' \
    --s3-config 'BucketAccessRoleArn=arn:aws:iam::account-id:role/role-allowing-DS-
operations' \
    --subdirectory /your-folder
```

   The command returns a location ARN similar to the one shown following.

```
{
    "LocationArn": "arn:aws:datasync:us-east-1:111222333444:location/
loc-0b3017fc4ba4a2d8d"
}
```

   The location type information is encoded in the `LocationUri`. In this example, the `s3://` prefix in `LocationUri` shows the location's type.

   If your Amazon S3 bucket is located on an AWS Outpost, you must deploy an Amazon EC2 agent on your Outpost. The agent must be in a virtual private cloud (VPC) that's allowed to access the access point specified in the command. The agent also must be activated in the parent Region for the Outpost, and be able to route to the Amazon S3 on AWS Outposts endpoints for the bucket. For more information about launching a DataSync agent on AWS Outposts, see Deploy your agent on AWS Outposts (p. 25).

   Use the following command to create an Amazon S3 location on your Outpost.

```
aws datasync create-location-s3 \
    --s3-bucket-arn access-point-arn \
    --s3-config BucketAccessRoleArn=arn:aws:iam::account-id:role/role-allowing-DS-
operations \
    --agent-arns arn-of-datasync-agent-in-vpc-that-can-access-your-s3-acces-point
```

Note

- Changes to object data or metadata are equivalent to deleting an object and creating a new one to replace it. This results in additional charges in the following scenarios:
  - **When using object versioning** – Changes to object data or metadata create a new version of the object.
  - **When using storage classes that can incur additional charges for overwriting, deleting, or retrieving, objects** – Changes to object data or metadata result in such charges. For more information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).
- When you use object versioning, a single DataSync task execution might create more than one version of an Amazon S3 object.
- In addition to the IAM policies that grant DataSync permissions, we recommend creating a multipart upload bucket policy for your S3 buckets. Doing this can help you control your storage costs. For more information, see the blog post S3 lifecycle management update - support for multipart uploads and delete markers.

Creating a task

After you have created an agent and configured your source and destination, you create a task, as described following.

To create a task by using the CLI

1. Create an Amazon CloudWatch Logs log group by using the following command.

```
aws logs create-log-group \
    --log-group-name your-log-group
```

2. Attach an IAM resource policy to your log group. For instructions on how to attach the policy, see Allowing DataSync to upload logs to Amazon CloudWatch log groups (p. 115).
3. Create a task by using the following command.

```
aws datasync create-task \
    --source-location-arn 'arn:aws:datasync:region:account-id:location/location-id' \
    --destination-location-arn 'arn:aws:datasync:region:account-id:location/location-
id' \
    --cloud-watch-log-group-arn 'arn:aws:logs:region:account-id:log-group:log-group' \
    --name task-name
```

   This command returns the Amazon Resource Name (ARN) for a task, similar to the one shown following.

```
{
    "TaskArn": "arn:aws:datasync:us-east-1:111222333444:task/task-08de6e6697796f026"
}
```

   When creating a task that transfers data between AWS services in different Regions, and the other location needs to be specified in a different Region (for example, to transfer data between `us-east-1` and `us-east-2`), use DataSync in one of the Regions and create a task by using the following command.

You can transfer data between AWS Regions, except for the China Regions and the AWS GovCloud (US) Regions. You can also transfer data between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions.

```
aws datasync create-task \
    --source-location-arn 'arn:aws:datasync:us-east-1:account-id:location/location-id \
    --destination-location-arn 'arn:aws:datasync:us-east-2:account-
id:location/location-id \
    --cloud-watch-log-group-arn 'arn:aws:logs:region:account-id' \
    --name task-name \
    --options
 VerifyMode=NONE,OverwriteMode=NEVER,Atime=BEST_EFFORT,Mtime=PRESERVE,Uid=INT_VALUE,Gid=INT_VALUE,P
```

Your task is created with the default configuration options. If you want to configure different options as part of your task creation, add the `--options` parameter to your `create-task` command. The following example shows how to specify different options. For a description of these options, see the section called "Options" (p. 317).

```
aws datasync create-task \
    --source-location-arn 'arn:aws:datasync:region:account-id:location/location-id' \
    --destination-location-arn 'arn:aws:datasync:region:account-id:location/location-
id' \
    --cloud-watch-log-group-arn 'arn:aws:logs:region:account-id:log-group:log-group' \
    --name task-name \
    --options
 VerifyMode=NONE,OverwriteMode=NEVER,Atime=BEST_EFFORT,Mtime=PRESERVE,Uid=INT_VALUE,Gid=INT_VALUE,P
```

When you create a task, you can configure the task to include or exclude specific files, folders, and objects. For more information, see Filtering data transferred by AWS DataSync (p. 104). You can also schedule when you want your task to run. For more information, see Scheduling your DataSync task (p. 107).

Note

If a task remains in the **CREATING** status for more than a few minutes, your agent might be having trouble reaching your self-managed storage. Check the task's `ErrorCode` and `ErrorDetail` values. For example, NFS and SMB mount issues are often caused by a mistyped server hostname, or when the agent's access to your storage is blocked by firewall rules.

Starting a task

When a task execution starts, the task execution changes from **LAUNCHING** to **PREPARING** status within about 10 minutes. The time that the task execution takes to move through its other phases is proportional to the size of your volume. For information about task execution phases, see Task execution (p. 5).

Use the following command to start a task execution.

```
aws datasync start-task-execution \
    --task-arn 'arn:aws:datasync:region:account-id:task/task-id'
```

The command returns a task execution Amazon Resource Name (ARN) similar to the one shown following.

```
{
    "TaskExecutionArn": "arn:aws:datasync:us-east-1:209870788375:task/
task-08de6e6697796f026/execution/exec-04ce9d516d69bd52f"
}
```

You can override the task's options by specifying different options for the current execution, as shown in the example following. For a description of these options, see the section called “Options” (p. 317).

```
aws datasync start-task-execution [...] \
    --override-options \
    --VerifyMode=NONE,OverwriteMode=NEVER,PosixPermissions=NONE
```

When you run a task, you can optionally configure the task to include specific files, folders, and objects to transfer. For more information, see Filtering data transferred by AWS DataSync (p. 104).

> **Note**
> Each agent can run a single task at a time.

Monitoring your task

To monitor the status of your task execution with the CLI, use the `describe-task-execution` command.

```
aws datasync describe-task-execution \
    --task-execution-arn 'arn:aws:datasync:region:account-id:task/task-id/execution/task-
execution-id'
```

This command returns information about a task execution similar to that shown following.

```
{
    "TaskExecutionArn": "arn:aws:datasync:us-east-1:112233445566:task/
task-08de6e6697796f026/execution/exec-04ce9d516d69bd52f",
    "Status": "VERIFYING",
    "Options": {
        "VerifyMode": "POINT_IN_TIME_CONSISTENT",
        "Atime": "BEST_EFFORT",
        "Mtime": "PRESERVE",
        "Uid": "INT_VALUE",
        "Gid": "INT_VALUE",
        "PreserveDevices": "NONE",
        "PosixPermissions": "PRESERVE",
        "PreserveDeletedFiles": "PRESERVE"
        "OverwriteMode": "NEVER",
        "TaskQueueing": "ENABLED"
    },
    "StartTime": 1532658526.949,
    "EstimatedFilesToTransfer": 0,
    "EstimatedBytesToTransfer": 0,
    "FilesTransferred": 0,
    "BytesWritten": 0,
    "BytesTransferred": 0,
    "Result": {
        "PrepareDuration": 4355,
        "PrepareStatus": "Ok",
        "TransferDuration": 5889,
        "TransferStatus": "Ok",
        "VerifyDuration": 4538,
        "VerifyStatus": "Pending"
    }
}
```

If the task execution succeeds, the value of **Status** changes to **SUCCESS**. If the `describe-task-execution` command fails, the result sends error codes that can help you troubleshoot issues. For information about the error codes, see the section called “TaskExecutionResultDetail” (p. 330) in the *DataSync API Reference.*

Monitoring your task in real time

To monitor the progress of your task execution in real time from the command line, use the standard Unix `watch` utility. Task execution duration values are measured in milliseconds.

The `watch` utility doesn't recognize the DataSync alias, so invoke the CLI directly as shown in the following example.

```
# pass '-n 1' to update every second and '-d' to highlight differences
$ watch -n 1 -d \ "aws datasync describe-task-execution --task-execution-arn
 'arn:aws:datasync:region:account-id:task/task-id/execution/task execution-id'"
```

API filters for ListTasks and ListLocations

AWS DataSync supports filters as input arguments to the `ListTasks` and `ListLocations` API calls. This enables you to retrieve configurations of data transfer tasks by using filters such as the source or destination for the data transfer.

Topics

Parameters for API filtering

You can use API filters to narrow down the list of resources returned by `ListTasks` and `ListLocations`. For example, to retrieve all of your Amazon S3 locations, you can use `ListLocations` with the filter name `LocationType` *`S3`* and `Operator` *`Equals`*.

To filter API results, you must specify a filter name, operator, and value.

- `Name` – The name of the filter that's being used. Each API call supports a list of filters that are available for it (for example, `LocationType` for `ListLocations`).
- `Values` – The values that you want to filter for. For example, you might want to display only Amazon S3 locations.
- `Operator` – The operator that's used to compare filter values (for example, `Equals` or `Contains`).

The following table lists the available operators.

Operator	**Key types**
`Equals`	String, Number

Operator	Key types
`NotEquals`	String, Number
`LessThan`	Number
`LessThanOrEqual`	Number
`GreaterThan`	Number
`GreaterThanOrEqual`	Number
`In`	String
`Contains`	String
`NotContains`	String
`BeginsWith`	String

API filtering for ListLocations

`ListLocations` supports the following filter names:

- `LocationType` – Filters on the location type: `SMB`, `NFS`, `HDFS`, `S3`, `FSXW`, `FSXL`, `FSXZ`, and `OBJECT_STORAGE`.
- `LocationUri` – Filters on the uniform resource identifier (URI) assigned to the location, as returned by the `DescribeLocation*` API call (for example, `s3://`*`bucket-name`*`/`*`your-prefix`* for Amazon S3 locations).
- `CreationTime` – Filters on the time that the location was created. The input format is `yyyy-MM-dd:mm:ss` in Coordinated Universal Time (UTC).

The following AWS CLI example lists all locations of type Amazon S3 that have a location URI starting with the string `"s3://DOC-EXAMPLE-BUCKET"` and that were created at or after 2019-12-15 17:15:20 UTC.

```
aws datasync list-locations \
    --filters [{Name=LocationType, Values=["S3"], Operator=Equals},
 {Name=LocationUri, Values=["s3://DOC-EXAMPLE-BUCKET"], Operator=BeginsWith},
 {Name=CreationTime,Values=["2019-12-15 17:15:20"],Operator=GreaterThanOrEqual}]
```

This command returns output similar to the following.

```
{
    "Locations": [
        {
            "LocationArn": "arn:aws:datasync:us-east-1:111122223333:location/
loc-333333333abcdef0",
            "LocationUri": "s3://DOC-EXAMPLE-BUCKET-examples/"
        },
        {
            "LocationArn": "arn:aws:datasync:us-east-1:123456789012:location/
loc-987654321abcdef0",
            "LocationUri": "s3://DOC-EXAMPLE-BUCKET-examples-2/"
        }
    ]
}
```

API filtering for ListTasks

`ListTasks` supports the following filter names.

- `LocationId` – Filters on both source and destination locations on Amazon Resource Name (ARN) values.
- `CreationTime` – Filters on the time that the task was created. The input format is `yyyy-MM-dd:mm:ss` in UTC.

The following AWS CLI example shows the syntax when filtering on `LocationId`.

```
aws datasync list-tasks \
    --filters Name=LocationId,Values=arn:aws:datasync:us-east-1:your-account-id:location/your-location-id,Operator=Contains
```

The output of this command looks similar to the following.

```
{
    "Tasks": [
        {
            "TaskArn": "arn:aws:datasync:us-east-1:your-account-id:task/your-task-id",
            "Status": "AVAILABLE",
            "Name": "DOC-EXAMPLE-BUCKET"
        }
    ]
}
```

Working with agents

An AWS DataSync agent is a virtual machine (VM) that you own and use to read or write data from a self-managed storage system.

Topics

Creating a DataSync agent

Creating an AWS DataSync agent

In general, activate your agent in the AWS Region where the Amazon S3 bucket, Amazon EFS file system, Amazon FSx for Windows File Server file system, Amazon FSx for Lustre file system, or Amazon FSx for OpenZFS file system that you plan to use with AWS DataSync resides. The activation process associates your agent with your AWS account in the most secure way available. An agent can be associated with only one AWS account at a time.

A DataSync agent can communicate with AWS by connecting to one of the following types of endpoints:

- **Public service endpoint** – Data transfers over the public internet.
- **Private virtual private cloud (VPC) endpoint** – Data transfers within your VPC instead of the public internet, increasing the security of the copied data.

  For more information about activating an agent with a private VPC endpoint, see Using AWS DataSync in a virtual private cloud (p. 56).
- **Federal Information Processing Standard (FIPS) endpoint** – Data transfers over the public internet using processes that comply with FIPS.

Your agent is managed by AWS, which automatically updates the agent without interrupting your tasks. To access the agent's local console, see Logging in to the agent local console (p. 64).

For the agent to work properly, make sure that your network is configured properly. For information about network requirements, see Network requirements (p. 10). You can use the virtual machine's (VM's) local console to test for internet connectivity. For more information, see Testing your agent connection to DataSync endpoints (p. 67).

In some cases, an agent is activated but isn't functioning properly. This issue can come from problems with a network partition, firewall misconfiguration, or other events that prevent the agent VM from connecting to AWS. For information about how to troubleshoot connectivity and activation issues, see Testing your agent connection to DataSync endpoints (p. 67).

For instructions on how to create an agent on a VMware ESXi host, see Deploy your agent on VMware (p. 21).

For instructions on how to create an agent on a KVM host, see Deploy your agent on KVM (p. 22).

For instructions on how to create an agent on a Microsoft Hyper-V host, see Deploy your agent on Hyper-V (p. 22).

For instructions on how to create an agent on an Amazon EC2 instance, see Deploy your agent as an Amazon EC2 instance (p. 23).

Using AWS DataSync in a virtual private cloud

You can deploy AWS DataSync in your virtual private cloud (VPC) based on the Amazon VPC service by using VPC endpoints. With this feature, the connection between an agent and the DataSync service doesn't cross the public internet and doesn't require public IP addresses. These connection restrictions increase the security of your data by keeping network traffic within your VPC.

VPC endpoints for DataSync are powered by VPC endpoint services (AWS PrivateLink). AWS PrivateLink is a highly available, scalable AWS service that enables you to privately connect your VPC to supported AWS services. For more information, see VPC endpoint services (AWS PrivateLink) in the *Amazon VPC User Guide*.

To use VPC endpoints, you can transfer files using AWS Direct Connect or a virtual private network (VPN). With this kind of transfer, you use private IP addresses that are accessible only from inside your VPC.

How DataSync works with VPC endpoints

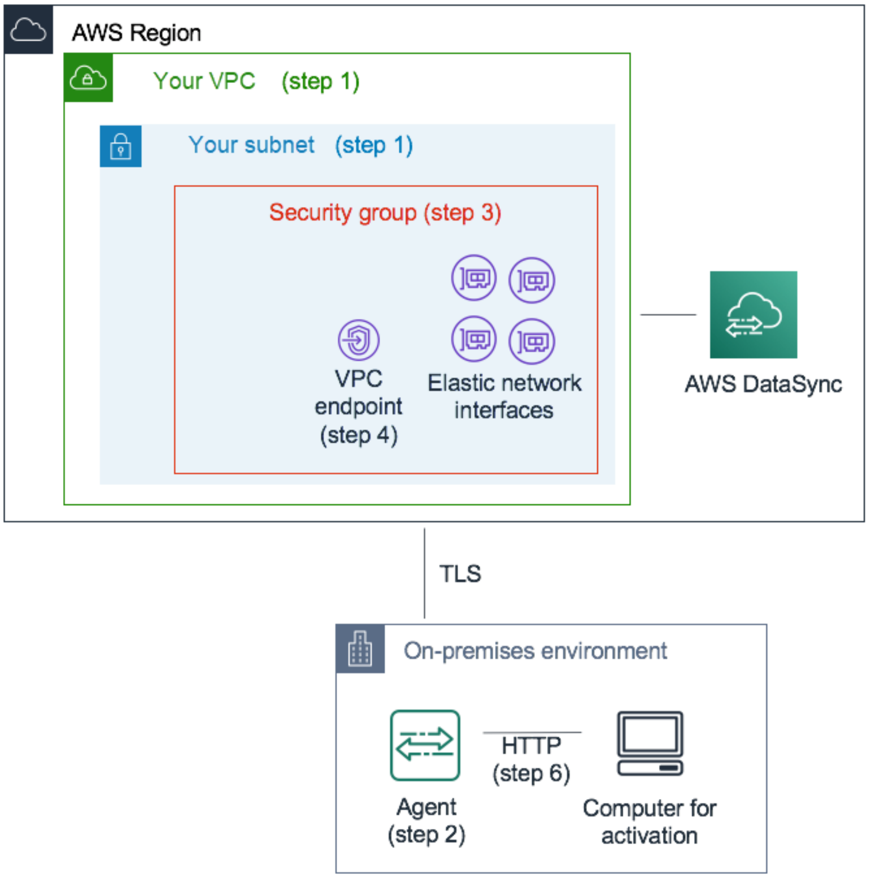
The DataSync agent transfers data between self-managed storage and AWS. You deploy the agent as a virtual machine in the same local network as your source storage. This approach minimizes network overhead associated with transferring data using network protocols such as Network File System (NFS) and Server Message Block (SMB), or when accessing your self-managed object storage using the Amazon S3 API.

When you use DataSync with a private VPC endpoint, the DataSync agent can communicate directly with AWS without the need to cross the public internet.

Configuring DataSync to use private IP addresses for data transfer

In the following procedure, you can find the steps to configure a DataSync agent and a task that communicate with AWS by using VPC endpoints.

The diagram following illustrates the setup process.



To configure a DataSync agent and task to communicate with AWS by using VPC endpoints

1. Choose the VPC and subnet where you want to set up the DataSync private IP addresses.

   The VPC should extend to your local environment, where your SMB, NFS, or self-managed object storage is located, by using routing rules over AWS Direct Connect or VPN. This setup ensures that all communications between the DataSync agent and the DataSync service remain within the VPC.
2. Deploy a DataSync agent close to your local storage. The agent must be able to access your source storage location by using NFS, SMB, or the Amazon S3 API. You can download the .ova file for the DataSync agent from the DataSync console. The agent doesn't need a public IP address. For more information about downloading and deploying an .ova image, see Creating an agent (p. 35).

   Note
   You can use one agent for only one type of endpoint—private, public, or Federal Information Processing Standards (FIPS). If you already have an agent configured for transferring data over the public internet, deploy a new agent to transfer data to private DataSync endpoints. For detailed instructions, see Deploy your DataSync agent (p. 21).
3. In the VPC that you chose in step 1, create a security group to ensure access to the private IP addresses that DataSync uses. These addresses include one VPC endpoint for control traffic and four elastic network interfaces (ENIs) (p. 18) to use for data transfer. You use this security group to manage access to these private IP addresses and ensure that your agent can route to them.

   The agent must be able to establish connections to these IP addresses. In the security group attached to the endpoints, configure inbound rules to allow the agent's private IP address to connect to these endpoints.

4. Create a VPC endpoint for the DataSync service.

   To do this, open the Amazon VPC console at https://console.aws.amazon.com/vpc/, and choose **Endpoints** from the navigation pane at left. Choose **Create Endpoint**.

   For **Service category**, choose **AWS service**. For **Service Name**, choose **DataSync** in your AWS Region (for example, `com.amazonaws.us-east-1.datasync`). Then choose the VPC and security group that you chose in steps 1 and 3. Make sure that you clear the **Enable Private DNS Name** check box.

   > **Important**
   > If you are using a DataSync Amazon EC2 agent, choose the Availability Zone where your agent resides to avoid charges for network traffic between Availability Zones.
   > To learn more about data transfer prices for all AWS Regions, see Amazon EC2 On-Demand pricing.

   For additional details on creating VPC endpoints, see Creating an interface endpoint in *Amazon VPC User Guide*.

5. When your new VPC endpoint becomes available, make sure that the network configuration for your self-managed environment allows agent activation.

   Activation is a one-time operation that securely associates the agent with your AWS account. To activate the agent, use a computer that can reach the agent by using port 80. After activation, this access can be revoked. The agent should be able to reach the private IP address of the VPC endpoint that you created in step 4.

   To find this IP address, open the Amazon VPC console at https://console.aws.amazon.com/vpc/, and choose **Endpoints** from the navigation pane at left. Choose the DataSync endpoint, and check the **Subnets** list for the private IP address for the subnet that you chose. This is the IP address of your VPC endpoint.

   > **Note**
   > Make sure to allow outbound traffic from the agent to the VPC endpoint by using ports 443, 1024–1064, and port 22. Port 22 is optional and is used for the AWS Support channel.

6. Activate the agent. If you have a computer that can route to the agent by using port 80 and that can access the DataSync console, open the console and choose **Create Agent**. In the service endpoint section, choose **VPC endpoints using AWS PrivateLink**.

   Choose the VPC endpoint from step 4, the subnet from step 1, and the security group from step 3. Enter the agent's IP address.

   If you can't access the agent and the DataSync console using the same computer, activate the agent using the command line from a computer that can reach the agent's port 80. For more information, see Creating an agent (p. 35).

7. Choose **Get Key**, optionally enter an agent name and tags, and choose **Create agent**. Your new agent now appears on the **Agents** tab of the DataSync console. The green **VPC Endpoint** banner indicates that all tasks performed with this agent use private endpoints, without crossing the public internet.

8. Create your task by configuring a source and a destination for your data transfer. For more information on choosing endpoints, see Choose a service endpoint for AWS DataSync (p. 25).

   To make transfer easier by using private IP addresses, your task creates four ENIs in the VPC and subnet that you chose.

9. Make sure that your agent can reach the four ENIs and related IP addresses that your task creates.

   To find these IP addresses, open the Amazon EC2 console at https://console.aws.amazon.com/ec2/, and choose **Network Interfaces** on the dashboard. Enter the task ID into the search filter to see the task's four ENIs. These are the ENIs used by your VPC endpoint. Make sure that you allow outbound traffic from the agent to these interfaces by using port 443.

You can now start your task. For each additional task that uses this agent, repeat step 9 to allow the task's traffic through port 443.

Deploying your DataSync agent in AWS Regions

You can use an AWS DataSync agent deployed in an AWS Region to copy data between cloud storage systems and Amazon S3 (including across AWS accounts).

> **Note**
> To transfer files or objects between Amazon S3, Amazon EFS, or Amazon FSx in the same AWS account, you don't need to deploy a DataSync agent. To learn more, see Transferring between AWS storage services (p. 4).

For more information about deploying a DataSync agent in an AWS Region, see Deploy your agent as an Amazon EC2 instance (p. 23).

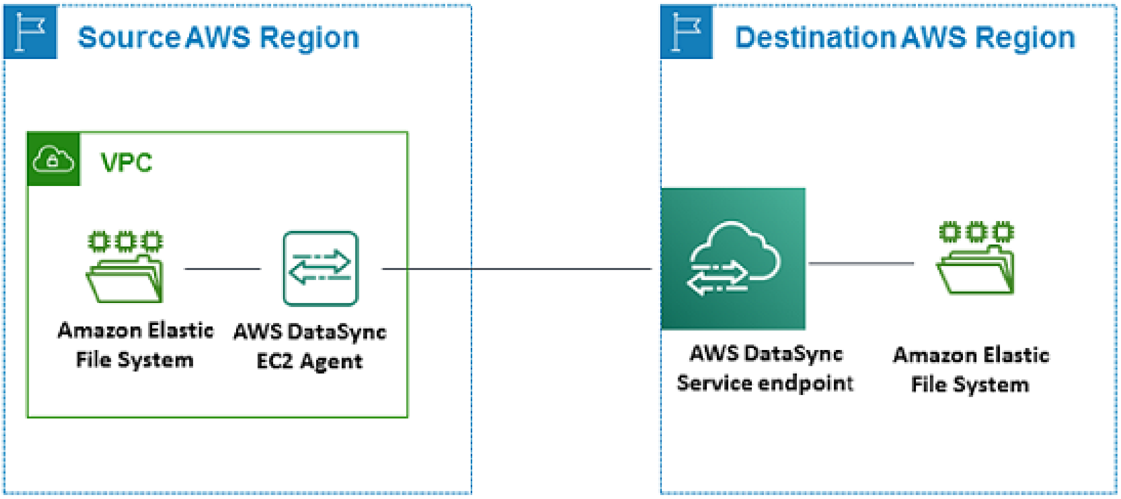
Transferring data from a cloud file system to another cloud file system or Amazon S3

To transfer data from one AWS account to another, or from a cloud file system, the DataSync agent must be located in the same AWS Region and AWS account where the source file system resides. This type of transfer includes the following:

- Transfers between Amazon EFS or FSx for Windows File Server file systems to AWS storage in a different AWS account.
- Transfers from self-managed file systems to AWS storage services.

> **Important**
> Deploy your agent such that it doesn't require network traffic between Availability Zones (to avoid charges for such traffic).
>
> - To access your Amazon EFS or FSx for Windows File Server file system, deploy the agent in an Availability Zone that has a mount target to your file system.
> - For self-managed file systems, deploy the agent in the Availability Zone where your file system resides.
>
> To learn more about data transfer prices for all AWS Regions, see Amazon EC2 On-Demand pricing.

For example, the following diagram shows a high-level view of the DataSync architecture for transferring data from in-cloud Network File System (NFS) to in-cloud NFS or Amazon S3.

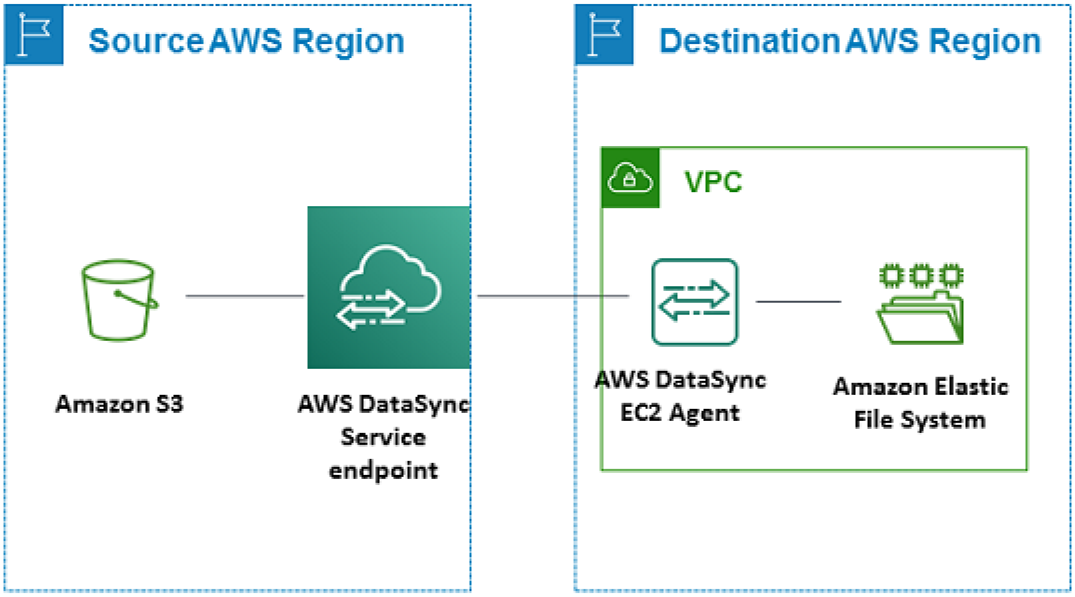

Source AWS Region
VPC
Amazon Elastic File System
AWS DataSync EC2 Agent
Destination AWS Region
AWS DataSync Service endpoint
Amazon Elastic File System

Note
Deploy the agent in the AWS Region and AWS account where the source file system resides.

- When you're copying between two Amazon EFS file systems in different AWS accounts, we recommend that you use the NFS (source) to EFS (destination) transfer.
- When you're copying between two Amazon FSx file systems in different AWS accounts, we recommend that you use the Server Message Block (SMB) (source) to Amazon FSx (destination) transfer.

Data transfer from S3 to in-cloud file systems

The following diagram provides a high-level view of the DataSync architecture for transferring data from Amazon S3 to an in-cloud file system. You can use this architecture to transfer data from one AWS account to another, or to transfer data from Amazon S3 to a self-managed in-cloud file system.



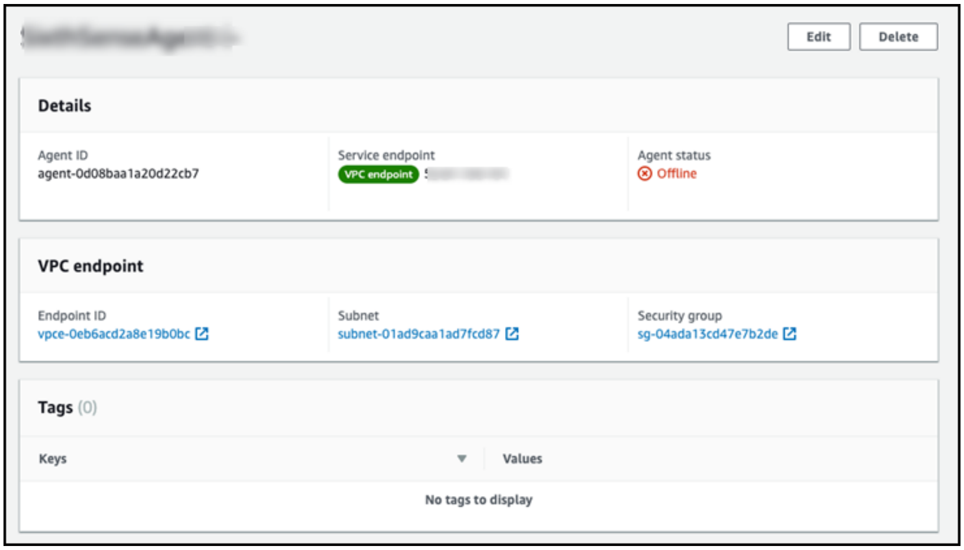
Editing your DataSync agent's properties

You can get detailed information about your agent and edit the agent's properties on the agent's details page.

To edit your agent's properties

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the left navigation pane, choose **Agent** to open the **Agents** page.
3. In the **Agent ID** column, choose the agent that you want to edit.

   For agents activated with a private virtual private cloud (VPC) endpoint, details about the VPC endpoint display.



[redacted]

Details

Agent ID
agent-0d08baa1a20d22cb7

Service endpoint
VPC endpoint [redacted]

Agent status
⊗ Offline

VPC endpoint

Endpoint ID
vpce-0eb6acd2a8e19b0bc

Subnet
subnet-01ad9caa1ad7fcd87

Security group
sg-04ada13cd47e7b2de

Tags (0)

Keys	Values
No tags to display	

4. Choose **Edit** and make the changes that you want.

   Note
   You can't change the agent type.

Using multiple AWS DataSync agents for a location

For most workloads, we recommend that you use one AWS DataSync agent for each self-managed location. However, there are exceptions:

- Some workloads have tens of millions of small files. In these cases, we recommend up to four agents for each location.
- If your network has limited bandwidth (for example, an agent is on a network link with less than 2.5 Gbps), we recommend four agents for each location.

When using multiple agents for a location, remember the following:

- All the agents must be online to run your DataSync task.

  Note
  If even one of the agents goes offline, you can't use the location in a task.
- If you're using a VPC endpoint to communicate with AWS, all the agents must use the same endpoint and subnet.

DataSync agent statuses

The following table describes each agent status, and if and when you should take action based on the status.

Agent status	Meaning
ONLINE	The agent is configured properly and is available to use. The ONLINE status is the normal running status for an agent.

Agent status	Meaning
OFFLINE	The agent's virtual machine (VM) is turned off or the agent is in an unhealthy state and has been out of contact with the service for five minutes or longer. When the issue that caused the unhealthy state is resolved, the agent returns to ONLINE status.

Deleting a DataSync agent

When you delete a DataSync agent, it's no longer associated with your AWS account and can't be undone.

> **Note**
> Deleting doesn't remove the agent's virtual machine (VM) from your environment. You can reuse the VM to create and activate a new agent.

To delete an agent

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the left navigation pane, choose **Agents**.
3. Choose the agent that you want to delete.
4. Choose **Delete**, enter **`delete`** in the text box that appears, and then choose **Delete**.

To create and activate an agent on a VM or Amazon EC2 instance after deleting an agent

1. Delete the old agent (see the preceding steps for instructions). Do not delete the VM or Amazon EC2 instance.
2. Wait until the old agent is deleted and the VM is ready to be activated, usually about three minutes. Alternatively, you can verify that the agent has been deleted by checking the status of port 80. When the VM is ready to be activated, port 80 will be open.
3. Create and activate a new DataSync agent on the existing VM or Amazon EC2 instance. For information about creating a DataSync agent, see Creating a DataSync agent (p. 55). The new agent can be activated in a different AWS Region, depending on network connectivity.

Configuring your DataSync agent for multiple NICs

If you configure your agent to use multiple network adapters (NICs), the agent can be accessed by more than one IP address. You might want to do this in the following situations:

- **Maximizing throughput** – You might want to maximize throughput to an agent when network adapters are a bottleneck.
- **Network isolation** – Your Network File System (NFS), Server Message Block (SMB), Hadoop Distributed File System (HDFS), or object storage server might reside on a virtual LAN (VLAN) that lacks internet connectivity for security reasons.

In a typical multiple-adapter use case, one adapter is configured as the route by which the agent communicates with AWS (as the default agent). Except for this one adapter, NFS, SMB, HDFS, or self-managed object storage locations must be in the same subnet as the adapter that connects to them. Otherwise, communication with the intended NFS, SMB, HDFS, or object storage locations might not be possible. In some cases, you might configure an NFS, SMB, HDFS, or object storage location on the same adapter that's used for communication with AWS. In these cases, NFS, SMB, HDFS, or object storage traffic for that server and AWS traffic flows through the same adapter.

In some cases, you might configure one adapter to connect to the AWS DataSync console and then add a second adapter. In such a case, DataSync automatically configures the route table to use the second adapter as the preferred route.

Working with your DataSync agent's local console

If you deployed your AWS DataSync agent on-premises, you can troubleshoot issues with the agent using the VM's local console. For example, you'll log in to the console if you need to run a connectivity test or open a support channel with AWS.

> **Note**
> You don't need to use the agent's local console for standard DataSync functionality.

Topics

Logging in to the agent local console

For security reasons, you can't remotely connect to the local console of the DataSync agent VM.

To log in to the agent's local console

- If this is your first time using the local console, log in with the default credentials. The default user name is **`admin`** and the password is **`password`**. Otherwise, use your credentials to log in.

  > **Note**
  > We recommend changing the default password. You do this by running the `passwd` command from the local console menu. (Item **`5`** on the main menu opens the command prompt. For VMware VMs, choose item **`6`**.) For information about how to run the command, see Running AWS DataSync commands on the local console (p. 69).

Obtaining an activation key using the local console

If your agent isn't activated yet, you can obtain its activation key from the local console. This option is displayed only until the agent has been activated.

To get an activation key for your agent from the local console

1. Log in to your agent's local console.

2. On the **AWS DataSync Activation - Configuration** main menu, enter `0` to get an activation key.
3. Enter the AWS Region that your agent will be activated in.
4. Enter the service endpoint type that your agent will be using. Options include public, Federal Information Processing Standard (FIPS), and virtual private cloud (VPC) with AWS PrivateLink.
5. The activation key is automatically generated and displayed on screen. Select and copy this value.
6. Using the activation key copied from the last step, use the following CLI command to create and activate the agent:

```
$ aws datasync create-agent --agent-name your-new-agent-name --activation-
key generated-activation-key
```

   On successful activation, this command returns something similar to the following.

```

{
"AgentArn": "arn:aws:datasync:us-west-1:1234567890A:agent/agent-ID"
}
```

   You can also insert the activation key in the DataSync console using the agent creation wizard.

   After the agent is activated, the console menu displays the **Agent ID** and **AWS Region**. The option for getting an activation key is no longer visible in the console menu.

Configuring your agent network settings

The default network configuration for the agent is Dynamic Host Configuration Protocol (DHCP). With DHCP, your agent is automatically assigned an IP address. In some cases, you might need to manually assign your agent's IP as a static IP address, as described following.

To configure your agent to use static IP addresses

1. Log in to your agent's local console
2. On the **AWS DataSync Activation - Configuration** main menu, enter `1` to begin configuring your network.
3. On the **Network Configuration** menu, choose one of the following options.

To	Do this
Get information about your network adapter	Enter `1`. A list of adapter names appears, and you are prompted to enter an adapter name—for example, **`eth0`**. If the adapter you specify is in use, the following information about the adapter is displayed: • Media access control (MAC) address • IP address • Netmask • Agent IP address • DHCP enabled status

To	Do this
	You use the same adapter name when you configure a static IP address (option **3**) as when you set your agent's default route adapter (option **5**).
Configure DHCP	Enter **`2`**. You are prompted to configure the network interface to use DHCP.
Configure a static IP address for your agent	Enter **`3`**. You are prompted to enter the Network adapter name. **Important** If your agent has already been activated, you must shut it down and restart it from the DataSync console for the settings to take effect.
Reset all your agent's network configuration to DHCP	Enter **`4`**. All network interfaces are set to use DHCP. **Important** If your agent has already been activated, you must shut down and restart your agent from the DataSync console for the settings to take effect.
Set your agent's default route adapter	Enter **`5`**. The available adapters for your agent are shown, and you are prompted to choose one of the adapters—for example, **`eth0`**.
Edit your agent's Domain Name System (DNS) configuration	Enter **`6`**. The available adapters of the primary and secondary DNS servers are displayed. You are prompted to provide the new IP address.
View your agent's DNS configuration	Enter **`7`**. The available adapters of the primary and secondary DNS servers are displayed. **Note** For some versions of the VMware hypervisor, you can edit the adapter configuration in this menu.
View routing tables	Enter **`8`**. The default route of your agent is displayed.

Testing your agent connection to DataSync endpoints

You can use your agent's local console to test your internet connection. This test can be useful when you are troubleshooting network issues with your agent.

To test your agent's connection to AWS DataSync endpoints

1. Log in to your agent's local console.
2. On the **AWS DataSync Activation - Configuration** main menu, enter **2** to begin testing network connectivity.
3. Enter the service endpoint type that your agent is connecting to. Valid endpoint types include public, FIPS, and VPC endpoints using AWS PrivateLink.

   When the agent is activated, the **Test Network Connectivity** option can be initiated without any additional user input, because the Region and endpoint type are taken from the activated agent information.

   a. To test public endpoint connectivity, enter **1**, followed by the AWS Region in which your agent is activated. Connectivity test results against the correct endpoints for your agent's Region are displayed. For information about AWS Regions and endpoints, see Where can I use DataSync? (p. 8).

      Each endpoint in the selected AWS Region displays either a **PASSED** or **FAILED** message.
   b. To test FIPS endpoint connectivity, enter **2**, followed by the AWS Region in which your agent is activated. Connectivity test results against the correct endpoints for your agent's Region are displayed. For information about AWS Regions and endpoints, see Where can I use DataSync? (p. 8).

      Each endpoint in the selected AWS Region displays either a **PASSED** or **FAILED** message.
   c. To test VPC connectivity, enter **3**. Network connectivity test results for your agent's VPC endpoints are displayed.

      Each VPC endpoint displays either a **PASSED** or **FAILED** message.

For information about network and firewall requirements, see Network requirements (p. 10).

Testing connectivity to storage systems

You can use the console to test connectivity to storage systems involved in your transfer, including Network File System (NFS), Server Message Block (SMB), Hadoop Distributed File System (HDFS), or object storage servers.

To test connectivity to storage systems

1. Log in to your agent's local console.
2. On the **AWS DataSync Activation - Configuration** main menu, enter **3** to begin network testing.
3. Choose the location type you're testing using one of the following options.

   a. Enter **1** to test an NFS server connection.
   b. Enter **2** to test an SMB server connection.
   c. Enter **3** to test an object storage server connection.
   d. Enter **4** to test an HDFS connection.
4. Enter the IP address or server domain name of the storage server.

For HDFS, enter the IP address or hostname of the NameNode or DataNode in the Hadoop cluster, followed by the TCP port number.

Connectivity test results, either **PASSED** or **FAILED**, are displayed for the specified server, along with the IP address and port of the tested server.

Viewing your agent system resource status

When you log in to your agent console, virtual CPU cores, root volume size, and RAM are automatically checked. If there are any errors or warnings, they're flagged on the console menu display with a banner that provides details about those errors or warnings.

If there are no errors or warnings when the console starts, the menu displays white text. The **View System Resource Check** option will display `(0 Errors)`.

If there are errors or warnings, the console menu displays the number of errors and warnings, in red and yellow respectively, in a banner across the top of the menu. For example, `(1 ERROR, 1 WARNING)`.

To view the status of a system resource check

1. Log in to your agent's local console.
2. On the **AWS DataSync Activation - Configuration** main menu, enter **4** to view the results of the system resource check.

   The console displays an **[OK]**, **[WARNING]**, or **[FAIL]** message for each resource as described in the table following.

   For Amazon EC2 instances, the system resource check verifies that the instance type is one of the instances recommended for use with DataSync. If the instance type matches that list, a single result is displayed in green text, as follows.

   ```
   [ OK ] Instance Type Check
   ```

   If the Amazon EC2 instance is not on the recommended list, the system resource check verifies the following resources.

   - CPU cores check: At least four cores are required.
   - Disk size check: A minimum of 80 GB of available disk space is required.
   - RAM check: A minimum of 32 GiB of RAM is required for up to 20 million file transfers per task. A minimum of 64 GiB of RAM is required for more than 20 million file transfers per task.
   - CPU flags check: The agent VM CPU must have either SSSE3 or SSE4 instruction set flags.

   If the Amazon EC2 instance is not on the list of recommended instances for DataSync, but it has sufficient resources, the result of the system resource check displays four results, all in green text.

   The same resources are verified for agents deployed in Hyper-V, Linux Kernel-based Virtual Machine (KVM), and VMware VMs.

   VMware agents are also checked for supported version; unsupported versions trigger a red banner error. Supported versions include VMware versions 6.5 and 6.7.

Configuring a Network Time Protocol (NTP) server for VMware agents

If you are using a VMware VM, you can view Network Time Protocol (NTP) server configurations and synchronize the VM time on your agent with your VMware hypervisor host.

To manage system time

1. Log in to your agent's local console.
2. On the **AWS DataSync Activation - Configuration** main menu, enter **5** to manage your system's time.
3. On the **System Time Management** menu, enter **1** to view and synchronize the VM system time.

To	Do this
View and synchronize your VM time with NTP server time	Enter **1**. The current time of your agent is displayed. Your agent determines the time difference between your agent VM and your NTP server time, and prompts you to synchronize the agent time with NTP time. After your agent is deployed and running, in some scenarios the agent's time can drift. For example, suppose that there is a prolonged network outage and your hypervisor host and agent don't get time updates. In this case, the agent's time is different from the true time. When there is a time drift, a discrepancy occurs between the stated times when operations such as snapshots occur and the actual times that the operations occur.
Edit your NTP server configuration	Enter **2**. You are prompted to provide a preferred and a secondary NTP server.
View your NTP server configuration	Enter **3**. Your NTP server configuration is displayed.

Running AWS DataSync commands on the local console

The VM local console in AWS DataSync helps provide a secure environment for configuring and diagnosing issues with your agent. Using the local console commands, you can perform maintenance tasks such as saving routing tables, connecting to AWS Support, and so on.

To run a configuration or diagnostic command

1. Log in to your agent's local console.
2. On the **AWS DataSync Activation - Configuration** main menu, enter **5** for **Command Prompt**.

> **Note**
> If you are using a VMware VM, enter **6** for the **Command Prompt**.

3. The commands available to be used through the console include the following.

Use this command	To do this
`ip`	Display or configure routing, devices, and tunnels
`save-routing-table`	Save a newly added routing table entry
`ifconfig`	Display or configure network interfaces
`iptables`	Administer IPv4 packet filtering and network address translation (NAT)
`save-iptables`	Persist IP tables
`dig`	Perform DNS lookup for DNS hostname
`open-support-channel`	Connect to AWS Support
`h`	Display available command list
`exit`	Return to console configuration menu

4. At the command prompt, enter the command that you want to use and follow the instructions.

Enabling AWS Support to help troubleshoot your running agent

You can allow AWS Support to access your AWS DataSync agent and assist you with troubleshooting agent issues. By default, AWS Support access to DataSync is disabled. You enable this access through the host's local console. To give AWS Support access to DataSync, you first log in to the local console for the host and then connect to the support server.

To log in to an agent running on Amazon EC2, create a rule for the instance's security group that opens TCP port 22 for Secure Shell (SSH) access.

> **Note**
> If you add a new rule to an existing security group, the new rule applies to all instances that use that security group. For more information about security groups and how to add a security group rule, see Amazon EC2 security groups for Linux instances in the *Amazon EC2 User Guide for Linux Instances.*

To enable AWS Support access to AWS DataSync

1. Log in to your host's local console.

   If this is your first time logging in to the local console, see Logging in to the agent local console (p. 64).
2. At the prompt, enter **5** to open the command prompt (for VMware VMs, use **6**).
3. Enter **h** to open the **AVAILABLE COMMANDS** window.
4. In the **AVAILABLE COMMANDS** window, enter the following to connect to AWS Support:

   ```
   open-support-channel
   ```

If you are using the agent with VPC endpoints, you must provide a VPC endpoint IP address for your support channel, as follows:

```
open-support-channel vpc-ip-address
```

Your firewall must allow the outbound TCP port 22 to initiate a support channel to AWS. When you connect to AWS Support, DataSync assigns you a support number. Make a note of your support number.

> **Note**
> The channel number is not a Transmission Control Protocol/User Datagram Protocol (TCP/UDP) port number. Instead, it makes a Secure Shell (SSH) (TCP 22) connection to servers and provides the support channel for the connection.

5. When the support channel is established, provide your support service number to AWS Support so that they can provide troubleshooting assistance.
6. When the support session is finished, press **`Enter`** to end it.
7. Enter **`exit`** to log out of the DataSync local console.
8. Follow the prompts to exit the local console.

Working with AWS DataSync locations

In this section, you can find information about how to create and configure locations. A *location* defines the storage system or service that you want to read data from or write data to. AWS DataSync supports the following location types:

- Network File System (NFS)
- Server Message Block (SMB)
- Hadoop Managed File System (HDFS)
- Object storage systems
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx for Windows File Server
- Amazon FSx for Lustre
- Amazon FSx for OpenZFS
- Amazon S3

When you create a task that transfers data between AWS services in different AWS Regions, one of the two locations that you specify must reside in the Region where DataSync is being used. The other location must be specified in a different Region.

You can transfer data between AWS Regions, except for the China Regions and the AWS GovCloud (US) Regions. You can also transfer data between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions.

DataSync supports the following source and destination location combinations.

> **Note**
> AWS Regions indicates Regions other than the China Regions and the AWS GovCloud (US) Regions. Transfers involving the AWS GovCloud (US) Regions can only be between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions.

Source (from)	Destination (to)
NFS, SMB, HDFS, object storage, or NFS on AWS Snowcone	Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS
Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS	NFS, SMB, HDFS, object storage, or NFS on AWS Snowcone
Amazon S3 (in AWS Regions), Amazon EFS,	Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS

Source (from)	Destination (to)
FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS	
Amazon S3 (in AWS Regions)	Amazon S3 on AWS Outposts
Amazon S3 on AWS Outposts	Amazon S3 (in AWS Regions)

In addition, you can use the following combinations to transfer data between managed file systems and Amazon S3 buckets in different AWS accounts. When these kinds of transfers only involve Amazon EFS or supported Amazon FSx file systems, you must use a DataSync agent.

Source (from)	Destination (to)
Amazon EFS (configured as an NFS location) or FSx for Windows File Server (configured as an SMB location)	Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS
Amazon S3 (in AWS Regions)	Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS
Amazon S3 (in AWS Regions), Amazon EFS, FSx for Windows File Server, FSx for Lustre, or FSx for OpenZFS	Amazon S3 (in AWS Regions)
NFS, SMB, HDFS, or object storage	Amazon S3 (in AWS Regions)

> **Important**
> When you use DataSync to copy files or objects between AWS Regions, you pay for data transfer between Regions. This transfer is billed as data transfer OUT from your source Region to your destination Region. For more information, see Data transfer pricing.

Topics

- Deleting a location (p. 97)

Creating a location for NFS

AWS DataSync supports the Network File System (NFS) v3, NFS v4.0, and NFS v4.1 protocols.

To create an NFS location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**. The locations that you previously created appear in the list of locations.
3. On the **Locations** page, choose **Create location**.
4. For **Location type**, choose **NFS**. You configure this location as a source or destination later.
5. For **Agents**, choose the agent that you want to use. If you have previously created agents, the agents appear in the list. The agent connects to your self-managed NFS server and makes it easier to securely transfer data between the self-managed location and AWS.
6. For **NFS Server**, provide the Domain Name System (DNS) name or IP address of the NFS server.
7. For **Mount path**, enter the mount path for your NFS location.
8. (Optional) Expand **Additional settings** and choose a specific **NFS Version** to use. By default, DataSync uses NFS 4.1.
9. (Optional) Select **Add tag** to create tags for your NFS location. A *tag* is a key-value pair that helps you manage, filter, and search for your locations.
10. When you're done, choose **Create location**.

For detailed information about these NFS location settings, see NFS location settings (p. 74).

NFS location settings

Following, you can find descriptions for the configuration settings for NFS locations in DataSync.

Agent

An *agent* is a VM that's deployed in your self-managed environment to connect to your self-managed location. An agent makes it easier to securely transfer data between the self-managed location and AWS. You can use an agent for more than one location.

If a task is using multiple agents, all the agents must have the status **Available** for the task to run. If you use multiple agents for a source location, the status of all the agents must be **Available** for the task to run. Agents are automatically updated by AWS on a regular basis, using a mechanism that doesn't interrupt your tasks.

NFS server

The name of the NFS server, the IP address, or DNS name of the NFS server. An agent that's installed on-premises uses this name to mount the NFS server in a network.

Mount path

The mount path for your NFS file system. This path must be a path that's exported by the NFS server, or a subdirectory of an exported path. This path should be such that it can be mounted by other NFS clients in your network. For information about how to resolve mount path issues, see My task status is unavailable and indicates a mount error (p. 142).

To transfer all the data in the folder you specified, DataSync must have permissions to read all the data. To ensure this, either configure the NFS export with `no_root_squash`, or ensure that the permissions of the files that you want DataSync to transfer allow read access for all users. Doing either enables the agent to read the files. For the agent to access directories, you must additionally enable all execute access.

Tag

A *tag* is a key-value pair that helps you manage, filter, and search for your location. Adding a tag is optional. We recommend using tags for naming your resources.

> **Note**
> DataSync supports the NFS v3, NFS v4.0, and NFS v4.1 protocols. DataSync automatically chooses the NFS version that it uses when reading from an NFS location. If you need to force DataSync to use a specific NFS version, see I need DataSync to use a specific NFS or SMB version to mount my share (p. 141).

NFS server on AWS Snowcone

If you are copying data to or from your AWS Snowcone device, note the following configuration.

- **Agents**: Select the Amazon EC2 agent that you launched on your AWS Snowcone device. For more information about using DataSync with Snowcone, see Using DataSync to transfer files to AWS in the *AWS Snowcone User Guide*.
- **NFS server**: Specify the virtual IP address that you attached to the NFS server on your Snowcone device using the AWS OpsHub for Snow Family. For more information about using AWS OpsHub, see Using AWS OpsHub for Snow Family to manage devices.
- **Mount path**: Specify the NFS export path for the bucket you want to transfer data to or from. The format of the export path of an Amazon S3 bucket is `/buckets/bucket-name`. For more information about using the AWS Snowcone NFS server, see Using NFS file shares to manage file storage in the *AWS Snowcone User Guide*.

Creating a location for SMB

A *location* is an endpoint for a Server Message Block (SMB) file share, which can be hosted on-premises or by another cloud provider (for example, Azure Files).

Creating the location

Your SMB file share can be a source or destination location for AWS DataSync.

To create an SMB location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**.
3. On the **Locations** page, choose **Create location**.
4. For **Location type**, choose **Server Message Block (SMB)**.

   You configure this location as a source or destination later.

5. For **Agents**, choose the agent that you want to use.

   The agent connects to your SMB server and makes it easier to securely transfer data between the self-managed location and AWS.
6. For **SMB Server**, provide the Domain Name System (DNS) name or IP address of the SMB server.
7. For **Share name**, enter the name of the share exported by your SMB server.

   You can include a folder from within this share. Specify the share by using slashes (for example, `/path/to/folder`).
8. (Optional) Expand **Additional settings** and choose a specific **SMB Version** to use.

   You can also let DataSync automatically choose a version based on negotiation with the SMB server.
9. For **User**, enter the user who can mount the location and has the permissions to access the SMB server.

   For more information, see User to understand how to choose a user with sufficient permissions to files, folders, and metadata.
10. For **Password**, enter the password of the user who can mount the location and has the permissions to access the SMB server.
11. (Optional) For **Domain**, enter the domain that the SMB server belongs to.
12. (Optional) Select **Add tag** to create tags for your SMB location.

    A *tag* is a key-value pair that helps you manage, filter, and search for your locations.
13. Choose **Create location**.

Understanding the location settings

Agent

An *agent* is a VM that is deployed in your on-premises environment to connect to your self-managed location. An agent makes it easier to securely transfer data between the self-managed location and AWS. You can use an agent for more than one location.

If a task is using multiple agents, all the agents must have the status **Available** for the task to run. If you use multiple agents for a source location, the status of all the agents must be **Available** for the task to run. Agents are automatically updated by AWS on a regular basis, using a mechanism that doesn't interrupt your tasks.

SMB server

The name of the SMB server, the IP address, or DNS name of the SMB server. An agent that is installed on-premises uses this name to mount the SMB server in a network.

Share name

The name of the share exported by your SMB server. You can include a folder from within this share. Specify the share by using slashes (for example `/path/to/folder`).

SMB version

DataSync supports SMB 2.1 and SMB 3 for mounting an SMB file share. DataSync can automatically choose a version based on negotiation with the SMB server.

User

The name of user who can mount the location and has the permissions to access the SMB file share. This user can be a local user on your Windows file server or a domain user defined in your Active Directory.

To set object ownership, DataSync requires the `SE_RESTORE_NAME` privilege, which is usually granted to members of the built-in Active Directory groups **Backup Operators** and **Domain Admins**. Providing a user to DataSync with this privilege also helps ensure sufficient permissions to files, folders, and file metadata except for NTFS system access control lists (SACLs).

Additional privileges are required to copy SACLs. Specifically, this requires the Windows `SE_SECURITY_NAME` privilege, which is granted to members of the **Domain Admins** group. If your task will be configured to copy SACLs, make sure that the user has the required privileges. To learn more about configuring a task to copy SACLs, see Ownership and permissions-related options (p. 31).

When you copy data between SMB shares and FSx for Windows File Server locations or between two FSx for Windows File Server locations, your source and destination locations must either:

- Belong to the same Active Directory domain.
- Have an Active Directory trust relationship between their domains.

Password

The password of the user who can mount the location and has the permissions to access files and folders in the SMB file share.

Domain

The name of the domain that the user is part of.

Tag

A *tag* is a key-value pair that helps you manage, filter, and search for your location. Adding a tag is optional. We recommend using tags for naming your resources.

Creating a location for HDFS

To connect to your Hadoop Distributed File System (HDFS) cluster, AWS DataSync uses an agent. The agent is a virtual machine that you deploy near your HDFS cluster. To learn more about DataSync agents, see Working with agents (p. 55). The DataSync agent acts as an HDFS client and communicates with the NameNodes and DataNodes in your clusters.

When you start a task, DataSync queries the NameNode for locations of files and folders on the cluster. If the HDFS location is configured as a source, then DataSync reads files and folder data from the DataNodes in the cluster and copies the data to the destination. If the HDFS location is configured as a destination, then DataSync writes files and folders from the destination to the DataNodes in the cluster. Before running your DataSync task, verify agent connectivity to the HDFS cluster. For more information, see Testing connectivity to storage systems (p. 67).

Authentication

When connecting to an HDFS cluster, DataSync supports simple authentication or Kerberos authentication. To use simple authentication, provide the user name of a user with rights to read and write to the HDFS cluster. To use Kerberos authentication, provide a Kerberos configuration file, a Kerberos key table (keytab) file, and a Kerberos principal name. The credentials of the Kerberos principal must be in the provided keytab file.

Encryption

When using Kerberos authentication, DataSync supports encryption of data as it's transmitted between the DataSync agent and your HDFS cluster. Encrypt your data by using the Quality of Protection (QOP) configuration settings on your HDFS cluster and by specifying the QOP settings when creating your HDFS location. The QOP configuration includes settings for data transfer protection and Remote Procedure Call (RPC) protection.

DataSync supports the following Kerberos encryption types:

- des-cbc-crc
- des-cbc-md4
- des-cbc-md5
- des3-cbc-sha1
- arcfour-hmac
- arcfour-hmac-exp
- aes128-cts-hmac-sha1-96
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha384-192
- camellia128-cts-cmac
- camellia256-cts-cmac

You can also configure HDFS clusters for encryption at rest using Transparent Data Encryption (TDE). When using simple authentication, DataSync reads and writes to TDE-enabled clusters. If you're using DataSync to copy data to a TDE-enabled cluster, first configure the encryption zones on the HDFS cluster. DataSync doesn't create encryption zones.

> **Note**
> Before creating your HDFS location, verify network connectivity between your agent and your Hadoop cluster. Test access to the TCP ports listed in the Network requirements to connect to your self-managed storage (p. 10) table. To test access between your local agent and your Hadoop cluster, follow the procedure in Testing connectivity to storage systems (p. 67).

To create an HDFS location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. On the **Locations** page, choose **Create location**.
3. For the **Location type**, choose **Hadoop Distributed File System (HDFS)**. You can configure this location as a source or destination later.
4. For **Agents**, choose one or more agents that you want to use from the list of available agents. The agent connects to your HDFS cluster to securely transfer data between the HDFS cluster and DataSync.
5. For **NameNode**, provide the domain name or IP address of the HDFS cluster's primary NameNode.
6. For **Folder**, enter a folder on your HDFS cluster that DataSync will use for the data transfer. When the location is used as a source for a task, DataSync copies files in the provided folder. When your location is used as a destination for a task, DataSync writes all files to the provided folder.
7. To set the **Block size** or **Replication factor**, choose **Additional settings**. The default block size is 128 MiB, and any provided block sizes must be a multiple of 512 bytes. The default replication factor is three DataNodes when transferring data to the HDFS cluster.

8. In the **Security** section, choose the **Authentication type** used on your HDFS cluster.
    - **Simple**: Provide the user name of the **User** with read and write permissions on the HDFS cluster. Optionally, provide the URI of the Key Management Server (KMS) of the HDFS cluster.
    - **Kerberos**: Provide the Kerberos **Principal** with access to your HDFS cluster. Next, provide the **KeyTab file** that contains the provided Kerberos principal. Then, provide the **Kerberos configuration file**. Finally, specify the type of encryption in transit protection in the **RPC protection** and **Data transfer protection** dropdown lists.
9. (Optional) **Tags** are key-value pairs that help you manage, filter, and search for your location. Adding a tag is optional. We recommend using tags for naming your resources.
10. When you're done, choose **Create location**.

Unsupported HDFS features

The following capabilities of HDFS aren't currently supported by DataSync:

- Transparent Data Encryption (TDE) when using Kerberos authentication
- Configuring multiple NameNodes
- Hadoop HDFS over HTTP (HttpFS)
- POSIX access control lists (ACLs)
- HDFS extended attributes (xattrs)

Creating a location for object storage

A *location* is an endpoint for an object storage system, which can be hosted on-premises or by another cloud provider (for example, a Google Cloud Storage bucket).

Prerequisites

Your object storage system must be compatible with the following Amazon S3 API operations for AWS DataSync to connect to it:

- `AbortMultipartUpload`
- `CompleteMultipartUpload`
- `CopyObject`
- `CreateMultipartUpload`
- `DeleteObject`
- `DeleteObjects`
- `DeleteObjectTagging`
- `GetBucketLocation`
- `GetObject`
- `GetObjectTagging`
- `HeadBucket`
- `HeadObject`
- `ListObjectsV2`

- `PutObject`
- `PutObjectTagging`
- `UploadPart`

Your object storage system must also support AWS Signature Version 4 for authenticating requests. (AWS Signature Version 2 is deprecated.)

Considerations when migrating to or from a Google Cloud Storage bucket

Because DataSync communicates with Google Cloud Storage using the Amazon S3 API, there's a limitation that may cause your DataSync task to fail if you try to copy object tags. To prevent this, deselect the **Copy object tags** option when configuring your task settings. For more information, see File metadata and management options (p. 31).

For detailed instructions on migrating from Google Cloud Storage, see Tutorial: Transferring data from Google Cloud Storage to Amazon S3 (p. 158).

Creating the location

Your object storage system can be a source or destination location for DataSync.

To create an object storage location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**.
3. On the **Locations** page, choose **Create location**.
4. For **Location type**, choose **Object storage**.

   You configure this location as a source or destination later.
5. For **Agents**, choose one or more agents that you want to use.

   During the transfer, the agent securely connects to your object storage server.
6. For **Server**, provide the domain name or IP address of the object storage server.
7. For **Bucket name**, enter the name of the object storage bucket involved in the transfer.
8. For **Folder**, enter an object prefix.

   If this is a source location, DataSync only copies objects with this prefix. If this is a destination location, DataSync writes all objects with this prefix.
9. To select the object storage server protocol and server port, choose **Additional settings** and select either **HTTP** or **HTTPS**.

   By default, the **Server port** is set to **80** for HTTP and **443** for HTTPS. You can specify a custom port if needed.
10. (Optional) If credentials are required to access the object storage location, select **Requires credentials** and enter the **Access key** and **Secret key** for accessing the bucket.

    You can also use the access key and secret key settings to provide a user name and password, respectively.
11. (Optional) **Tags** are key-value pairs that help you manage, filter, and search for your location.

We recommend using tags for naming your locations.

12. Choose **Create location**.

Limitation

DataSync can't copy an object if it has one of the following characters in its name: `$`, `&`, `,`, `:`, `=`, or `@`.

Creating a location for Amazon EFS

An AWS DataSync *location* is an endpoint for an Amazon EFS file system.

Accessing Amazon EFS file systems

DataSync mounts your Amazon EFS file system as the root user from your virtual private cloud (VPC) using network interfaces (p. 18).

When creating your location, you specify the subnet and security groups that DataSync uses to connect to one of your Amazon EFS file system's mount targets or access points using Network File System (NFS) port 2049.

DataSync can also mount Amazon EFS file systems configured for restricted access. For example, you can specify an AWS Identity and Access Management (IAM) role that gives DataSync the necessary level of permission to connect to your file system. For more information, see Using IAM policies to access your Amazon EFS file system (p. 83).

Considerations with Amazon EFS locations

Think about the following when creating a DataSync location for an Amazon EFS file system:

- DataSync doesn't support transferring files to Amazon EFS file system volumes in VPCs with dedicated tenancy. For more information, see Creating a VPC with an instance tenancy of dedicated in the *Amazon EC2 User Guide for Linux Instances*.
- When you create an Amazon EFS file system in Bursting Throughput mode, you get an allocation of 2.1-TB worth of burst credits. All Amazon EFS file systems can burst up to 100 MB per second of throughput with Bursting Throughput mode. File systems with more than 1 TiB of Amazon S3 Standard class storage can drive 100 MiB per second per TB when burst credits are available.

  DataSync consumes file system burst credits. This can have an impact on the performance of your applications. When using DataSync with a file system that has an active workload, consider using EFS Provisioned Throughput.
- Amazon EFS file systems that are in General Purpose performance mode have a limit of 35,000 file system operations per second. This limit can impact the maximum throughput DataSync can achieve when copying files.

  Operations that read data or metadata consume one file operation. Operations that write data or update metadata consume five file operations. This means a file system can support 35,000 read operations per second, 7,000 write operations, or some combination of the two. File operations are counted from all connecting clients.

  For more information, see Amazon EFS performance in the *Amazon Elastic File System User Guide*.

Creating the location

To create the location, you need an existing Amazon EFS file system. If you don't have one, see Getting started with Amazon Elastic File System in the *Amazon Elastic File System User Guide*.

To create an Amazon EFS location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**, and then choose **Create location**.
3. For **Location type**, choose **EFS**.

   You configure this location as a source or destination later.
4. For **File system**, choose the **EFS file system** that you want to use as a location.

   You configure this location as a source or destination later.
5. For **Mount path**, enter a mount path for your Amazon EFS file system.

   This specifies where DataSync reads or writes data (depending on if this is a source or destination location).

   By default, DataSync uses the root directory (or access point if you configure one). You can also specify subdirectories using forward slashes (for example, `/path/to/directory`).
6. For **Subnet** choose a subnet where DataSync creates the network interfaces for managing traffic during your transfer.

   The subnet must be located:

   - In the same VPC as the Amazon EFS file system.
   - In the same Availability Zone as at least one file system mount target.

   > **Note**
   > You don't need to specify a subnet that includes a file system mount target.
7. For **Security groups**, choose the security groups associated with an Amazon EFS file system's mount target.

   > **Note**
   > The security groups you specify must allow inbound traffic on NFS port 2049. For more information, see Using VPC security groups for Amazon EC2 instances and mount targets in the *Amazon Elastic File System User Guide*.
8. For **In-transit encryption**, choose whether you want DataSync to use Transport Layer Security (TLS) encryption when it copies data to or from your file system.

   > **Note**
   > You must enable this setting if you want to configure an access point, IAM role, or both with your location.
9. (Optional) For **EFS access point**, choose an access point that DataSync can use to mount your Amazon EFS file system.
10. (Optional) For **IAM role**, specify a role that allows DataSync to access your file system.

    For information on creating this role, see Using IAM policies to access your Amazon EFS file system (p. 83)
11. (Optional) Select **Add tag** to tag your file system.

    A *tag* is a key-value pair that helps you manage, filter, and search for your locations.

12. Choose **Create location**.

Using IAM policies to access your Amazon EFS file system

You can configure your Amazon EFS file system with a higher level of security by using IAM policies. In your file system policy (p. 83), you can specify an IAM role that still allows DataSync to connect with the file system.

> **Note**
> To use an IAM role, you must enable TLS for in-transit encryption when creating a DataSync location for your file system.

Creating an IAM role for DataSync

Create a role with DataSync as the trusted entity.

To create the IAM role

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, under **Access management**, choose **Roles**, and then choose **Create role**.
3. On the **Select trusted entity** page, for **Trusted entity type**, choose **Custom trust policy**.
4. Paste the following JSON into the policy editor:

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

5. Choose **Next**. On the **Add permissions** page, search for and select the `AmazonElasticFileSystemClientFullAccess` policy.

   > **Important**
   > `AmazonElasticFileSystemClientFullAccess` is an AWS managed policy. It includes the `elasticfilesystem:ClientRootAccess` action, which DataSync needs to mount your Amazon EFS file system as the root user.

6. Choose **Next**. Give your role a name and choose **Create role**.

Specify this role when creating the location for your Amazon EFS file system.

Example Amazon EFS file system policy

The following sample IAM policy includes elements that help restrict access to an Amazon EFS file system (identified in the policy as `fs-1234567890abcdef0`):

- `Principal`: Specifies an IAM role that gives DataSync permission to connect to the file system.
- `Action`: Gives DataSync root access and allows it to read from and write to the file system.

- `aws:SecureTransport`: Requires NFS clients to use TLS when connecting to the file system.
- `elasticfilesystem:AccessPointArn`: Allows access to the file system only through a specific access point.

```
{
    "Version": "2012-10-17",
    "Id": "ExampleEFSFileSystemPolicy",
    "Statement": [{
        "Sid": "AccessEFSFileSystem",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/MyDataSyncRole"
        },
        "Action": [
            "elasticfilesystem:ClientMount",
            "elasticfilesystem:ClientWrite",
            "elasticfilesystem:ClientRootAccess"
        ],
        "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/
fs-1234567890abcdef0",
        "Condition": {
            "Bool": {
                c: "true"
            },
            "StringEquals": {
                "elasticfilesystem:AccessPointArn": "arn:aws:elasticfilesystem:us-
east-1:111122223333:access-point/fsap-abcdef01234567890"
            }
        }
    }]
}
```

Creating a location for FSx for Windows File Server

A location is an endpoint for your Amazon FSx for Windows File Server. AWS DataSync accesses your FSx for Windows File Server using the Server Message Block (SMB) protocol.

DataSync authenticates against your FSx for Windows File Server file system with a user name and password that you configure in the DataSync console or AWS CLI. When you copy data between SMB shares and Amazon FSx or between Amazon FSx locations, the source and destination must belong to the same Active Directory domain or have an Active Directory trust relationship between their domains.

See User to learn more about choosing a user that ensures sufficient permissions to files, folders, and metadata.

The DataSync service mounts your file system from your virtual private cloud (VPC) using elastic network interfaces managed by the DataSync service. DataSync fully manages the creation, the use, and the deletion of these network interfaces on your behalf.

If you don't have an FSx for Windows File Server in the current AWS Region, create one. For information about how to create an FSx for Windows File Server, see Getting started with Amazon FSx in the *Amazon FSx for Windows File Server User Guide*.

> **Note**
> DataSync currently doesn't support transferring files to FSx for Windows File Server volumes that are in dedicated tenancy VPCs. For information about dedicated tenancy VPCs, see Creating a VPC with an instance tenancy of dedicated in the *Amazon EC2 User Guide for Linux Instances*.

To create an Amazon FSx location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**. The locations that you previously created appear in the list of locations.
3. Choose **Create Location** to open the **Create location** page. For **Location type**, choose **Amazon FSx for Windows File Server**.
4. For **FSx for Windows File system**, choose the FSx for Windows File Server that you want to use as an endpoint. You configure this location as a source or destination later.
5. For **Share name**, enter the mount path for your Amazon FSx file server. The path can include a subdirectory. If so, this is a subdirectory in the FSx for Windows File Server that's used to read data from the FSx location or write data to the Amazon FSx destination.

   Note
   The subdirectory must be specified with forward slashes (for example, `//FSx.DNS/share/path/to/folder`).
6. For **Security Group**, the DataSync console automatically chooses the default security group of the subnet for the chosen FSx for Windows File Server. We recommend using these default settings.

   Note
   DataSync uses the security group specified in this step to connect to your FSx for Windows File Server. If the security group is configured to disallow connections from within itself, you have two options:

   - Change the security group configuration to allow the security group to communicate within itself.
   - Choose a different security group, so the selected security group can communicate with the mount target's security group.
7. In the **User settings** section, provide the information for FSx for Windows File Server:

   User

   The user that can mount the location and has the permissions to access the Amazon FSx server.

   To ensure sufficient permissions to files, folders, and file metadata, we recommend that you make this user a member of the file system administrators group. If you are using AWS Directory Service for Microsoft Active Directory with FSx for Windows File Server, the user must be a member of the **AWS Delegated FSx Administrators** group. If you are using a self-managed Microsoft Active Directory with your FSx for Windows File Server, the user must be a member of one of two groups. These are the group of **Domain Admins** or the custom group you specified for file system administration when you created your file system.

   To set object ownership, DataSync requires the `SE_RESTORE_NAME` privilege, which is usually granted to members of the built-in Active Directory groups **Backup Operators** and **Domain Admins**. Providing a user to DataSync with this privilege also helps ensure sufficient permissions to files, folders, and file metadata except for NTFS system access control lists (SACLs).

   Additional privileges are required to copy SACLs. Specifically, this requires the Windows `SE_SECURITY_NAME` privilege, which is granted to members of the **Domain Admins** group. If your task will be configured to copy SACLs, make sure that the user has the required privileges. To learn more about configuring a task to copy SACLs, see Ownership and permissions-related options (p. 31).

   When you copy data between SMB shares and Amazon FSx, or between two Amazon FSx locations, both the source and the destination must belong to the same Active Directory domain, or have an Active Directory trust relationship between their domains.

   Password

The password of the user that can mount the location and has the permissions to access files and folders in the FSx for Windows File Server.

Domain

(Optional) The name of the domain the FSx for Windows File Server belongs to.

8. (Optional) Provide values for the **Key** and **Value** fields to tag the FSx for Windows File Server. A *tag* is a key-value pair that helps you manage, filter, and search for your locations. We recommend using tags to name your resources.
9. When you are done, choose **Create location**. The location that you just created appears in the list of locations.

Creating a location for FSx for Lustre

A location is an endpoint for your Amazon FSx for Lustre file system. AWS DataSync uses the information that you provide in the location to access your file system as a Lustre client.

The DataSync service mounts your FSx for Lustre file system from your virtual private cloud (VPC) using elastic network interfaces managed by the DataSync service. DataSync manages the creation, use, and deletion of these network interfaces on your behalf.

For more information about FSx for Lustre file systems, see What is Amazon FSx for Lustre?

Authorization

DataSync connects to your FSx for Lustre file system using the Lustre client. DataSync requires access to all data on your FSx for Lustre file system. To have this level of access, DataSync mounts your file system as the root user using a user ID (UID) and group ID (GID) of `0`.

To create an FSx for Lustre location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**, and then choose **Create location**.
3. For **Location type**, choose **Amazon FSx**.

   You configure this location as a source or destination later.
4. For **FSx file system**, choose the FSx for Lustre file system that you want to use as a location.
5. For **Mount path**, enter the mount path for you FSx for Lustre file system. The path can include a subdirectory. When the location is used as a source, DataSync reads data from the mount path. When the location is used as a destination, DataSync writes all data to the mount path. If a subdirectory isn't provided, DataSync uses the root directory (`/`).
6. For **Security groups**, select up to five security groups that provide network access to your FSx for Lustre file system. The security groups must provide access on the network ports used by the FSx for Lustre file system. The FSx for Lustre file system must allow network access from the security groups that you select.

   For more information about security groups, see File System Access Control with Amazon VPC in the *Amazon FSx for Lustre User Guide*.
7. (Optional) Provide values for the **Key** and **Value** fields to tag the FSx for Lustre file system. Tags help you manage, filter, and search for your location. We recommend using tags for naming your resources.
8. When you're done, choose **Create location**. The location that you just created appears in the list of locations.

Creating a location for FSx for OpenZFS

A location is an endpoint for your Amazon FSx for OpenZFS file system. AWS DataSync uses the location to access your file system by using the network protocol that you specify.

DataSync mounts your FSx for OpenZFS file system by using elastic network interfaces from the virtual private cloud (VPC) that's associated with your file system. DataSync manages the creation, use, and deletion of these network interfaces for you.

Creating the location

To create the location, you need an existing FSx for OpenZFS file system. If you don't have one, see Getting started with Amazon FSx for OpenZFS in the *Amazon FSx for OpenZFS User Guide*.

To create an FSx for OpenZFS location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**, and then choose **Create location**.
3. For **Location type**, choose **Amazon FSx**.

   You configure this location as a source or destination later.
4. For **FSx file system**, choose the FSx for OpenZFS file system that you want to use as a location.
5. For **Mount path**, enter the mount path for your FSx for OpenZFS file system.

   The path must begin with `/fsx` and can be any existing directory path in the file system. When the location is used as a source, DataSync reads data from the mount path. When the location is used as a destination, DataSync writes all data to the mount path. If a subdirectory isn't provided, DataSync uses the root volume directory (for example, `/fsx`).
6. For **Security groups**, select up to five security groups that provide network access to your FSx for OpenZFS file system.

   The security groups must provide access to the network ports that are used by the FSx for OpenZFS file system. The file system must allow network access from the security groups.

   For more information about security groups, see File system access control with Amazon VPC in the *Amazon FSx for OpenZFS User Guide*.
7. (Optional) Expand **Additional settings** and for **NFS version** choose the NFS version that DataSync uses to access your file system. By default, DataSync uses NFS 4.1.
8. (Optional) Provide values for the **Key** and **Value** fields to tag the FSx for OpenZFS file system. Tags help you manage, filter, and search for your location. We recommend using tags to name your resources.
9. Choose **Create location**.

   Once created, the location displays on the **Locations** page.

Configuring file system authorization

DataSync accesses your FSx for OpenZFS file system as an NFS client, mounting the file system as a root user with a user ID (UID) and group ID (GID) of `0`.

For DataSync to copy all of your file metadata, you must configure the NFS export settings on your file system volumes using `no_root_squash`. However, you can limit this level of access to only a specific DataSync task.

For more information, see Volume properties in the *Amazon FSx for OpenZFS User Guide*.

Configuring NFS exports specific to DataSync (recommended)

You can configure an NFS export specific to each volume that's only accessed by your DataSync task. Do this for the most recent ancestor volume of the mount path you specify when creating your FSx for OpenZFS location.

To configure an NFS export specific to DataSync

1. Create your DataSync task (p. 98).

   This creates the task's elastic network interfaces that you'll specify in your NFS export settings.
2. Locate the private IP addresses of the task's network interfaces by using the Amazon EC2 console or AWS CLI.
3. For your FSx for OpenZFS file system volume, configure the following NFS export settings for each of the task's network interfaces:

   - **Client address**: Enter the network interface's private IP address (for example, `10.24.34.0`).
   - **NFS options**: Enter `rw,no_root_squash`.

Configuring NFS exports for all clients

You can specify an NFS export that allows root access to all clients.

To configure an NFS export for all clients

- For your FSx for OpenZFS file system volume, configure the following NFS export settings:

  - **Client address**: Enter `*`.
  - **NFS options**: Enter `rw,no_root_squash`.

Creating a location for Amazon S3

An Amazon S3 bucket can be a source or destination location for AWS DataSync.

Remember the following when using Amazon S3 with DataSync:

- Changes to object data or metadata are equivalent to deleting and replacing an object. These changes result in additional charges in the following scenarios:
  - **When using object versioning**: Changes to object data or metadata create a new version of the object.
  - **When using storage classes that can incur additional charges for overwriting, deleting, or retrieving objects**: Changes to object data or metadata result in such charges. For smore information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89).
- When using object versioning, running a DataSync task once might create more than one version of an Amazon S3 object.
- DataSync requires access to your Amazon S3 bucket. To do this, DataSync assumes an IAM role that includes an IAM policy and security token service trust (STS) relationship. The policy determines which actions the role can perform. Let DataSync create the role for you or specify a role you created. For more information, see Manually configuring an IAM role to access your Amazon S3 bucket (p. 91).

- In addition to the IAM policies that grant DataSync permissions, we recommend creating a multipart upload bucket policy for your S3 buckets to help control your storage costs. For more information, see the blog post Amazon S3 Lifecycle Management Update - Support for Multipart Uploads and Delete Markers.

To create an Amazon S3 location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. Go to the **Locations** page and select **Create location**.
3. For **Location type**, choose **Amazon S3**.
4. For **S3 bucket**, choose the bucket that you want to use as a location. (When creating your DataSync task later, you specify whether this location is a source or destination location.)

   If your S3 bucket is located on AWS Outposts, you must specify an Amazon S3 access point. For more information, see Managing data access with Amazon S3 access points in the *Amazon S3 User Guide*.
5. For **S3 storage class**, choose a storage class that you want to transfer objects into.

   Some storage classes can affect your Amazon S3 costs. For more information, see Considerations when working with Amazon S3 storage classes in DataSync (p. 89). For Amazon S3 on AWS Outposts, DataSync by default uses the S3 Outposts storage class.
6. For **Agents**, specify the Amazon Resource Name (ARN) of the DataSync agent on your AWS Outposts.

   For more information, see Deploy your agent on AWS Outposts (p. 25).
7. For **Folder**, enter a prefix in the S3 bucket that DataSync reads from or writes to (depending on whether the bucket is a source or destination location).

   > **Note**
   > The prefix can't begin with a slash (for example, `/photos`) or include consecutive slashes, such as `photos//2006/January`.
8. For **IAM role**, do one of the following:

   - Choose **Autogenerate** for DataSync to automatically create an IAM role with the permissions required to access the S3 bucket.

     If DataSync previously created an IAM role for this S3 bucket, that role is chosen by default.
   - Select a custom IAM role you created. For more information, see Manually configuring an IAM role to access your Amazon S3 bucket (p. 91)
9. (Optional) Select **Add tag** to tag your Amazon S3 location.

   A *tag* is a key-value pair that helps you manage, filter, and search for your locations.
10. Choose **Create location**.

    Once created, the location displays on the **Locations** page.

Considerations when working with Amazon S3 storage classes in DataSync

DataSync can transfer objects directly into the Amazon S3 storage class that you choose. For more information about Amazon S3 storage classes, see Amazon S3 storage classes. Some storage classes have behaviors that can affect your Amazon S3 storage costs. For more information, see Amazon S3 pricing.

Following, you can find some considerations for how each Amazon S3 storage class works with DataSync.

Amazon S3 storage class	Considerations
S3 Standard	Choose S3 Standard to store your frequently accessed files redundantly in multiple Availability Zones that are geographically separated. This is the default if you don't specify a storage class.
S3 Intelligent-Tiering	Choose S3 Intelligent-Tiering to optimize storage costs by automatically moving data to the most cost-effective storage access tier. You pay a monthly charge per object stored in the S3 Intelligent-Tiering storage class. This Amazon S3 charge includes monitoring data access patterns and moving objects between tiers.
S3 Standard-IA	Choose S3 Standard-IA to store your infrequently accessed files redundantly in multiple Availability Zones that are geographically separated. Objects stored in the S3 Standard-IA storage class can incur additional charges for overwriting, deleting, or retrieving. Consider how often these objects change, how long you plan to keep these objects, and how often you need to access them. Changes to object data or metadata are equivalent to deleting an object and creating a new one to replace it. This results in additional charges for objects stored in the S3 Standard-IA storage class. Objects less than 128 KB are smaller than the minimum capacity charge per object in the S3 Standard-IA storage class. These objects are stored in the S3 Standard storage class.
S3 One Zone-IA	Choose S3 One Zone-IA to store your infrequently accessed files in a single Availability Zone. Objects stored in the S3 One Zone-IA storage class can incur additional charges for overwriting, deleting, or retrieving. Consider how often these objects change, how long you plan to keep these objects, and how often you need to access them. Changes to object data or metadata are equivalent to deleting an object and creating a new one to replace it. This results in additional charges for objects stored in the S3 One Zone-IA storage class. Objects less than 128 KB are smaller than the minimum capacity charge per object in the S3 One Zone-IA storage class. These objects are stored in the S3 Standard storage class.
S3 Glacier Flexible Retrieval	Choose S3 Glacier Flexible Retrieval for more active archives. Objects stored in S3 Glacier Flexible Retrieval can incur additional charges for overwriting, deleting, or retrieving. Consider how often these objects change, how long you plan to keep these objects, and how often you need to access them. Changes to object data or metadata are equivalent to deleting an object and creating a new one to replace it. This results in additional charges for objects stored in the S3 Glacier Flexible Retrieval storage class. Objects less than 40 KB are smaller than the minimum capacity charge per object in the S3 Glacier Flexible Retrieval storage class. These objects are stored in the S3 Standard storage class.

Amazon S3 storage class	Considerations
	You must restore objects archived in this storage class before DataSync can read them. For information, see Working with archived objects in the *Amazon S3 User Guide*. When using S3 Glacier Flexible Retrieval, choose **Verify only the data transferred** to compare data and metadata checksums at the end of the transfer. You can't use the **Verify all data in the destination** option for this storage class because it requires retrieving all existing objects from the destination.
S3 Glacier Deep Archive	Choose S3 Glacier Deep Archive to archive your files for long-term data retention and digital preservation where data is accessed once or twice a year. Objects stored in S3 Glacier Deep Archive can incur additional charges for overwriting, deleting, or retrieving. Consider how often these objects change, how long you plan to keep these objects, and how often you need to access them. Changes to object data or metadata are equivalent to deleting an object and creating a new one to replace it. This results in additional charges for objects stored in the S3 Glacier Deep Archive storage class. Objects less than 40 KB are smaller than the minimum capacity charge per object in the S3 Glacier Deep Archive storage class. These objects are stored in the S3 Standard storage class. You must restore objects archived in this storage class before DataSync can read them. For information, see Working with archived objects in the *Amazon S3 User Guide*. When using S3 Glacier Deep Archive, choose **Verify only the data transferred** to compare data and metadata checksums at the end of the transfer. **Verify all data in the destination** isn't an available option for this storage class, because it requires retrieving all existing objects from the destination.
S3 Outposts	The storage class for Amazon S3 on Outposts.

Manually configuring an IAM role to access your Amazon S3 bucket

When you use the DataSync console to create an Amazon S3 location, DataSync automatically creates an IAM role that has the required permissions for you. If you want to create the IAM role manually, use the following procedure.

To manually configure an IAM role to access your Amazon S3 bucket

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, choose **Roles**, and then choose **Create role** to open the **Create role** page.
3. In the **Select type of trusted entity** section, make sure that **AWS service** is selected.
4. Under **Choose the service that will use this role**, choose **DataSync** or manually configure it (see the following example).

To prevent the cross-service confused deputy problem (p. 128), we recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in the policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "datasync.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:datasync:us-east-2:123456789012:*"
                }
            }
        }
    ]
}
```

5. Under **Select your use case**, choose **DataSync - S3 Location**.
6. Choose **Next: Permissions**.
7. For Amazon S3 buckets in AWS Regions, choose **AmazonS3FullAccess**. You can also manually configure a more restrictive policy. For an example of such a policy, see the following.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn"
        },
        {
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListMultipartUploadParts",
                "s3:GetObjectTagging",
                "s3:PutObjectTagging",
                "s3:PutObject"
              ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn/*"
        }
    ]
}
```

For Amazon S3 buckets on Outposts, use the following policy.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3-outposts:ListBucket",
                "s3-outposts:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": [
                "s3OutpostsBucketArn",
                "s3OutpostsAccessPointArn"
            ],
            "Condition": {
                "StringLike": {
                    "s3-outposts:DataAccessPointArn": "s3OutpostsAccessPointArn"
                }
            }
        },
        {
            "Action": [
                "s3-outposts:AbortMultipartUpload",
                "s3-outposts:DeleteObject",
                "s3-outposts:GetObject",
                "s3-outposts:ListMultipartUploadParts",
                "s3-outposts:GetObjectTagging",
                "s3-outposts:PutObjectTagging"
            ],
            "Effect": "Allow",
            "Resource": [
                "s3OutpostsBucketArn/*",
                "s3OutpostsAccessPointArn"
            ],
            "Condition": {
                "StringLike": {
                    "s3-outposts:DataAccessPointArn": "s3OutpostsAccessPointArn"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3-outposts:GetAccessPoint"
            ],
            "Resource": "s3OutpostsAccessPointArn"
        }
    ]
}
```

8. (Optional) **Choose Next: Tags** to create tags for the role.
9. Choose **Next: Review**, choose the role name, and then choose **Create role**.
10. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
11. Choose the refresh button on the right side of the IAM role list, and then choose the role that you just created.

How DataSync handles metadata and special files

AWS DataSync saves metadata and special files (links and directories) when copying data to and from storage systems and services.

However, DataSync does not copy system-level settings. For example, when copying objects, DataSync doesn't copy your storage system's encryption setting. If you're copying from an SMB share, DataSync doesn't copy the permissions you configured at the file system level.

Topics

Metadata copied by DataSync

DataSync preserves metadata between storage systems that have similar metadata structures.

When copying between self-managed Network File System (NFS), Amazon FSx for Lustre, Amazon FSx for OpenZFS, or Amazon EFS, and Amazon EFS, Amazon FSx for Lustre, or Amazon FSx for OpenZFS – In this case, DataSync can copy the following metadata:

- File and folder modification timestamps
- User ID (UID) and group ID (GID)
- POSIX permissions

When copying between Hadoop Distributed File System (HDFS) and Amazon EFS, FSx for Lustre, or FSx for OpenZFS – In this case, DataSync can copy the following metadata:

- File and folder modification timestamps
- POSIX permissions

> **Note**
> HDFS uses strings to store file and folder user and group ownership, rather than numeric identifiers (such as UIDs and GIDs). When copying from HDFS to Amazon EFS, FSx for Lustre, or FSx for OpenZFS, default values for UIDs and GIDs are applied on the destination file system. For more information about default values, see Default POSIX metadata applied by DataSync (p. 96).

When copying between self-managed Server Message Block (SMB) or Amazon FSx for Windows File Server, and FSx for Windows File Server – In this case, DataSync can copy the following metadata:

- File timestamps: access time, modification time, and creation time
- File owner security identifier (SID)
- Standard file attributes:
  - Read-only (R)
  - Archive (A)
  - System (S)
  - Hidden (H)
  - Compressed (C)
  - Not content indexed (N)
  - Encrypted (E)
  - Temporary (T)
  - Offline (O)
  - Sparse (P)

> **Note**
> DataSync attempts to copy the archive, compressed, and sparse attributes. If these attributes aren't applied on the destination, they're ignored during task verification.

- NTFS discretionary access lists (DACLs), which determine whether to grant access to an object
- NTFS system access control lists (SACLs), which are used by administrators to log attempts to access a secured object

When copying between self-managed NFS, FSx for Lustre, FSx for OpenZFS, or Amazon EFS and Amazon S3 – In this case, the following metadata is stored as Amazon S3 user metadata:

- File and folder modification timestamps
- User ID and group ID
- POSIX permissions

The file metadata that is stored in Amazon S3 user metadata is interoperable with NFS shares on file gateways in AWS Storage Gateway. A file gateway enables low-latency access from on-premises networks to data that was copied to Amazon S3 by DataSync. This metadata is also interoperable with Amazon FSx for Lustre.

When DataSync copies objects that contain this metadata back to an NFS server, the file metadata is restored. Restoring metadata requires granting elevated permissions to the NFS server. For more information, see Creating a location for NFS (p. 74).

When copying between HDFS and Amazon S3 – In this case, the following metadata is stored as Amazon S3 user metadata:

- File and folder modification timestamps
- User name and group name
- POSIX permissions

> **Note**
> HDFS uses strings to store file and folder user and group ownership, rather than numeric identifiers, such as UIDs and GIDs. DataSync ignores user and group name metadata values stored in Amazon S3 when copying to EFS or self-managed NFS.

When copying between object storage systems and Amazon S3 or between two Amazon S3 buckets – In this case, DataSync only copies user-defined metadata and tags. DataSync doesn't copy other object information, such as object access control lists (ACLs) or prior object versions.

> **Important**
> If you're transferring objects from a Google Cloud Storage bucket, copying object tags may cause your DataSync task to fail. To prevent this, deselect the **Copy object tags** option when configuring your task settings. For more information, see File metadata and management options (p. 31).

When copying between storage systems that don't have similar metadata structure – In this case, DataSync sets metadata using the following rules.

If you copy this way	This happens to metadata
From an SMB share to Amazon EFS, FSx for Lustre, FSx for OpenZFS, or Amazon S3	Default POSIX metadata is set for all files and folders on the target NFS server, FSx for Lustre file

If you copy this way	This happens to metadata
From FSx for Windows File Server to an NFS share or HDFS	system, FSx for OpenZFS file system, or Amazon EFS file system, or stored in the Amazon S3 object's metadata. This approach includes using the default POSIX user ID and group ID values. On HDFS, file and folder timestamps are applied from the source. The file or folder owner is set based on the user or Kerberos principal specified in DataSync. The Groups Mapping configuration on the Hadoop cluster determines the group.
From an NFS share or HDFS to FSx for Windows File Server From Amazon EFS, FSx for Lustre, FSx for OpenZFS, or Amazon S3 to an SMB share	File and folder timestamps are applied from the source. Ownership is set based on the Windows user that was specified in DataSync to access the Amazon FSx or SMB share. Permissions are inherited from the parent directory.

Default POSIX metadata applied by DataSync

When the source and destination don't have a similar metadata structure, or when source metadata is missing, DataSync applies default POSIX metadata.

Specifically, DataSync applies this metadata in these situations:

- When transferring files from an Amazon S3 or self-managed object storage location to an Amazon EFS, FSx for Lustre, FSx for OpenZFS, NFS, or HDFS location, in cases where Amazon S3 objects don't have DataSync POSIX metadata
- When transferring from an SMB location to an NFS, HDFS, Amazon S3, FSx for Lustre, FSx for OpenZFS, or Amazon EFS location

The following table shows the default POSIX metadata and permissions that DataSync applies.

Permission	Value
UID	65534
GID	65534
Folder Permission	0755
File Permission	0644

HDFS stores file and folder user and group ownership using strings, rather than numeric identifiers, such as UIDs and GIDs. When there is no equivalent metadata on the source location, file and folder ownership is set based on the user or Kerberos principal that you specified in DataSync. The group is determined by the Groups Mapping configuration on the Hadoop cluster.

Links and directories copied by DataSync

Understand how DataSync handles copied hard links, symbolic links, and directories in different storage locations.

Hard links

When copying between an NFS server, FSx for Lustre, FSx for OpenZFS, and Amazon EFS, hard links are preserved.

When copying to Amazon S3, each hard link is transferred only once. Separate Amazon S3 objects are created for each copy. If a hard link is unchanged in Amazon S3, it's correctly restored upon transfer to an NFS server, FSx for Lustre, FSx for OpenZFS, or Amazon EFS.

When copying between SMB file shares and FSx for Windows File Server, hard links aren't supported. If DataSync encounters hard links in these situations, the task completes with an error. Check your Amazon CloudWatch Logs to learn more.

When copying to HDFS, hard links aren't supported. When copying to HDFS, hard links on the source are skipped and logged to CloudWatch.

Symbolic links

When copying between an NFS server, FSx for Lustre, FSx for OpenZFS, and Amazon EFS, symbolic links are preserved.

When copying to Amazon S3, the link target path is stored in the Amazon S3 object. The link is correctly restored upon transfer to an NFS server, FSx for Lustre, FSx for OpenZFS, or Amazon EFS.

When copying between SMB file shares and FSx for Windows File Server, symbolic links aren't supported. If DataSync encounters symbolic links in these situations, the task completes with an error. Check your CloudWatch Logs to learn more.

When copying to HDFS, symbolic links aren't supported. When copying to HDFS, symbolic links are skipped and logged to CloudWatch.

Directories

When copying to or from Amazon S3 buckets, directories are represented as empty objects ending with `/`.

For information on logging with DataSync, see Monitoring AWS DataSync (p. 113).

Deleting a location

You can delete any type of location.

To delete a location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Locations**.
3. On the **Locations** page, choose the location that you want to delete.
4. Choose **Delete**. Confirm the deletion by entering **`delete`** and select **Delete**.

Working with AWS DataSync tasks

In AWS DataSync, a *task* defines where and how you're transferring data. You specify a source and destination location for your task. You also can customize the task to fit your scenario, such as scheduling when the task runs and filtering certain types of data.

Topics

Creating your DataSync task

If this is your first time using DataSync, the instructions in Getting started with AWS DataSync (p. 20) walk you through the process of creating a task.

Topics

Prerequisite: Creating the locations for your DataSync task

A DataSync task requires a source and destination location to transfer data between.

- Creating a location for NFS (p. 74)
- Creating a location for SMB (p. 75)
- Creating a location for HDFS (p. 77)
- Creating a location for object storage (p. 79)
- Creating a location for Amazon EFS (p. 81)
- Creating a location for FSx for Windows File Server (p. 84)
- Creating a location for FSx for Lustre (p. 86)
- Creating a location for FSx for OpenZFS (p. 87)
- Creating a location for Amazon S3 (p. 88)

For a list of supported DataSync locations and transfer scenarios, see Working with AWS DataSync locations (p. 72).

Creating a task to transfer data between self-managed storage and AWS

If you have previously created a task and want to create additional tasks, use the following procedure.

To create a task

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Tasks**, and then choose **Create task**.
3. On the **Configure source location** page, choose **Create new location** and configure a new location if you want to use a new location for your source. Provide the configuration settings and choose **Next**. For instructions on how to create a location, see Working with AWS DataSync locations (p. 72).

   If you want to use a source location that you previously created, choose **Choose existing location**, choose your source location from the list, and then choose **Next**.

   For step-by-step instruction, see Configure a source location (p. 28).

Creating a task to transfer between in-cloud locations

Use the following instructions to set up the DataSync agent on an Amazon EC2 instance for data transfers. The examples in this section cover these use cases:

- Transferring data from a cloud file system to another cloud file system or Amazon S3 (p. 59) – Transfer data from Amazon EFS to Amazon EFS, from self-managed NFS to EFS, or to Amazon S3.
- Data transfer from S3 to in-cloud file systems (p. 60) – Transfer data from Amazon S3 to Amazon EFS, or from Amazon S3 to self-managed NFS.

Creating a task to transfer from in-cloud NFS to in-cloud NFS or Amazon S3

Use the following instructions to transfer data from an in-cloud NFS file system to AWS. To perform this transfer, the DataSync agent must be located in the same AWS Region and same AWS account where the file system is deployed. This type of transfer includes transfers from EFS to EFS, transfers from self-managed NFS to Amazon EFS, and transfers to Amazon S3. For information about how in-cloud NFS to in-cloud NFS or Amazon S3 works, see Transferring data from a cloud file system to another cloud file system or Amazon S3 (p. 59).

> **Note**
> Deploy the agent in the AWS Region and AWS account where the source EFS or self-managed NFS file system resides.

Deploying your DataSync agent as an Amazon EC2 instance to read files from in-cloud

To deploy the DataSync agent as an Amazon EC2 instance

1. From the AWS account where the source EFS resides, launch the agent by using your Amazon Machine Image (AMI) from the Amazon EC2 launch wizard. Use the following URL to launch the AMI.

```
https://console.aws.amazon.com/ec2/v2/home?region=source-efs-or-nfs-
region#LaunchInstanceWizard:ami=ami-id
```

In the URL, replace the *source-efs-or-nfs-region* and *ami-id* with your own.

After the AMI launches, the **Choose an Instance Type** appears on the Amazon EC2 console. For a list of AMI IDs by AWS Region, see Deploy your agent as an Amazon EC2 instance (p. 23).

2. Choose one of the recommended instance types for your use case, and choose **Next: Configure Instance Details**. For the recommended instance types, see Amazon EC2 instance requirements (p. 10).
3. On the **Configure Instance Details** page, do the following:

   a. For **Network**, choose the VPC where your source EFS or NFS is located.

   b. Choose a value for **Auto-assign Public IP**. If you want your instance to be accessible from the public internet, set **Auto-assign Public IP** to **Enable**. Otherwise, set **Auto-assign Public IP** to **Disable**. If a public IP address isn't assigned, activate the agent in your VPC using its private IP address.

   When you transfer files from an in-cloud NFS, to increase performance, we recommend that you choose the **Placement Group** where your NFS server resides.
4. Choose **Next: Add Storage**. The agent doesn't require additional storage, so you can skip this step and choose **Next: Add tags**.
5. (Optional) On the **Add Tags** page, you can add tags to your Amazon EC2 instance. When you're finished on the page, choose **Next: Configure Security Group**.
6. On the **Configure Security Group** page, do the following:

   a. Make sure that the selected security group allows inbound access to HTTP port 80 from the web browser that you plan to use to activate the agent.

   b. Make sure that the security group of source EFS or NFS allows inbound traffic from the agent. In addition, make sure that the agent allows outbound traffic to the source EFS or NFS. The traffic goes through the standard NFS port, 2049.

   For the complete set of network requirements for DataSync, see Network requirements (p. 10).
7. Choose **Review and Launch** to review your configuration, then choose **Launch** to launch your instance. Remember to use a key pair that's accessible to you. A confirmation page appears and indicates that your instance is launching.
8. Choose **View Instances** to close the confirmation page and return to the Amazon EC2 instances screen. When you launch an instance, its initial state is **pending**. After the instance starts, its state changes to **running**. At this point, it's assigned a public Domain Name System (DNS) name and IP address, which can be found in the **Descriptions** tab.
9. If you set **Auto-assign Public IP** to **Enable**, choose your instance and note the public IP address in the **Description** tab. You use this IP address later to connect to your sync agent.

   If you set **Auto-assign Public IP** to **Disable**, launch or use an existing instance in your VPC to activate the agent. In this case, you use the private IP address of the sync agent to activate the agent from this instance in the VPC.

Creating a task to transfer data from EFS or self-managed storage

Next, you create a task to transfer data.

Note
Create the task in the AWS Region and AWS account where the destination EFS or Amazon S3 bucket resides.

To create a task

1. Open the DataSync console in the AWS Region where your destination Amazon EFS file system is located. The destination EFS or Amazon S3 bucket must be in the same AWS account.
2. Choose **Create task**, then choose **On-premises to AWS** on the **Use case options** page, and then choose **Create agent**.
3. In the **Create agent** wizard's **Activation** section, enter the Amazon EC2 instance's IP address for **Agent address**, and then choose **Get key**. This IP address can be private or public. For more details, see step 9 of To deploy the DataSync agent as an Amazon EC2 instance (p. 99).

   Your browser connects to this IP address to get a unique activation key from your agent. This key securely associates your agent with your AWS account. This IP address doesn't need to be accessible from outside your network, but must be accessible from your browser.
4. Enter an agent name that you can easily identify later, and choose **Create agent** when done. You can optionally add tags to the agent.
5. Choose **Tasks** from the navigation pane.
6. Choose **On-premises to AWS**, and choose **Next** to open the **Source configuration** page.
7. In the **Source location options**, choose **Create new location** and choose **Network File System (NFS) or Server Message Block (SMB)**. Fill in the following options:

   - For agent, choose your newly created agent from the list.
   - If you are copying from EFS, do the following:
     - For **NFS Server**, enter the **DNS name** of your source EFS.
     - For **Mount path**, enter **/** (forward slash) and choose **Next**.
   - If you are copying from self-managed NFS or SMB, do the following:
     - For **NFS Server**, enter the private DNS or IP address of your source NFS.
     - For **Mount path**, enter a path that's exported by your NFS server and choose **Next**. For more information, see Creating an NFS location (p. 39).
8. Choose **Create new location**. This is the destination location for your data transfer. Fill in the following options:

   - If you are copying to EFS, do the following:
     - For **Location type**, choose **EFS**.
     - Choose your destination EFS.
     - For **Mount path**, enter **/** (forward slash).
     - For **Subnet** and **Security groups**, use the default settings and choose **Next**.
   - If you are copying to Amazon S3, do the following:
     - For **Location type**, choose Amazon S3 bucket.
     - For **Amazon S3 bucket**, choose your source Amazon S3 bucket.
     - For **Folder**, choose a folder prefix to use for the transfer, or you can keep it blank.
     - Choose your destination Amazon S3 bucket and an optional folder. DataSync can generate an AWS Identity and Access Management (IAM) role to access your bucket, or you can create on your own.
9. Choose **Next**, and optionally name the task and add tags.
10. Choose or create an Amazon CloudWatch Logs log group at the bottom of the page, and choose **Next**. For more information on working with CloudWatch Logs, see Allowing DataSync to upload logs to Amazon CloudWatch log groups (p. 115).

11. Review the settings on the next page, and choose **Create task**.
12. Choose **Start** to run the task that you just created to start transferring data.

Creating a task to transfer from Amazon S3 to in-cloud NFS

Use the following instructions to transfer data from Amazon S3 to an in-cloud NFS file system that's located in the same AWS account and AWS Region where the agent is deployed. This approach includes transfers from Amazon S3 to EFS, or from Amazon S3 to self-managed NFS. The following diagram illustrates this type of transfer. For information about how Amazon S3 to in-cloud NFS works, see Data transfer from S3 to in-cloud file systems (p. 60).

Deploying the DataSync agent on an Amazon EC2 instance to write to your destination location

First, deploy the DataSync agent on an Amazon EC2 instance in the AWS Region and AWS account where the destination EFS file system or self-managed NFS server resides.

To deploy the agent

- Launch the agent from the selected AMI by using the Amazon EC2 launch wizard. To do so, use the following URL.

```
https://console.aws.amazon.com/ec2/v2/home?region=DESTINATION-EFS-or-NFS-
REGION#LaunchInstanceWizard:ami=AMI-ID.
```

  In the URL, replace the AWS Region and AMI ID with your own. You are redirected to the **Choose an Instance Type** page on the Amazon EC2 console. For a list of AMI IDs by AWS Region, see Deploy your agent as an Amazon EC2 instance (p. 23).

Creating a task to transfer data from Amazon S3

Next, you create a task to transfer data.

> **Note**
> Create the task in the AWS account and AWS Region where the source Amazon S3 bucket resides.

To create a task that transfers data from Amazon S3 to EFS or a self-managed NFS or SMB

1. Open the DataSync console in the AWS Region where your source Amazon S3 bucket is located.
2. Choose **Create task**, and choose the use case **AWS to on-premises**.
3. Choose **Create agent**.
4. If you set **Auto-assign Public IP** to **Enable**, choose your instance and note the public IP address in the **Description** tab. You use this IP address later to connect to your sync agent.

   If you set **Auto-assign Public IP** to **Disable**, launch or use an existing instance in your VPC to activate the agent. In this case, you use the private IP address of the sync agent to activate the agent from this instance in the VPC.
5. In the **Create agent** wizard, for **Agent address** enter the Amazon EC2 instance's IP address (private or public, as explained in step 3), and then choose **Get key**.

   Your browser connects to this IP address to get a unique activation key from your agent. This key securely associates your agent with your AWS account. This IP address doesn't need to be accessible from outside your network, but must be accessible from your browser.

6. Choose an agent name that you can easily identify later. You can optionally add tags. When you're done, choose **Create agent**.
7. Choose **AWS to on-premises**, and choose **Next**.
8. Choose **Create new location**:

   - For **Location type**, choose Amazon S3 bucket.
   - For **Amazon S3 bucket**, choose your source Amazon S3 bucket.
   - For **Folder**, choose a folder prefix for the transfer, or you can keep it blank.

   DataSync can generate an IAM role to access your bucket, or you can create on your own.
9. Choose **Next**. Choose **Create new location**, choose **NFS or SMB** for **Location type**, and choose the agent that you just created from the list.
10. a. If you are copying to EFS, do the following:

       - For **NFS Server**, enter the **DNS name** of your source EFS.
       - For **Mount path**, enter **/** (forward slash) and choose **Next**.

    b. If you are copying to in-cloud NFS, do the following:

       - For **NFS Server**, enter the private DNS or IP address of your source NFS.
       - For **Mount path**, enter a path that is exported by your NFS server. For more information, see Creating an NFS location (p. 39).
11. Choose **Next**, and optionally name the task and add tags.
12. Choose or create a CloudWatch Logs log group at the bottom of the page, and choose **Next**. For more information on working with CloudWatch Logs, see Allowing DataSync to upload logs to Amazon CloudWatch log groups (p. 115).
13. Review the settings on the next page, and choose **Create task**.
14. Choose **Start** to run the task that you just created to transfer data, and then choose **Start** again on the **Start Task** page.

Configuring task settings

Following, you can find information on how to configure a task setting. You use these settings to control how a task execution behaves. These settings are available in the **Options** section.

These options control the behavior of a task execution. Behavior includes preserving metadata such as the user ID (UID) or group ID (GID), preserving file permissions, and data integrity verification. If you don't specify values for these options, DataSync uses a set of default options that can be overridden for a task execution.

For more information about configuring a DataSync task, see Configure task settings (p. 30).

The available options are as follows:

- **Data verification** – Task data verification options specify how to verify data that's transferred by the task. For more information about configuring these options, see Data verification options (p. 30).
- **Ownership and permissions** – DataSync preserves metadata between storage systems that have similar metadata structures. Depending on the storage system type, different options are used to configure such metadata preservation. For more information about configuring these options, see Ownership and permissions-related options (p. 31).
- **File metadata and management** – You can configure DataSync tasks to copy file metadata, keep deleted files, and overwrite files in the destination. For more information, see File metadata and management options (p. 31).

- **Bandwidth** – You can configure a bandwidth limit for DataSync tasks. For more information about configuring bandwidth options, see Bandwidth options (p. 32).
- **Filtering** – When you transfer data from your source to your destination location, you can apply filters to transfer only a subset of the files in your source location. For more information about configuring filtering options, see Filtering options (p. 32).
- **Scheduling and queueing** – You can schedule a DataSync task to be run at a specific time. If you are using a single agent to run multiple tasks, you can queue those tasks. For more information about configuring scheduling and queueing options, see Scheduling and queueing options (p. 32).
- **Tags and logging** – You can add one or more tags to organize a DataSync task. For logging, you can have DataSync publish logs for individual files or objects to the CloudWatch log group that you specify. For more information about configuring tags and logging options, see Tags and logging options (p. 33).

Filtering data transferred by AWS DataSync

AWS DataSync lets you apply filters if you only want to transfer a subset of data (such as specific files, folders, or objects).

For example, if your source location includes temporary files that end with `.tmp`, you can create an exclude filter that keeps these files from making their way to the destination location. You also can use a combination of exclude and include filters in the same task.

Topics

Filtering terms, definitions, and syntax

These are some terms and definitions for use with filtering:

Filter

The whole string that makes up a particular filter, for example: `*.tmp|*.temp` or `/folderA|/folderB`

Filters are made up of patterns delimited with a | (pipe). A delimiter isn't needed when you add patterns on the console because you add each pattern separately.

Pattern

A pattern within a filter. For example, `*.tmp` is a pattern that's part of the `*.tmp|*.temp` filter.

Folders

- All filters are relative to the source location path. For example, suppose that you specify `/my_source/` as the source path when you create your source location and task and specify the include filter `/transfer_this/`. In this case, DataSync transfers only the directory `/my_source/transfer_this/` and its contents.
- To specify a folder directly under the source location, include a forward slash (/) in front of the folder name. In the example preceding, the pattern uses `/transfer_this`, not `transfer_this`.
- DataSync interprets the following patterns the same way and matches both the folder and its content.

```
/dir
```

`/dir/`

- When you are transferring data from or to an Amazon S3 bucket, DataSync treats the / character in the object key as the equivalent of a folder on a file system.

Special characters

Following are special characters for use with filtering.

Special character	Description
`*` (wildcard)	A character used to match zero or more characters. For example, `/movies_folder*` matches both `/movies_folder` and `/movies_folder1`.
`|` (pipe delimiter)	A character used as a delimiter between patterns. It enables specifying multiple patterns, any of which can match the filter. For example, `*.tmp|*.temp` matches files ending with either `tmp` or `temp`. **Note** This delimiter isn't needed when you add patterns on the console because you add each pattern on a separate line.
`\` (backslash)	A character used for escaping special characters (*, |, \) in a file or object name. A double backslash (\\) is required when a backslash is part of a file name. Similarly, \\\\ represents two consecutive backslashes in a file name. A backslash followed by a pipe (\\|) is required when a pipe is part of a file name. A backslash (\) followed by any other character, or at the end of a pattern, is ignored.

Excluding data from a transfer

Exclude filters define files, folders, and objects that are excluded when you transfer files from a source to a destination location. You can configure these filters when you create, edit, or start a task.

To create a task with an exclude filter in the DataSync console, specify a list of patterns in the **Data transfer configuration** section under **Exclude patterns**. For example, to exclude the temporary folders named `temp` or `tmp`, you can specify `*/temp` in the **Exclude patterns** text box, choose **Add patterns** and then specify `*/tmp` in the second text box. To add more patterns to the filter, choose **Add pattern**. When you're using the AWS Command Line Interface (AWS CLI), single quotation marks (`'`) are required around the filter and a | (pipe) is used as a delimiter. For this example, you would specify `'*/temp|*/tmp'`.

After you have created a task, you can edit the task configuration to add or remove patterns from the exclude filter. Your changes are applied to future executions of the task.

When you run a task, you can modify the exclude filter patterns by using the **Start with overrides** option. Any changes that you make are applied only to that execution of the task.

You can also use the AWS CLI to create or edit an exclude filter. The following example shows such a CLI command.

```
aws datasync create-task
    --source-location-arn 'arn:aws:datasync:region:account-id:location/location-id'
    --destination-location-arn 'arn:aws:datasync:region:account-id:location/location-id'
    --cloud-watch-log-group-arn 'arn:aws:logs:region:account-id:log-group:your-log-group'
    --name your-task-name
    --excludes FilterType=SIMPLE_PATTERN,Value='*/temp|*/tmp'
```

> **Note**
> If you are migrating files from a NetApp system, we recommend that you exclude NetApp backup folders by specifying `*/.snapshot` as a pattern in your exclude filter.

Including data in a transfer

Include filters define files, folders, and objects that DataSync transfers when you run a task. You can configure include filters when you create, edit, or start a task.

To create a task with an include filter, choose the **Specific files and folders** option, and then specify a list of patterns to include under **Include patterns**.

DataSync scans and transfers only files and folders that match the include filters. For example, to include a subset of your source folders, you might specify `/important_folder_1|/important_folder_2`.

After you have created a task, you can edit the task configuration to add or remove patterns from the include filter. Any changes that you make are applied to future executions of the task.

When you run a task, you can modify the include filter patterns by using the **Start with overrides** option. Any changes that you make are applied only to that execution of the task.

You can also use the AWS CLI to create or edit an include filter. The following example shows the CLI command. Take note of the quotation marks (`'`) around the filter and the `|` (pipe) that's used as a delimiter.

```
aws datasync start-task-execution
   --task-arn 'arn:aws:datasync:region:account-id:task/task-id'
   --includes FilterType=SIMPLE_PATTERN,Value='/important_folder1|/important_folder2'
```

> **Note**
> Include filters support the wildcard (*) character only as the rightmost character in a pattern. For example, `/documents*|/code*` is supported, but `*.txt` isn't supported.

Example filters

The following examples show common filters you can use with DataSync.

> **Note**
> When creating your own filters, make sure you know the Filter limits (p. 140).

Exclude some folders from your source location

In some cases, you might exclude folders in your source location to not copy them to your destination location. For example, you might have temporary work-in-progress folders. Or you might use a NetApp system and want to exclude NetApp backup folders. In these cases, you use the following filter.

```
*/.snapshot
```

To exclude folders at any level in the file hierarchy, you can create a task to configure an exclude filter like the following.

```
*/folder-to-exclude-1|*/folder-to-exclude-2
```

To exclude folders at the top level of the source location, you can create a task to configure an exclude filter like the following.

`/top-level-folder-to-exclude-1|/top-level-folder-to-exclude-2`

Include a subset of the folders on your source location

In some cases, your source location might be a large share, and you need to transfer a subset of the folders under the root. To include specific folders, start a task execution with an include filter like the following.

`/folder-to-transfer/*`

Exclude specific file types

To exclude certain file types from the transfer, you can create a task execution with an exclude filter such as `*.temp`.

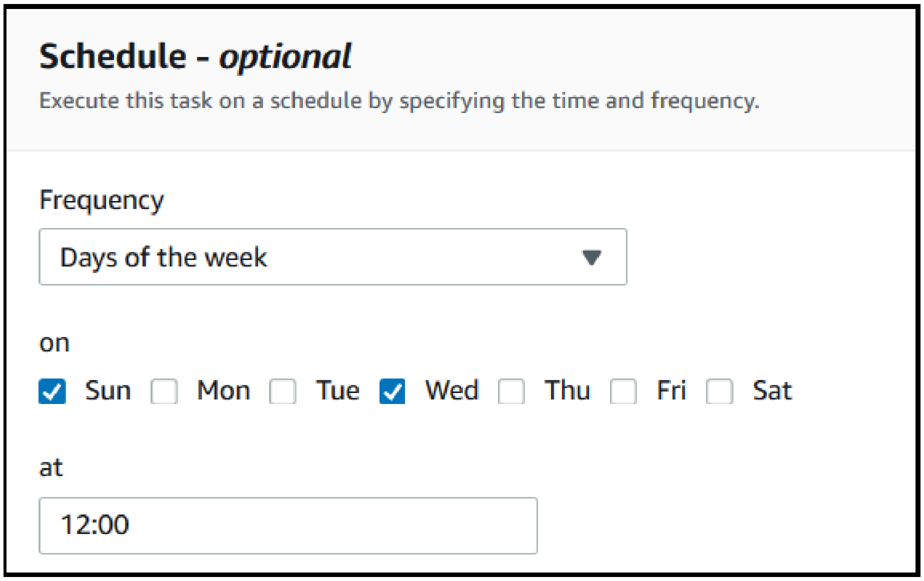
Transfer individual files you specify

To transfer a list of individual files, start a task execution with an include filter like the following: `"/folder/subfolder/file1.txt|/folder/subfolder/file2.txt|/folder/subfolder/file2.txt"`

Scheduling your DataSync task

Using task scheduling in AWS DataSync, you can periodically execute a transfer task from your source storage system to the destination.

A scheduled task automatically runs at a frequency that you configure with a minimum interval of 1 hour. For example, the following screenshot shows a configuration that runs a task every Sunday and Wednesday at 12:00 PM UTC.



Schedule - *optional*

Execute this task on a schedule by specifying the time and frequency.

Frequency

Days of the week

on

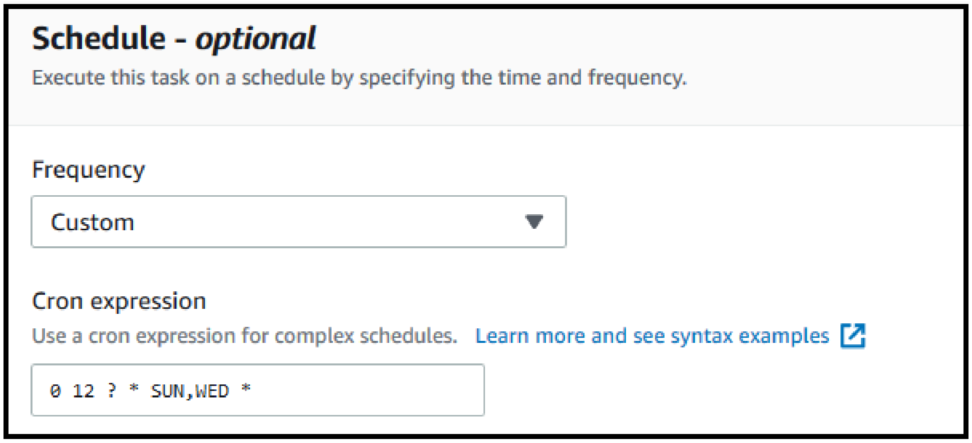
☑ Sun □ Mon □ Tue ☑ Wed □ Thu □ Fri □ Sat

at

12:00

You can also execute a task schedule using a cron expression specified in UTC time. For example, configure a task to run on every Sunday and Wednesday at 12:00 PM by using the following cron expression.

```
0 12 ? * SUN,WED *
```



Schedule - *optional*

Execute this task on a schedule by specifying the time and frequency.

Frequency

Custom ▼

Cron expression

Use a cron expression for complex schedules. Learn more and see syntax examples

```
0 12 ? * SUN,WED *
```

> **Important**
>
> Even with a cron expression, you can't schedule a task to run at an interval faster than 1 hour.

For detailed information about schedule expressions syntax, see Schedule expressions for rules in the *Amazon CloudWatch User Guide*.

Configuring a task schedule

You can configure the frequency of the task execution by using the DataSync console or API. When you create or edit a task, the following options are available for **Frequency** in the console:

- Choose **Not Scheduled** if you don't want to schedule your task to run periodically.
- Choose **Hourly** and choose the minute in the hour that the task should run. The task runs every hour on the specified minute.
- Choose **Daily** and enter the UTC time that you want the task to run, in the format HH:MM. This task runs every day at the specified time.
- Choose **Weekly** and the day of the week and enter the UTC time the task should run, in the format HH:MM. This task runs every week on the specified day at the specified time.
- Choose **Days of the week**, choose the specific day or days, and enter the UTC time that the task should run in the format HH:MM. This task runs on the days and the time that you specified.
- Choose **Custom** if you want to use a custom cron expression to run your task, with a minimum interval of 1 hour. Then enter your expression in the **Cron expression** box.

For detailed information about schedule expressions, see Schedule expressions for rules in the *Amazon CloudWatch User Guide*.

Editing a task schedule

You can configure scheduling when you initially create a task (p. 30), or you can edit a task schedule after a task is created. Use the following procedure to configure a schedule after you have created a task.

To edit a task schedule

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. On the navigation pane, choose **Tasks**, and then choose the task that you want to edit.
3. For **Actions**, choose **Edit** to open the **Edit tasks** page and expand **Schedule (optional)**.
4. In the **Schedule (optional)** section, configure your task to run on a schedule that you specify.

5. For **Frequency**, configure how frequently you want the task to run, with a minimum interval of 1 hour. For frequency configurations options, see Configuring a task schedule (p. 108).

DataSync task creation statuses

You can look at your AWS DataSync task in the AWS Management Console to understand if it's ready to run or having an issue.

Status	Description
CREATING	Depending on the locations in your task, you might see this status. For example, this status displays while DataSync mounts your Network File System (NFS) or Server Message Block (SMB) server to make sure it can connect with your location. (Note: DataSync mounts your server again once you run the task and unmounts it once the task completes.) If a task is in the **CREATING** state for more than a few minutes, your agent might be having mounting issues that are often caused by a misconfigured firewall or mistyped NFS or SMB server hostname. To understand the issue, check the response elements `ErrorCode` and `ErrorDetail` that are returned with the DescribeTask operation. You can also see Troubleshooting AWS DataSync issues (p. 141).
AVAILABLE	The task is configured properly and ready to start.
RUNNING	The task is in progress.
UNAVAILABLE	The agent that's associated with a location is offline.
QUEUED	Another task is running and using the same agent. DataSync runs tasks in the queue in series (first in, first out). For more information, see Queueing task executions (p. 109).

Starting your DataSync task

Starting a task creates a task execution. A *task execution* is an individual run of a task. Each task can have at most one task execution at a time. You can run a task with the DataSync options already configured on the task level when creating it. Alternatively, you can change the options for a specific task run and execution before you run the task. For instructions on how to start a task, see Start your task (p. 33).

> **Note**
> Each agent can execute a single task at a time.

The time that AWS DataSync spends in the **PREPARING** status depends on the number of files in both the source and destination file systems. It also depends on the performance of these file systems. When a task starts, DataSync performs a recursive directory listing to discover all files and file metadata in the source and destination file system. These listings are used to identify differences and determine what to copy, and usually takes between a few minutes to a few hours.

Queueing task executions

When you use the same agent to run multiple tasks, you can queue one task execution for each task. By using queueing, you can make tasks run in series (first in, first out) even if the agent is already running other tasks. You can set queuing either by using the DataSync console or the API.

You can queue multiple executions of the same task by using different filter settings for each task execution. You can configure filter settings for a task execution using the **Start with overrides** option when you start a task. For more information about filters, see Filtering data transferred by AWS DataSync (p. 104).

To enable queueing on the DataSync console, choose **Enabled** for **Queueing** for the option when you configure task settings. If you enable queueing and the agent is running an execution from another task or an execution using different filters, the current task's execution is automatically queued. After a task execution finishes, DataSync runs the next queued execution. If you want to remove a task execution from the queue yourself, cancel the execution.

To enable queueing by using the DataSync API, set the `TaskQueueing` property to `ENABLED`.

Working with task executions in DataSync

In AWS DataSync, a *task execution* is an individual run of a task, which includes information such as start time, end time, bytes written, and status.

After a task execution starts, you can monitor its progress, add or adjust bandwidth throttling for it, or cancel it before it completes.

Topics

Adjusting bandwidth throttling for a task execution

You can modify bandwidth throttling for a task execution using the AWS Management Console or the DataSync API. For information about using the API, see UpdateTaskExecution.

To modify bandwidth throttling

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Tasks**.
3. Select the **Task ID** for the running task that you want to monitor. Using the console, you can modify any task execution that is currently running or that is queued.
4. Choose **History** to view task execution instances.
5. Select the active task execution to be modified.
6. For **Action**, choose **Edit**.
7. In the **Edit Task Execution** dialog box that appears, choose **Use available** to remove bandwidth throttling and use all available bandwidth for the task execution.

   Choose **Set bandwidth limit (MIB/s)** to change the bandwidth limit.

   To save changes to your task execution's bandwidth limit, choose **Save changes**. The new bandwidth limit setting goes into effect on the running or queued task execution within 60 seconds.

Task execution statuses

Following, you can find information about the possible statuses (phases) a task execution might go through.

DataSync phase or status	Meaning
QUEUEING	This is the first phase of a task execution if there is another task running and it's using the same agent. For more information, see Queueing task executions (p. 109).
LAUNCHING	This is the first phase of a task execution if there is no other task running and using the same agent or if queueing isn't enabled. At this point, AWS DataSync is initializing the task execution. This status usually goes quickly, but can take up to a few minutes.
PREPARING	This is the second phase of a task execution. AWS DataSync is computing which files need to be transferred. The time that this phase takes is proportional to the number of files in the source location. It usually takes between a few minutes to a few hours, depending on both the source and destination file systems and the performance of these file systems. For more information, see Starting your DataSync task (p. 109).
TRANSFERRING	This is the third phase of a task execution. DataSync is performing the actual transfer of your data to AWS. While the DataSync is transferring files, the number of bytes and files that are transferred is updated in real time.
VERIFYING	This is the fourth and optional phase of a task execution. If the `VerifyMode` sync option is set to **POINT_IN_TIME_CONSISTENT**, DataSync performs a full data and metadata integrity verification. This verification ensures that the data in your destination is an exact copy of the data in your source location. This process requires reading back all files in the destination and can take a significant amount of time on very large volumes. If you want to skip verification, specify `VerifyMode=NONE` when configuring the task execution. Alternatively, in your task's options in the console, don't choose **Enable verification**. For more information, see How AWS DataSync verifies data integrity (p. 6).
SUCCESS	This value is returned if the data transfer is successful. If the `VerifyMode` option isn't set, this status occurs after the **TRANSFERRING** phase. Otherwise, it occurs after the **VERIFYING** phase. For more information, see Task execution (p. 5).
ERROR	This value is returned if the data transfer fails. If the `VerifyMode` option isn't set, this status occurs after the **TRANSFERRING** phase. Otherwise, it occurs after the **VERIFYING** phase.

Canceling a DataSync task execution

Using the console, you can cancel any task execution that is currently running or that is queued. You can also cancel a task execution using the API. For more information, see CancelTaskExecution.

To cancel a task execution

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the navigation pane, choose **Tasks**.
3. Select the **Task ID** for the running task that you want to monitor. The **Status** should be **Running**.
4. Choose **History** to view task execution instances.
5. Select the active task execution to be stopped.

6. For **Action**, choose **Stop**.
7. In the **Stop Task Execution** dialog box that appears, choose **Confirm** to stop the task execution.

Deleting your DataSync task

If you no longer need an AWS DataSync task, you can delete it using the AWS Management Console.

To delete a task

1. In the navigation pane, choose **Task**.
2. For **Actions**, choose **Delete**.
3. In the **Delete task** dialog, choose **Delete**.

   You can't get restore a deleted task.

Monitoring AWS DataSync

Topics

You can monitor AWS DataSync using Amazon CloudWatch, which collects and processes raw data from DataSync into readable, near real-time metrics. These statistics are retained for a period of 15 months. By default, DataSync metric data is automatically sent to CloudWatch at 5-minute periods. For more information about CloudWatch, see What are Amazon CloudWatch, CloudWatch Events, and CloudWatch logs? in the *Amazon CloudWatch User Guide*.

Accessing Amazon CloudWatch metrics for DataSync

Amazon CloudWatch provides metrics that you can use to get information about DataSync performance. You can see CloudWatch metrics for DataSync in many ways. You can view them through the CloudWatch console, or you can access them using the CloudWatch CLI or the CloudWatch API. You can also see these metrics on the task execution details page in the AWS DataSync console. For information about how to use CloudWatch metrics, see Using Amazon CloudWatch metrics in the *Amazon CloudWatch User Guide.*

DataSync CloudWatch metrics

The `AWS/DataSync` namespace includes the following metrics.

These statistics are retained for a period of 15 months.

Metric	Description
`BytesVerifiedSource`	The total number of bytes of data that are verified at the source location. Units: Bytes
`BytesPreparedSource`	The total number of bytes of data that are prepared at the source location. Unit: Bytes
`FilesVerifiedSource`	The total number of files that are verified at the source location. Unit: Count
`FilesPreparedSource`	The total number of files that are prepared at the source location. Unit: Count
`BytesVerifiedDestination`	The total number of bytes of data that are verified at the destination location.

Metric	Description
	Unit: Bytes
`BytesPreparedDestination`	The total number of bytes of data that are prepared at the destination location. Unit: Bytes
`FilesVerifiedDestination`	The total number of files that are verified at the destination location. Unit: Count
`FilesPreparedDestination`	The total number of files that are prepared at the destination location. Unit: Count
`FilesTransferred`	The actual number of files or metadata that were transferred over the network. This value is calculated and updated on an ongoing basis during the TRANSFERRING phase. It's updated periodically when each file is read from the source location and sent over the network. If failures occur during a transfer, this value can be less than `EstimatedFilesToTransfer`. This value can also be greater than `EstimatedFilesTransferred` in some cases. This element is implementation-specific for some location types, so don't use it as an indicator for a correct file number or to monitor your task execution. Unit: Count
`BytesTransferred`	The total number of bytes that are transferred over the network when the agent reads from the source location to the destination location. Unit: Bytes
`BytesWritten`	The total logical size of all files that have been transferred to the destination location. Unit: Bytes

CloudWatch events for DataSync

Amazon CloudWatch events describe changes in DataSync resources. You can set up rules to match these events and route them to one or more target functions or streams. Events are emitted on a best effort basis.

The following CloudWatch events are available for AWS DataSync.

Event	Description
State Changes for an Agent	For details, see DataSync agent statuses (p. 62).
ONLINE	The agent is configured properly and is available to use. The ONLINE status is the normal running status for an agent.
OFFLINE	The agent's VM is turned off or the agent is in an unhealthy state and has been out of contact with the service for 5 minutes or longer. When the issue

Event	Description
	that caused the unhealthy state is resolved, the agent returns to ONLINE status.
State Changes for a Location	
ADDING	DataSync is adding a location.
AVAILABLE	The location is created and is available to use.
State Changes for a Task	For details, see DataSync task creation statuses (p. 109).
CREATING	DataSync attempts to mount the Network File System (NFS) location and create the task.
RUNNING	DataSync has mounted the source and it is functioning properly.
AVAILABLE	The task is configured properly and is available to be started.
UNAVAILABLE	The task is not configured properly and is not available for use. If an agent that is associated with a source (NFS) location goes offline, the task transitions to the UNAVAILABLE status.
State Changes for a Task Execution	
LAUNCHING	DataSync is initializing the task execution.
PREPARING	DataSync is computing which files need to be transferred.
TRANSFERRING	DataSync is performing the actual transfer of your data to AWS
VERIFYING	DataSync performs a full data and metadata integrity verification to ensure that the data in your destination is an exact copy of your source.
SUCCESS	The transfer is successful.
ERROR	The sync has failed.

DataSync dimensions

DataSync metrics use the `AWS/DataSync` namespace and provide metrics for the following dimensions:

- AgentId—the unique ID of the agent.
- TaskId—the unique ID of the task. It takes the form `task-01234567890abcdef`.

Allowing DataSync to upload logs to Amazon CloudWatch log groups

You can use CloudWatch log groups to monitor and debug your tasks. To upload logs to your log group, DataSync requires a resource policy that grants sufficient permissions. When you create a task using the AWS Management Console, DataSync can automatically create the required resource policy. For more information, see Configure task settings (p. 30).

The following is an example resource policy that grants such permissions.

```
{
    "Statement": [
        {
            "Sid": "DataSyncLogsToCloudWatchLogs",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogStream"
            ],
            "Principal": {
                "Service": "datasync.amazonaws.com"
            },
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:datasync:region:account-id:task/*"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                }
            },
            "Resource": "arn:aws:logs:region:account-id:log-group:*:*"
        }
    ],
    "Version": "2012-10-17"
}
```

The policy uses condition statements to ensure that only DataSync tasks from the specified account have access to the specified CloudWatch log group. We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in these condition statements to protect against the confused deputy problem. For more information, see Cross-service confused deputy prevention (p. 128).

To specify the DataSync task or tasks, replace *`region`* with the Region code for the AWS Region where the tasks are located and replace *`account-id`* with the AWS account ID of the account that contains the tasks. To specify the CloudWatch log group, replace the same values. You can also modify the `Resource` statement to target specific log groups. For more information about using `SourceArn` and `SourceAccount`, see Global condition keys in the *IAM User Guide*.

To apply the policy, save this policy statement to a file on your local computer. Then run the following AWS Command Line Interface (AWS CLI) command to apply the resource policy:

```
aws logs put-resource-policy --policy-name trustDataSync --policy-document file://full-
path-to-policy-file
```

> **Note**
> Run this command using the same AWS account and Region that your DataSync agent is activated in.

For information about CloudWatch log groups, see Working with log groups and log streams in the *Amazon CloudWatch Logs User Guide*.

Security in AWS DataSync

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS DataSync, see AWS services in scope by compliance program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using DataSync. The following topics show you how to configure DataSync to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your DataSync resources.

Topics

Data protection in AWS DataSync

AWS DataSync securely transfers data between self-managed storage systems and AWS storage services and also between AWS storage services. How your storage data is encrypted in transit depends in part on the locations involved in the transfer.

After the transfer completes, data is encrypted at rest by the system or service that's storing the data (not DataSync).

Topics

AWS DataSync encryption in transit

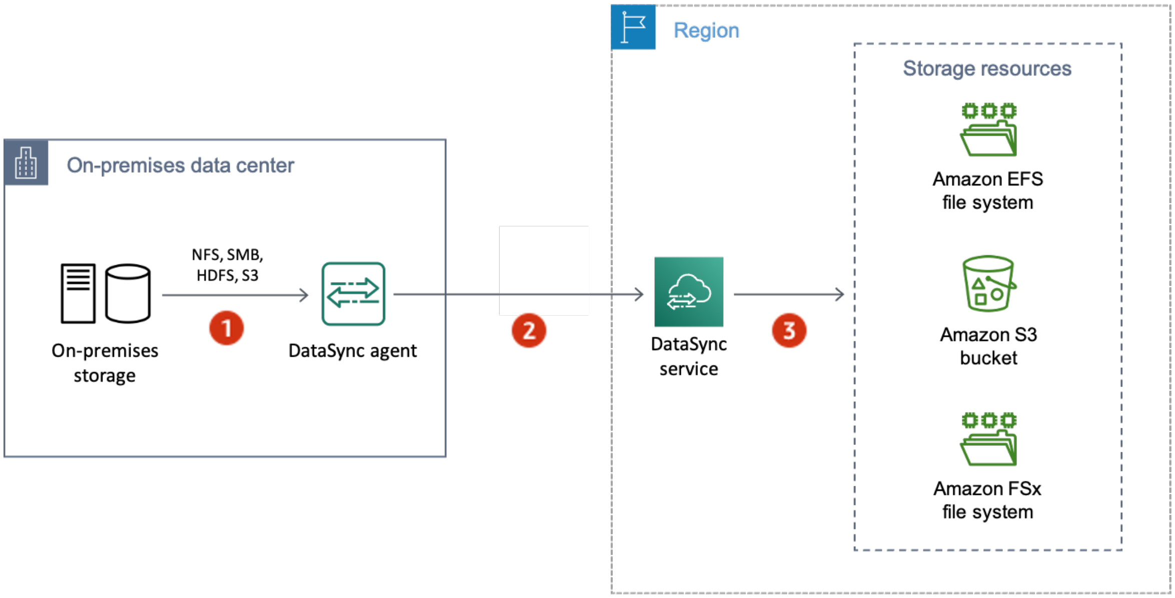
Your storage data (including metadata) is encrypted in transit, but how it's encrypted throughout the transfer depends on your source and destination locations.

When connecting with a location, DataSync uses the most secure options provided by that location's data access protocol. For example, when connecting with a Server Message Block (SMB) file system, DataSync uses the security features provided by SMB.

Network connections in a transfer

DataSync requires three network connections to copy data: a connection to read data from a source location, another to transfer data between locations, and one more to write data to a destination location.

The following diagram is an example of the network connections that DataSync uses to transfer data from an on-premises storage system to an AWS storage service. To understand where the connections happen and how data is protected as it moves through each connection, use the accompanying table.



Reference	Network connection	Description
1	Reading data from the source location	DataSync connects by using the storage system's protocol for accessing data (for example, SMB or the Amazon S3 API). For this connection, data is protected by using the security features of the storage system's protocol.
2	Transferring data between locations	For this connection, DataSync encrypts all network traffic with Transport Layer Security (TLS) 1.2.
3	Writing data to the destination location	Like it did with the source location, DataSync connects by using the storage system's protocol for accessing data. Data is again protected by using the security features of the storage system's protocol.

TLS ciphers

When transferring data between locations, DataSync uses different TLS ciphers. The TLS cipher that DataSync uses depends on the type of endpoint that's used to activate your DataSync agent.

Public or VPC endpoints

For these endpoints, DataSync uses one of the following TLS ciphers:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519)
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519)

FIPS endpoints

For FIPS endpoints, DataSync uses the following TLS cipher:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519)

AWS DataSync encryption at rest

Because AWS DataSync is a transfer service, it doesn't manage your storage data at rest. The storage services and systems that DataSync supports are responsible for protecting data in that state. However, there is some service-related data that DataSync manages at rest.

What's encrypted?

The only data that DataSync handles at rest relates to the information that it needs to complete your transfer (also known as a *task*). DataSync stores the following data with full at-rest encryption in Amazon DynamoDB:

- Task configurations (for example, details about the locations in your transfer).
- User credentials that allow your DataSync agent to authenticate with a location. These credentials are encrypted by using your agent's public keys. The agent can decrypt these keys as needed with its private keys.

For more information, see DynamoDB encryption at rest in the *Amazon DynamoDB Developer Guide*.

Key management

You can't manage the encryption keys that DataSync uses to store information in DynamoDB related to running your task. This information includes your task configurations and the credentials that agents use to authenticate with a storage location.

What's not encrypted?

Though DataSync doesn’t control how your storage data is encrypted at rest, we still recommend configuring your locations with the highest level of security that they support. For example, you can encrypt objects with Amazon S3 managed encryption keys (SSE-S3) or AWS Key Management Service (AWS KMS) keys (SSE-KMS).

Learn more about how AWS storage services manage reading and writing encrypted data:

- Amazon S3 default encryption for S3 buckets
- Amazon EFS file system encryption of data at rest
- Amazon FSx for Windows File Server encryption of data at rest
- Amazon FSx for Lustre encryption of data at rest
- Amazon FSx for OpenZFS encryption of data at rest

Internetwork traffic privacy

We recommend configuring your source and destination locations with the highest level of security that each one supports. When connecting to a location, AWS DataSync works with the most secure version of the data access protocol that the storage system uses. Additionally, consider limiting subnet traffic to known protocols and services.

DataSync secures the connection between locations—including between AWS accounts, AWS Regions, and Availability Zones—by using Transport Layer Security (TLS) 1.2.

Identity and access management in AWS DataSync

AWS uses security credentials to identify you and to grant you access to your AWS resources. You can use features of AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your AWS resources fully or in a limited way, without sharing your security credentials.

By default, IAM identities (users, groups, and roles) don't have permission to create, view, or modify AWS resources. To allow users, groups, and roles to access AWS DataSync resources and interact with the DataSync console and API, we recommend that you use an IAM policy that grants them permission to use the specific resources and API actions that they will need. You then attach the policy to the IAM identity that requires access. For an overview of the basic elements for a policy, see Overview of managing access permissions for DataSync (p. 120).

Topics

The following sections provide details on how you can use AWS Identity and Access Management (IAM) and DataSync to help secure your resources by controlling who can access them:

- Authentication (p. 124)
- Permissions (p. 125)

Overview of managing access permissions for DataSync

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

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

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

Topics

DataSync resources and operations

In DataSync, the primary resources are task, location, agent, and task execution.

These resources have unique Amazon Resource Names (ARNs) associated with them, as shown in the following table.

Resource type	ARN format
Task ARN	`arn:aws:datasync:region:account-id:task/task-id`
Location ARN	`arn:aws:datasync:region:account-id:location/location-id`
Agent ARN	`arn:aws:datasync:region:account-id:agent/agent-id`
Task execution ARN	`arn:aws:datasync:region:account-id:task/task-id/execution/exec-id`

To grant permissions for specific API operations, such as creating a task, DataSync defines a set of actions that you can specify in a permissions policy. An API operation can require permissions for more than one action. For a list of all the DataSync API actions and the resources that they apply to, see DataSync API permissions: Actions and resources (p. 129).

Understanding resource ownership

A *resource owner* is the AWS account that created the resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this behavior works:

- If you use the root account credentials of your AWS account to create a task, your AWS account is the owner of the resource (in DataSync, the resource is the task).
- If you create an IAM user in your AWS account and grant permissions to the `CreateTask` action to that user, the user can create a task. However, your AWS account, to which the user belongs, owns the task resource.

- If you create an IAM role in your AWS account with permissions to create a task, anyone who can assume the role can create a task. Your AWS account, to which the role belongs, owns the task resource.

Managing access to resources

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

> **Note**
> This section discusses using IAM in the context of DataSync. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What is IAM? in the *IAM User Guide.* For information about IAM policy syntax and descriptions, see AWS Identity and Access Management policy reference in the *IAM User Guide.*

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

Topics

Identity-based policies (IAM policies)

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

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create a DataSync resource, such as a task, location, agent, or task execution.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
  1. The Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. The Account A administrator attaches a trust policy to the role that identifies Account B as the principal who can assume the role.

     To grant an AWS service permissions to assume the role, the Account A administrator can specify an AWS service as the principal in the trust policy.
  3. The Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A.

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

The following example policy grants permissions to all `List*` actions on all resources. This action is a read-only action. Thus, the policy doesn't allow the user to change the state of the resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllListActionsOnAllResources",
            "Effect": "Allow",
            "Action": [
                "datasync:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information about using identity-based policies with DataSync, see Using identity-based policies (IAM policies) for DataSync (p. 125). For more information about users, groups, roles, and permissions, see Identities (users, groups, and roles) in the *IAM User Guide*.

Resource-based policies

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

Specifying policy elements: Actions, effects, resources, and principals

For each DataSync resource (see DataSync API permissions: Actions and resources (p. 129)), the service defines a set of API operations (see Actions). To grant permissions for these API operations, DataSync defines a set of actions that you can specify in a policy. For example, for the DataSync resource, the following actions are defined: `CreateTask`, `DeleteTask`, and `DescribeTask`. Performing an API operation can require permissions for more than one action.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For DataSync resources, you can use the wildcard character `(*)` in IAM policies. For more information, see DataSync resources and operations (p. 121).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, depending on the specified `Effect` element, the `datasync:CreateTask` permission allows or denies the user permissions to perform the DataSync `CreateTask` operation.
- **Effect** – You specify the effect when the user requests the specific action—this effect can be either `Allow` or `Deny`. If you don't explicitly grant access to (`Allow`) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants that user access. For more information, see Authorization in the *IAM User Guide*.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). DataSync doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS Identity and Access Management policy reference in the *IAM User Guide*.

For a table showing all of the DataSync API actions, see DataSync API permissions: Actions and resources (p. 129).

Specifying conditions in a policy

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

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

Controlling access

In this section, you can find information about how to control access to AWS resources.

Authentication

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

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

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. DataSync supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.
- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:
  - **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
  - **AWS service access** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Permissions

You can have valid credentials to authenticate your requests, but unless you have permissions, you cannot create or access DataSync resources. For example, you must have permissions to create a task in DataSync.

The following sections provide an overview and describe how to manage permissions for DataSync.

- Overview of managing access permissions for DataSync (p. 120)
- Identity-based policies (IAM policies) (p. 122)

Using identity-based policies (IAM policies) for DataSync

An account administrator can attach identity-based policies to IAM identities (that is, users, groups, and roles). You can also attach identity-based policies to service roles.

This topic provides examples of identity-based policies that you can use to grant permissions to IAM identities.

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your DataSync resources. For more information, see Overview of managing access permissions for DataSync (p. 120).

The sections in this topic cover the following:

- AWS managed policies for DataSync (p. 126)
- Permissions required to use the DataSync console (p. 126)
- Customer managed policy examples (p. 127)

The following shows an example of a policy that grants permissions to use certain DataSync actions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsSpecifiedActionsOnAllTasks",
            "Effect": "Allow",
            "Action": [
                "datasync:DescribeTask",
                "datasync:ListTasks"
            ],
            "Resource": "arn:aws:datasync:us-east-2:111222333444:task/*"
        },
}
```

The policy has one statement (note the `Action` and `Resource` elements in the statement) that does the following:

- The statement grants permissions to perform two DataSync actions (`datasync:DescribeTask` and `datasync:ListTasks`) on certain task resources by using an *Amazon Resource Name (ARN)*.
- In this statement, the task ARN specifies a wildcard character (`*`) because the IAM user, group, or role is allowed to perform the two actions on all tasks. To limit permissions for the actions to a specific task, specify the task ID in the ARN instead of the wildcard character.

AWS managed policies for DataSync

AWS creates and administers standalone IAM policies. These managed policies grant permissions for common use cases so that you can avoid investigating what permissions you need. For more information, see AWS managed policies in the *IAM User Guide*.

The managed policies that are created by AWS grant the required permissions for common use cases. You can attach these policies to your IAM users, groups, and roles, based on the access that they need to DataSync:

The following AWS managed policies, which you can attach to users in your account, are specific to DataSync:

- **AWSDataSyncReadOnlyAccess** – Provides read-only access to AWS DataSync.
- **AWSDataSyncFullAccess** – Provides full access to AWS DataSync and minimal access to its dependencies.

> **Note**
> You can review these managed policies by signing in to the IAM console and searching for specific policies there.

You can also create your own custom IAM policies to allow permissions for AWS DataSync API actions. You can attach these custom policies to the IAM users, groups, or roles that require those permissions. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

Permissions required to use the DataSync console

To use the DataSync console, you must have AWSDataSyncFullAccess permissions.

Customer managed policy examples

In this section, you can find example user policies that grant permissions for various DataSync actions. These policies work when you are using the AWS SDKs and the AWS Command Line Interface (AWS CLI). When you are using the console, you must grant additional permissions specific to the console, which is discussed in Permissions required to use the DataSync console (p. 126).

> **Note**
> All of these examples use fictitious account IDs and resource IDs.

Topics

Example 1: Create a trust relationship that allows DataSync to access your Amazon S3 bucket

The following is an example of a trust policy that allows DataSync to assume an IAM role. This role allows DataSync to access an Amazon S3 bucket. To prevent the cross-service confused deputy problem (p. 128), we recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in the policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "datasync.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:datasync:us-east-2:123456789012:*"
                }
            }
        }
    ]
}
```

Example 2: Allow DataSync to read and write to your Amazon S3 bucket

The following example policy grants DataSync the minimum permissions to read and write data to your S3 bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn"
        },
        {
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListMultipartUploadParts",
                "s3:GetObjectTagging",
                "s3:PutObjectTagging",
                "s3:PutObject"
              ],
            "Effect": "Allow",
            "Resource": "YourS3BucketArn/*"
        }
    ]
}
```

Example 3: Allow DataSync to upload logs to CloudWatch log groups

DataSync requires permissions to be able to upload logs to your Amazon CloudWatch log groups. You can use CloudWatch log groups to monitor and debug your tasks.

For an example of an IAM policy that grants such permissions, see Allowing DataSync to upload logs to Amazon CloudWatch log groups (p. 115).

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that AWS DataSync gives another service to the resource. If you use both global condition context keys and the `aws:SourceArn` value contains the account ID, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want any resource in that account to be associated with the cross-service use.

The value of `aws:SourceArn` must include the DataSync location ARN with which DataSync is allowed to assume the IAM role.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` key with the full ARN of the resource. If you don't know the full ARN or if you're specifying multiple resources, use wildcard characters (`*`) for the unknown portions. Here are some examples of how to do this for DataSync:

- To limit the trust policy to an existing DataSync location, include the full location ARN in the policy. DataSync will assume the IAM role only when dealing with that particular location.
- When creating an Amazon S3 location for DataSync, you don't know the location's ARN. In these scenarios, use the following format for the `aws:SourceArn` key: `arn:aws:datasync:us-east-2:123456789012:*`. This format validates the partition (`aws`), account ID, and Region.

The following full example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in to prevent the confused deputy problem with DataSync.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "datasync.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:datasync:us-east-2:123456789012:*"
                }
            }
        }
    ]
}
```

For more example policies that show how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys with DataSync, see the following topics:

- Create a trust relationship that allows DataSync to access your Amazon S3 bucket (p. 127)
- Configure an IAM role to access your Amazon S3 bucket (p. 91)

DataSync API permissions: Actions and resources

When creating IAM policies, this page can help you understand the relationship between AWS DataSync API operations, the corresponding actions that you can grant permissions to perform, and the AWS resources for which you can grant the permissions.

In general, here's how you add DataSync permissions to your policy:

- Specify an action in the `Action` element. The value includes a `datasync:` prefix and the API operation name. For example, `datasync:CreateTask`.
- Specify an AWS resource related to the action in the `Resource` element.

You can also use AWS condition keys in your DataSync policies. For a complete list of AWS keys, see Available keys in the *IAM User Guide*.

For a list of DataSync resources and their Amazon Resource Name (ARN) formats, see DataSync resources and operations (p. 121).

DataSync API operations and corresponding actions

CancelTaskExecution

Action: `datasync:CancelTaskExecution`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*`/execution/`*`exec-id`*

CreateAgent

Action: `datasync:CreateAgent`

Resources:

- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:subnet/`*`subnet-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:security-group/`*`security-group-id`*

CreateLocationEfs

Action: `datasync:CreateLocationEfs`

Resources:

- `arn:aws:elasticfilesystem:`*`region`*`:`*`account-id`*`:file-system/`*`file-system-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:subnet/`*`subnet-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:security-group/`*`security-group-id`*

CreateLocationFsxLustre

Action: `datasync:CreateLocationFsxLustre`

Resources:

- `arn:aws:fsx:`*`region`*`:`*`account-id`*`:file-system/`*`file-system-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:security-group/`*`security-group-id`*

CreateLocationFsxOpenZfs

Action: `datasync:CreateLocationFsxOpenZfs`

Resources:

- `arn:aws:fsx:`*`region`*`:`*`account-id`*`:volume/`*`file-system-id`*`/`*`volume-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:security-group/`*`security-group-id`*

CreateLocationFsxWindows

Action: `datasync:CreateLocationFsxWindows`

Resources:

- `arn:aws:fsx:`*`region`*`:`*`account-id`*`:file-system/`*`file-system-id`*
- `arn:aws:ec2:`*`region`*`:`*`account-id`*`:security-group/`*`security-group-id`*

CreateLocationHdfs

Action: `dataSync:CreateLocationHdfs`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*

CreateLocationNfs

Action: `datasync:CreateLocationNfs`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*

CreateLocationObjectStorage

Action: `dataSync:CreateLocationObjectStorage`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*

CreateLocationS3

Action: `datasync:CreateLocationS3`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:s3:::`*`bucket-name`*

- `arn:aws:iam::account-id:role/role-name`

CreateLocationSmb

Action: `datasync:CreateLocationSmb`

Resource: `arn:aws:datasync:region:account-id:agent/agent-id`

CreateTask

Action: `datasync:CreateTask`

Resources:

- `arn:aws:logs:region:account-id:log-group:log-group-name`
- `arn:aws:datasync:region:account-id:location/location-id` (source location)
- `arn:aws:datasync:region:account-id:location/location-id` (destination location)

DeleteAgent

Action: `datasync:DeleteAgent`

Resource: `arn:aws:datasync:region:account-id:agent/agent-id`

DeleteLocation

Action: `datasync:DeleteLocation`

Resource: `arn:aws:datasync:region:account-id:location/location-id`

DeleteTask

Action: `datasync:DeleteTask`

Resource: `arn:aws:datasync:region:account-id:task/task-id`

DescribeAgent

Action: `datasync:DescribeAgent`

Resource: `arn:aws:datasync:region:account-id:agent/agent-id`

DescribeLocationEfs

Action: `datasync:DescribeLocationEfs`

Resource: `arn:aws:datasync:region:account-id:location/location-id`

DescribeLocationFsxLustre

Action: `datasync:DescribeLocationFsxLustre`

Resource: `arn:aws:datasync:region:account-id:location/location-id`

DescribeLocationFsxOpenZfs

Action: `datasync:DescribeLocationFsxOpenZfs`

Resource: `arn:aws:datasync:region:account-id:location/location-id`

DescribeLocationFsxWindows

Action: `datasync:DescribeLocationFsxWindows`

Resource: `arn:aws:datasync:region:account-id:location/location-id`

DescribeLocationHdfs

Action: `datasync:DescribeLocationHdfs`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

DescribeLocationNfs

Action: `datasync:DescribeLocationNfs`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

DescribeLocationObjectStorage

Action: `datasync:DescribeLocationObjectStorage`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

DescribeLocationS3

Action: `datasync:DescribeLocationS3`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

DescribeLocationSmb

Action: `datasync:DescribeLocationSmb`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

DescribeTask

Action: `datasync:DescribeTask`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*

DescribeTaskExecution

Action: `datasync:DescribeTaskExecution`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*`/execution/`*`exec-id`*

ListAgents

Action: `datasync:ListAgents`

Resource: `None`

ListLocations

Action: `datasync:ListLocations`

Resource: `None`

ListTagsForResource

Action: `datasync:ListTagsForResource`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

ListTaskExecutions

Action: `datasync:ListTaskExecutions`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*

ListTasks

Action: `datasync:ListTasks`

Resource: `None`

StartTaskExecution

Action: `datasync:StartTaskExecution`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*

TagResource

Action: `datasync:TagResource`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UntagResource

Action: `datasync:UntagResource`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:task/`*`task-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UpdateAgent

Action: `datasync:UpdateAgent`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*

UpdateLocationHdfs

Action: `datasync:UpdateLocationHdfs`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UpdateLocationNfs

Action: `datasync:UpdateLocationNfs`

Resource: `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UpdateLocationObjectStorage

Action: `datasync:UpdateLocationObjectStorage`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UpdateLocationSmb

Action: `datasync:UpdateLocationSmb`

Resources:

- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:agent/`*`agent-id`*
- `arn:aws:datasync:`*`region`*`:`*`account-id`*`:location/`*`location-id`*

UpdateTask

Action: `datasync:UpdateTask`

Resources:

- `arn:aws:datasync:region:account-id:task/task-id`
- `arn:aws:logs:region:account-id:log-group:log-group-name`

UpdateTaskExecution

Action: `datasync:UpdateTaskExecution`

Resource: `arn:aws:datasync:region:account-id:task/task-id/execution/exec-id`

Related topics

- Permissions (p. 125)
- Customer managed policy examples (p. 127)

Logging AWS DataSync API calls with AWS CloudTrail

AWS DataSync is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS DataSync. CloudTrail captures all API calls for AWS DataSync as events. The calls captured include calls from the AWS DataSync console and code calls to the AWS DataSync API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS DataSync. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS DataSync, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Working with AWS DataSync information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS DataSync, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your AWS account, including events for AWS DataSync, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations

- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

All DataSync actions are logged by CloudTrail. (For more information, see the DataSync API reference.)

For example, calls to the `CreateAgent`, `CreateTask` and `ListLocations` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see CloudTrail userIdentity element in the *AWS CloudTrail User Guide.*

Understanding AWS DataSync log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateTask` action.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJOERGY7LS5PKXTMXO",
        "arn": "arn:aws:iam::123456789012:user/user1",
        "accountId": "123456789012",
        "accessKeyId": "access key",
        "userName": "user1",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-12-13T14:56:46Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2018-12-13T14:57:02Z",
    "eventSource": "datasync.amazonaws.com",
    "eventName": "CreateTask",
    "awsRegion": "ap-southeast-1",
    "sourceIPAddress": "12.345.123.45",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "cloudWatchLogGroupArn": "arn:aws:logs:ap-southeast-1:123456789012:log-
group:MyLogGroup",
        "name": "MyTask-NTIzMzY1",
        "tags": [],
        "destinationLocationArn": "arn:aws:datasync:ap-southeast-1:123456789012:location/
loc-020c33c5d9966f40a",
        "options": {
            "bytesPerSecond": -1,
            "verifyMode": "POINT_IN_TIME_CONSISTENT",
            "uid": "INT_VALUE",
            "posixPermissions": "PRESERVE",
            "mtime": "PRESERVE",
            "gid": "INT_VALUE",
            "preserveDevices": "NONE",
            "preserveDeletedFiles": "REMOVE",
            "atime": "BEST_EFFORT"
        },
        "sourceLocationArn": "arn:aws:datasync:ap-southeast-1:123456789012:location/
loc-04aaa9c609812135d"
    },
    "responseElements": {
        "taskArn": "arn:aws:datasync:ap-southeast-1:123456789012:task/
task-00e5db3f3f41f6cd2"
    },
    "requestID": "5890e03c-fee7-11e8-8b63-0b409054d4dc",
    "eventID": "e5f59b6a-05e6-4412-bd56-440d872e90e9",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Compliance validation for AWS DataSync

Third-party auditors assess the security and compliance of AWS DataSync as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using DataSync is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. If your use of DataSync is subject to compliance with standards such as HIPAA, PCI, or FedRAMP, AWS provides resources to help:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS DataSync

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

> **Note**
> If an Availability Zone you're migrating data to or from does fail while you're running a DataSync task, the task also will fail.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in AWS DataSync

As a managed service, AWS DataSync is protected by the AWS global network security procedures. For more information, see AWS Best Practices for Security, Identity, and Compliance.

You use AWS published API calls to access DataSync through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS DataSync quotas and limits

Following, you can find information on AWS DataSync resources and their quotas and limits.

Topics

Task quotas

These are the quotas on tasks for each AWS account in an AWS Region.

Resource	Quota	Can quota be increased?
Maximum number of tasks you can create	100	Yes
Maximum number of files or objects per task when transferring data between self-managed storage and AWS services	50 million **Important** For tasks that transfer more than 20 million files or objects, make sure that you allocate a minimum of 64 GB of RAM to the virtual machine (VM). For minimum resource requirements for DataSync, see Virtual machine requirements (p. 10).	Yes **Note** As an alternative to requesting an increase, you can create tasks on different subdirectories using include and exclude filters. For more information about using filters, see Filtering the data

Resource	Quota	Can quota be increased?
		transferred by AWS DataSync.
Maximum number of files or objects per task when transferring data between AWS storage services	25 million	Yes **Note** As an alternative to requesting an increase, you can create tasks on different subdirectories using include/exclude filters. For more information about using filters, see Filtering the data transferred by AWS DataSync.
Maximum number of files per task when running DataSync on an AWS Snowcone device	200,000	No
Maximum throughput per task	10 Gbps	No

Task execution quotas

These are the quotas on task executions for each AWS account in an AWS Region.

Resource	Quota
Number of days task execution history is retained	30

File system limits

These are the file system limits for DataSync.

If the storage systems at the source and destination locations have higher limits on the lengths of the total path and path components (file names, directories, and subdirectories), DataSync might not be able to access some objects on those systems.

Description	Limit
Maximum total file path length	4,096 bytes
Maximum file path component (file name, directory, or subdirectory) length	255 bytes
Maximum length of Windows domain	253 characters
Maximum length of server hostname	255 characters
Maximum Amazon S3 object name length	1,024 UTF-8 characters

Filter limits

These are the limits on DataSync filters per task.

Filter	Limit
Maximum number of characters in a filter string	102,400 characters

Request a quota increase

You can request an increase for some DataSync quotas (for example, some task quotas can go up). Increases aren't granted right away, so it might take a couple of days for them to take effect.

To request a quota increase

1. Open the AWS Support Center page, sign in if necessary, and then choose **Create case**.
2. For **Create case**, choose **Service limit increase**.
3. For **Limit type**, choose **DataSync**.
4. For **Region**, select your AWS Region, and for **Limit**, select the quota that you want to increase.
5. Fill in the case description, and then choose your preferred method of contact.

   If you need to increase a different quota, fill out a separate request.

Troubleshooting AWS DataSync issues

Use the following information to troubleshoot AWS DataSync issues.

I need DataSync to use a specific NFS or SMB version to mount my share

DataSync automatically selects the Network File System (NFS) or Server Message Block (SMB) version that is used to access your location. If you need DataSync to use a specific version, use the DataSync API, console, or the AWS CLI.

Action to take

Do the following with the API:

- For NFS, use the optional `Version` parameter for the CreateLocationNfs (p. 192) API operation.
- For SMB, use the optional `Version` parameter for the CreateLocationSmb (p. 205) API operation.

The following AWS CLI commands create an NFS source location and cause DataSync to use NFS version 4.0. Specify the `subdirectory` option with forward slashes, for example `/path/to/folder`.

```
$ aws datasync create-location-nfs --server-hostname
                           your-server-address --on-prem-config
                           AgentArns=your-agent-arns
                           --subdirectory nfs-export-path
                           --mount-options Version=NFS4_0
```

The following AWS CLI commands create an SMB source location and cause DataSync to use SMB version 3. Specify the `subdirectory` option with forward slashes, for example `/path/to/folder`.

```
$ aws datasync create-location-smb --server-hostname
                           your-server-address --on-prem-config
                           AgentArns=your-agent-arns
                           --subdirectory smb-export-path
                           --mount-options Version=SMB3
```

What does the "Failed to retrieve agent activation key" error mean?

When you are activating your DataSync agent, the agent connects to the specified endpoint to request an activation key. You can get this error in non-VPC endpoint use cases. For example, when your agent is deployed on-premises and your firewall settings block the connection. You can also get this error if your agent is deployed as an Amazon EC2 instance and the security groups are locked down.

Action to take

Verify that your security group is set up to allow your agent to connect to the VPC endpoint and that you have allowed the required ports. For information about required ports, see Network requirements (p. 10).

Also, check your firewall and router settings and make sure that they allow communication with endpoints in AWS. For information about endpoint communication, see Network requirements when using public service endpoints or FIPS endpoints (p. 14).

I can't activate an agent I created using a VPC endpoint

If you are having issues when you are activating an agent that is created using a VPC endpoint, open a support channel against your VPC endpoint elastic network interface. For information about Support Channel, see Enabling AWS Support to help troubleshoot your running agent (p. 70).

My task status is unavailable and indicates a mount error

When you create a task, your task status might transition from **CREATING** to **UNAVAILABLE** when the agent that you chose can't mount the location that you specified during configuration.

Action to take

First, make sure that the NFS server and export that you specified are both valid. If they aren't, delete the task, create a new one using the correct NFS server, and then export. For more information, see Creating an NFS location (p. 39).

If the NFS server and export are both valid, it generally indicates one of two things. Either a firewall is preventing the agent from mounting the NFS server, or the NFS server isn't configured to allow the agent to mount it.

Make sure that there is no firewall between the agent and the NFS server. Then make sure that the NFS server is configured to allow the agent to mount the export end specified in the task. For information about network and firewall requirements, see Network requirements (p. 10).

If you perform these actions and the agent still can't mount the NFS server and export, open a support channel and engage AWS Support. For information about how to open a support channel, see Enabling AWS Support to help troubleshoot your running agent (p. 70).

My task failed with an input/output error message

You can get an input/output error message if your storage system fails I/O requests from the DataSync agent. Common reasons for this include a server disk failure, changes to your firewall configuration, or a network router failure.

If the error involves an NFS server or Hadoop Distributed File System (HDFS) cluster, use the following steps to resolve the error.

Action to take (NFS)

First, check your NFS server's logs and metrics to determine if the problem started on the NFS server. If yes, resolve that issue.

Next, check that your network configuration hasn't changed. To check if the NFS server is configured correctly and that DataSync can access it, do the following:

1. Set up another NFS client on the same network subnet as the agent.
2. Mount your share on that client.
3. Validate that the client can read and write to the share successfully.

Action to take (HDFS)

Make sure that your HDFS cluster allows the agent to communicate with the cluster's NameNode and DataNode ports. In most clusters, you can find the port numbers the cluster uses in the following configuration files.

1. To find the NameNode port, look in the `core-site.xml` file under the `fs.default` or `fs.default.name` property (depending on the Hadoop distribution).
2. To find the DataNode port, look in the `hdfs-site.xml` file under the `dfs.datanode.address` property.

My task is stuck in launching status

Your task execution can become stuck in **LAUNCHING** status when DataSync can't instruct the specified source agent to begin a task. This issue usually occurs because the agent either is powered off or has lost network connectivity.

Action to take

Make sure that the agent is connected and the status is **ONLINE**. If the status is **OFFLINE**, then the agent is not connected. For information about how to test network connectivity, see Testing your agent connection to DataSync endpoints (p. 67).

Next, make sure that your agent is powered on. If it isn't, power it on.

If the agent is powered on and the task is still stuck in **LAUNCHING** status, then a network connectivity problem between the agent and DataSync is the most likely issue. Check your network and firewall settings to make sure that the agent can connect to DataSync.

If you perform these actions and the issue isn't resolved, open a support channel and engage AWS Support. For information about how to open a support channel, see Enabling AWS Support to help troubleshoot your running agent (p. 70).

My task failed with a permissions denied error message

You can get a "permissions denied" error message if you configure your NFS server with `root_squash` or `all_squash` enabled and your files don't have all read access.

Action to take

To fix this issue, you can configure the NFS export with `no_root_squash`. Or you can make sure that the permissions for all of the files that you want to transfer allow read access for all users. Doing either enables the agent to read the files. For the agent to access directories, you must additionally enable all-execute access.

To make sure that the directory can be mounted, first connect to any computer that has the same network configuration as your agent. Then run the following CLI command.

```
mount -t nfs -o nfsvers=<your nfs server version> <your nfs server name>:<the nfs export path you specified> <a new test folder on your computer>
```

If you perform these actions and the issue isn't resolved, contact AWS Support.

My task has had a preparing status for a long time

The time DataSync spends in the **PREPARING** status depends on the number of files in both the source and destination file systems, and the performance of these file systems. When a task starts, DataSync performs a recursive directory listing to discover all files and file metadata in the source and destination file system. These listings are used to identify differences and determine what to copy. This process usually takes between a few minutes to a few hours. For more information, see Starting your DataSync task (p. 109).

Action to take

You shouldn't have to do anything. Continue to wait for the **PREPARING** status to change to **TRANSFERRING**. If the status still doesn't change, contact AWS Support.

How long does it take to verify a task I've run?

The time DataSync spends in the **VERIFYING** status depends on a number of factors. These are the number of files, the total size of all files in the source and destination file systems, and the performance of these file systems. By default, **Verification mode** is enabled in the options setting. The verification DataSync performs includes an SHA256 checksum on all file content and an exact comparison of all file metadata.

Action to take

You shouldn't have to do anything. Continue to wait for the **VERIFYING** status to complete. If the status still doesn't change, contact AWS Support.

My storage cost is higher than I expected

If your storage cost is higher then expected, it might be due to one or more of the following reasons:

- DataSync uses the Amazon S3 multipart upload feature to upload objects to Amazon S3. This approach can result in unexpected storage charges for uploads that don't successfully complete.
- Object versioning might be enabled on your S3 bucket. Object versioning results in Amazon S3 storing multiple copies of objects that have the same name.

Action to take

In these cases, you can take the following steps:

- If the issue relates to multipart uploads, configure a policy for multipart uploads for your S3 bucket to clean up incomplete multipart uploads to reduce storage cost. For more information, see the AWS blog post S3 Lifecycle Management Update - Support for Multipart Uploads and Delete Markers.
- If the issue relates to object versioning, verify whether object versioning is enabled for your Amazon S3 bucket. If versioning is enabled, turn it off.

If you perform these actions and the issue isn't resolved, contact AWS Support. For information about how to contact AWS Support, see Getting started with AWS Support.

I don't know what's going on with my agent. Can someone help me?

If you're having issues with your deployed DataSync agent that you can't solve, AWS Support can assist you.

For instructions on how to open a support channel, see Enabling AWS Support to help troubleshoot your running agent (p. 70).

How do I connect to an Amazon EC2 agent's local console?

Make sure the Amazon EC2 instance's security group allows access with SSH (TCP port 22), then log in with the following command:

ssh -i PRIVATE-KEY admin@AGENT-PUBLIC-DNS-NAME

- The user name is **`admin`**.
- The `PRIVATE-KEY` value is the `.pem` file containing the private certificate of the Amazon EC2 key pair that you used to launch the instance. For more information, see retrieve the public key from the private key in the *Amazon EC2 User Guide for Linux Instances.*
- The `AGENT-PUBLIC-DNS-NAME` value is the public DNS name of your agent. You can find this public DNS name by choosing the instance in the Amazon EC2 console and going to the **Description** tab.

For more information on connecting to the Amazon EC2 instance, see Connect to your instance in the *Amazon EC2 User Guide for Linux Instances.*

My task fails when transferring to an S3 bucket in another AWS account.

Unlike DataSync transfers between resources in the same AWS account, copying data to an S3 bucket in a different AWS account requires some extra steps.

- **If your DataSync task fails with an error related to S3 bucket permissions**: When creating the task, make sure you're logged in to the AWS Management Console using the same IAM user name (or role) which you specified in your destination S3 bucket's policy. (Note: This isn't the IAM role that gives DataSync permission to write to the S3 bucket.)

- **If you're also copying data to a bucket in another AWS Region and get an S3 endpoint connection error**: Create the DataSync task in the same Region as the destination S3 bucket.

For complete instructions on cross-account transfers with Amazon S3, see the following tutorials:

- Transferring data from on-premises storage to Amazon S3 in a different AWS account (p. 147)
- Transferring data from Amazon S3 to Amazon S3 in a different AWS account (p. 152)

My task fails when transferring from a Google Cloud Storage bucket

Because DataSync communicates with Google Cloud Storage using the Amazon S3 API, there's a limitation that may cause your DataSync task to fail if you try to copy object tags. The following message related to the issue appears in your CloudWatch Logs:

[WARN] Failed to read metadata for file /*`your-bucket`*/*`your-object`*: S3 Get Object Tagging Failed: proceeding without tagging

To prevent this, deselect the **Copy object tags** option when configuring your task settings.

AWS DataSync tutorials

These tutorials walk you through some real-world scenarios with AWS DataSync.

Topics

Tutorial: Transferring data from on-premises storage to Amazon S3 in a different AWS account

When using AWS DataSync with on-premises storage, you typically copy data to an AWS storage service that belongs to the same AWS account as your DataSync agent. There are situations, however, where you might need to transfer data to an Amazon S3 bucket that's associated with a different account.

> **Important**
> Copying data across AWS accounts by using the methods in this tutorial works only when Amazon S3 is one of the DataSync locations.

Overview

In this tutorial, you'll learn how AWS Identity and Access Management (IAM) and the AWS Command Line Interface (AWS CLI) can help you create DataSync tasks that transfer data from on-premises storage to an S3 bucket in a different AWS account.

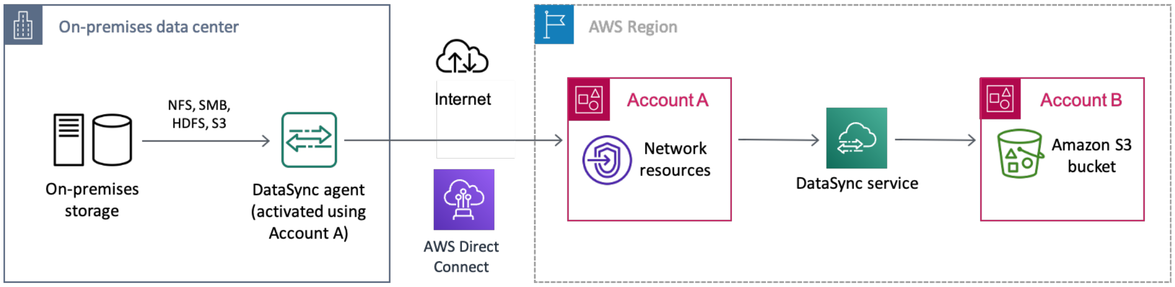
Here's what this kind of scenario can look like:

- **Account A**: The AWS account that you use for managing network resources. The endpoint that you activate the DataSync agent with also belongs to this account.

  > **Note**
  > The steps in this tutorial apply to any type of endpoint (p. 25) that you activate your agent with.

- **Account B**: The AWS account for the S3 bucket that you want to copy data to.

The following diagram illustrates this scenario.



Prerequisites

Before you begin the IAM work to facilitate the cross-account transfer, do the following if you already haven't:

1. Configure your network (p. 10) so that your on-premises storage system can connect with AWS.
2. Deploy and activate your DataSync agent (p. 55) with Account A.
3. Create a DataSync source location (p. 72) with Account A for the on-premises storage system that you're copying data from.
4. Set up the AWS CLI with Account A. You'll need the AWS CLI to create the DataSync destination location for the S3 bucket in Account B.

Step 1: Create an IAM role for DataSync in Account A

You need an IAM role that gives DataSync permission to write to the S3 bucket in Account B.

When you create a location for a bucket, DataSync can automatically create and assume a role with the right permissions to access that bucket. Since you're transferring across accounts, you must create the role manually.

For more information, see Creating a role for an AWS service (console) in the *IAM User Guide*.

Create the IAM role

Create a role with DataSync as the trusted entity.

To create the IAM role

1. Log in to the AWS Management Console with Account A.
2. Open the IAM console at https://console.aws.amazon.com/iam/.
3. In the left navigation pane, under **Access management**, choose **Roles**, and then choose **Create role**.
4. On the **Select trusted entity** page, for **Trusted entity type**, choose **AWS service**.
5. For **Use case**, choose **DataSync** in the dropdown list and select **DataSync**. Choose **Next**.
6. On the **Add permissions** page, choose **Next**.
7. Give your role a name and choose **Create role**.

Attach a custom policy to the IAM role

The IAM role needs a policy that allows DataSync to write to your S3 bucket in Account B.

To attach a custom policy to the IAM role

1. On the **Roles** page of the IAM console, search for the role that you just created and choose its name.
2. On the role's details page, choose the **Permissions** tab. Choose **Add permissions** then **Create inline policy**.
3. Choose the **JSON** tab and do the following:

   a. Paste the following JSON into the policy editor:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::account-b-bucket"
    },
    {
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::account-b-bucket/*"
    }
  ]
}
```

   b. Replace *`account-b-bucket`* with the name of the S3 bucket in Account B.

4. Choose **Review policy**.
5. Give your policy a name and choose **Create policy**.

Step 2: Disable ACLs for your S3 bucket in Account B

It's important that all the data that you copy to the S3 bucket belongs to Account B. To ensure that Account B is the owner of the data, disable the bucket's access control lists (ACLs). For more information, see Controlling ownership of objects and disabling ACLs for your bucket in the *Amazon S3 User Guide*.

To disable ACLs for an S3 bucket

1. In the AWS Management Console, switch over to Account B.
2. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. In the left navigation pane, choose **Buckets**.
4. In the **Buckets** list, choose the S3 bucket that you're transferring data to.
5. On the bucket's detail page, choose the **Permissions** tab.
6. Under **Object Ownership**, choose **Edit**.
7. If it isn't already selected, choose the **ACLs disabled (recommended)** option.
8. Choose **Save changes**.

Step 3: Update the S3 bucket policy in Account B

In Account B, modify the S3 bucket policy to allow access to the IAM role that you created for DataSync in Account A.

The updated policy (provided to you in the following instructions) includes two principals:

- The first principal specifies the IAM role that you created in Account A that allows DataSync to write to the S3 bucket.
- The second principal specifies the IAM user name for Account A, which allows you to create the DataSync destination for the S3 bucket by using the AWS CLI (you'll do this in Step 4).

Note

If you log in to the console and access the AWS CLI using an IAM role, specify that role instead of a user name for the second principal.

To update the S3 bucket policy

1. While still in the S3 console and using Account B, choose the S3 bucket that you're copying data to.
2. On the bucket's detail page, choose the **Permissions** tab.
3. Under **Bucket policy**, choose **Edit** and do the following to modify your S3 bucket policy:

   a. Update what's in the editor to include the following policy statements:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "DataSyncCreateS3LocationAndTaskAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-a-id:role/name-of-role"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::account-b-bucket",
        "arn:aws:s3:::account-b-bucket/*"
      ]
    },
    {
      "Sid": "DataSyncCreateS3Location",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-a-id:user/name-of-user"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::account-b-bucket"
    }
  ]
}
```

   b. Replace *`account-a-id`* with the AWS account number of Account A.

   c. Replace *`name-of-role`* with the IAM role that you created for DataSync in Account A (back in Step 1).

   d. Replace *`account-b-bucket`* with the name of the S3 bucket in Account B.

   e. Replace *`name-of-user`* with the IAM user name that you use to log in to the console with Account A.

   If you're specifying an IAM role instead, update the format of the Amazon Resource Name (ARN):

```
"arn:aws:iam::account-a-id:role/name-of-role"
```

4. Choose **Save changes**.

Step 4: Create a DataSync destination location for the S3 bucket

After you create a location for the S3 bucket, you can run your DataSync task. The DataSync console, however, doesn’t support creating locations in different accounts. You must create the location with the AWS CLI before you can run the task.

To create a DataSync location with the CLI

1. Open a terminal.
2. Make sure that your CLI profile is configured to use Account A.
3. Copy the following command:

```
aws datasync create-location-s3 --s3-bucket-arn arn:aws:s3:::account-b-bucket --s3-
config '{"BucketAccessRoleArn":"arn:aws:iam::account-a-id:role/name-of-role"}'
```

4. Replace *account-b-bucket* with the name of the S3 bucket in Account B.
5. Replace *account-a-id* with the AWS account number of Account A.
6. Replace *name-of-role* with the IAM role that you created for DataSync in Account A (back in Step 1).
7. Run the command.

   If the command returns a DataSync location ARN similar to this, you successfully created the location:

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:123456789012:location/loc-
abcdef01234567890"
}
```

8. Switch back to Account A in the AWS Management Console.
9. Open the DataSync console at https://console.aws.amazon.com/datasync/.
10. In the left navigation pane, choose **Locations**.

    You can see the location of the S3 bucket in Account B that you just created with the CLI.

Step 5: Create and start a DataSync task

Before you move your data, let's recap what you've done so far:

- Deployed and activated your DataSync agent in Account A so that the agent can read from your self-managed storage system and communicate with AWS.
- Created an IAM role in Account A so that DataSync can write data to the S3 bucket in Account B.
- Configured your S3 bucket in Account B to ensure that your DataSync task works.
- Created your DataSync source and destination locations in Account A.

To create and start the DataSync task

1. While still using the DataSync console in Account A, choose **Tasks** in the left navigation pane then **Create task**.

   > **Note**
   > You must be logged in to the console with the same IAM user name (or role) for Account A that you specified in the S3 bucket policy in Step 3.
2. On the **Configure source location** page, choose **Choose an existing location**. Choose the source location that you're copying data from (your on-premises storage) then **Next**.
3. On the **Configure destination location** page, choose **Choose an existing location**. Choose the destination location that you're copying data to (the S3 bucket in Account B) then **Next**.
4. On the **Configure settings** page, give the task a name. As needed, configure additional settings, such as specifying an Amazon CloudWatch log group. Choose **Next**.
5. On the **Review** page, review your settings and choose **Create task**.
6. On the task's details page, choose **Start**, and then choose one of the following:
   - To run the task without modification, choose **Start with defaults**.
   - To modify the task before running it, choose **Start with overriding options**.

When your task finishes, you'll see the data from your on-premises storage in the S3 bucket. You can now access the bucket data from Account B.

Related resources

For more information about what you did in this tutorial, see the following topics:

- Creating a role for an AWS service (console)
- Modifying a role trust policy (console)
- Adding a bucket policy by using the Amazon S3 console
- Create an S3 location with the AWS CLI

Tutorial: Transferring data from Amazon S3 to Amazon S3 in a different AWS account

With AWS DataSync, you can move data between Amazon S3 buckets that belong to different AWS accounts.

> **Important**
> Copying data across AWS accounts using the methods in this tutorial works only with Amazon S3.

Overview

In this tutorial, you’ll learn how AWS Identity and Access Management (IAM) and the AWS Command Line Interface (AWS CLI) can help you create DataSync tasks that transfer data from Amazon S3 to another S3 bucket in a different AWS account.

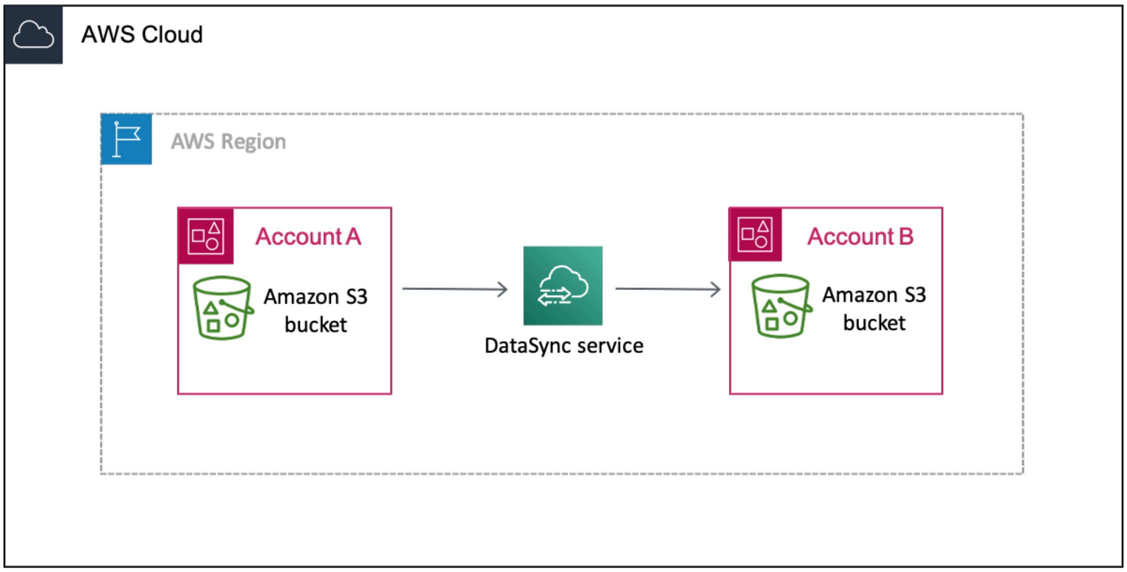
> **Tip**
> Follow this tutorial if your S3 buckets are also in different AWS Regions. The process is mostly the same except for some extra steps. Keep in mind, however, that DataSync doesn't support these kinds of transfers for Regions disabled by default.

Here's what this kind of scenario can look like:

- **Account A**: The AWS account that you use for managing the S3 bucket that you want to copy data from.
- **Account B**: The AWS account that you use for managing the S3 bucket that you want to copy data to.

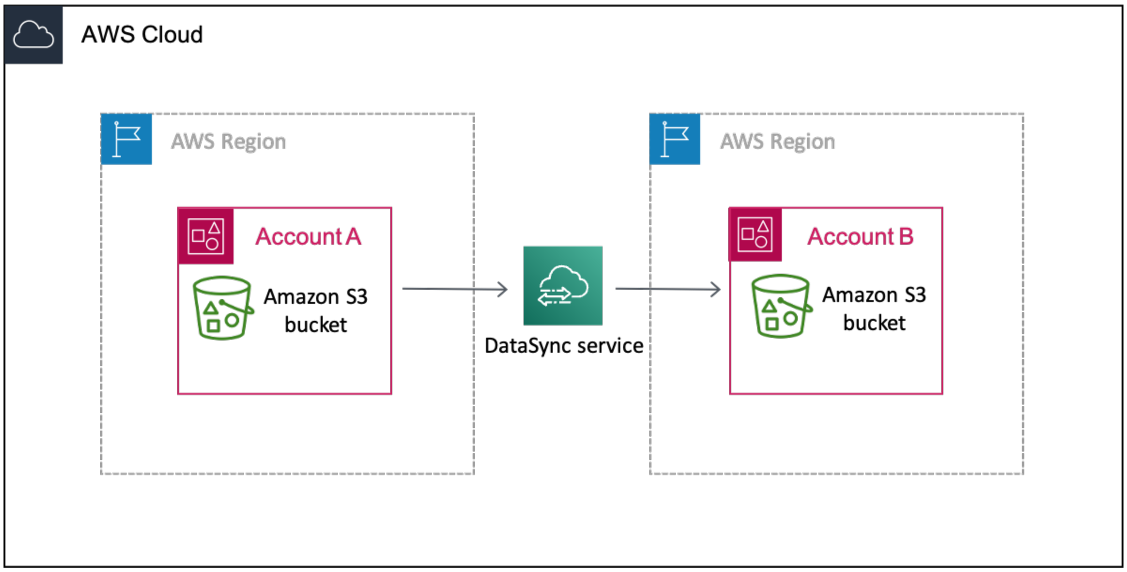
Transfers across accounts

The following diagram illustrates a scenario where you copy data from an S3 bucket to another S3 bucket that's in a different AWS account.



Transfers across accounts and Regions

The following diagram illustrates a scenario where you copy data from an S3 bucket to another S3 bucket that's in a different AWS account and Region.



Prerequisites

Before you begin the IAM work to facilitate the cross-account transfer, do the following if you already haven't:

1. Determine how many objects you're copying. Use Amazon S3 Storage Lens to figure out how many objects are in your bucket.

   Tip
   When transferring between S3 buckets, DataSync can't copy more than 25 million objects per task. If your bucket has more than 25 million objects, we recommend a couple options:
   - Organizing your objects using prefixes that you don't include more than 25 million objects. You can then create separate DataSync tasks for each prefix.
   - Filtering the data (p. 104) transferred by DataSync.
2. Create a DataSync source location (p. 88) with Account A for the S3 bucket that you're copying data from.
3. Set up the AWS CLI with Account A. You'll need the AWS CLI to create the DataSync destination location for the S3 bucket in Account B.

Step 1: Create an IAM role for DataSync in Account A

You need an IAM role that gives DataSync permission to write to the S3 bucket in Account B.

When you create a location for a bucket, DataSync can automatically create and assume a role with the right permissions to access that bucket. Since you're transferring across accounts, you must create the role manually.

For more information, see Creating a role for an AWS service (console) in the *IAM User Guide*.

Create the IAM role

Create a role with DataSync as the trusted entity.

To create the IAM role

1. Log in to the AWS Management Console with Account A.
2. Open the IAM console at https://console.aws.amazon.com/iam/.
3. In the left navigation pane, under **Access management**, choose **Roles**, and then choose **Create role**.
4. On the **Select trusted entity** page, for **Trusted entity type**, choose **AWS service**.
5. For **Use case**, choose **DataSync** in the dropdown list and select **DataSync**. Choose **Next**.
6. On the **Add permissions** page, choose **Next**.
7. Give your role a name and choose **Create role**.

Attach a custom policy to the IAM role

The IAM role needs a policy that allows DataSync to write to your S3 bucket in Account B.

To attach a custom policy to the IAM role

1. On the **Roles** page of the IAM console, search for the role that you just created and choose its name.
2. On the role's details page, choose the **Permissions** tab. Choose **Add permissions** then **Create inline policy**.
3. Choose the **JSON** tab and do the following:

a. Paste the following JSON into the policy editor:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::account-b-bucket"
    },
    {
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::account-b-bucket/*"
    }
  ]
}
```

b. Replace *`account-b-bucket`* with the name of the S3 bucket in Account B.

4. Choose **Review policy**.
5. Give your policy a name and choose **Create policy**.

Step 2: Disable ACLs for your S3 bucket in Account B

It's important that all the data that you copy to the S3 bucket belongs to Account B. To ensure that Account B is the owner of the data, disable the bucket's access control lists (ACLs). For more information, see Controlling ownership of objects and disabling ACLs for your bucket in the *Amazon S3 User Guide*.

To disable ACLs for an S3 bucket

1. In the AWS Management Console, switch over to Account B.
2. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. In the left navigation pane, choose **Buckets**.
4. In the **Buckets** list, choose the S3 bucket that you're transferring data to.
5. On the bucket's detail page, choose the **Permissions** tab.
6. Under **Object Ownership**, choose **Edit**.
7. If it isn't already selected, choose the **ACLs disabled (recommended)** option.
8. Choose **Save changes**.

Step 3: Update the S3 bucket policy in Account B

In Account B, modify the S3 bucket policy to allow access to the IAM role that you created for DataSync in Account A.

The updated policy (provided to you in the following instructions) includes two principals:

- The first principal specifies the IAM role that you created in Account A that allows DataSync to write to the S3 bucket.
- The second principal specifies the IAM user name for Account A, which allows you to create the DataSync destination for the S3 bucket by using the AWS CLI (you'll do this in Step 4).

  Note

  If you log in to the console and access the AWS CLI using an IAM role, specify that role instead of a user name for the second principal.

To update the S3 bucket policy

1. While still in the S3 console and using Account B, choose the S3 bucket that you're copying data to.
2. On the bucket's detail page, choose the **Permissions** tab.
3. Under **Bucket policy**, choose **Edit** and do the following to modify your S3 bucket policy:

   a. Update what's in the editor to include the following policy statements:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "DataSyncCreateS3LocationAndTaskAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-a-id:role/name-of-role"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::account-b-bucket",
        "arn:aws:s3:::account-b-bucket/*"
      ]
    },
    {
      "Sid": "DataSyncCreateS3Location",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-a-id:user/name-of-user"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::account-b-bucket"
    }
  ]
}
```

   b. Replace *account-a-id* with the AWS account number of Account A.

   c. Replace *name-of-role* with the IAM role that you created for DataSync in Account A (back in Step 1).

   d. Replace *account-b-bucket* with the name of the S3 bucket in Account B.

e. Replace *`name-of-user`* with the IAM user name that you use to log in to the console with Account A.

   If you're specifying an IAM role instead, update the format of the Amazon Resource Name (ARN):

   `"arn:aws:iam::`*`account-a-id`*`:role/`*`name-of-role`*`"`

4. Choose **Save changes**.

Step 4: Create a DataSync destination location for the S3 bucket

After you create a location for the S3 bucket, you can run your DataSync task. The DataSync console, however, doesn't support creating locations in different accounts. You must create the location with the AWS CLI before you can run the task.

To create a DataSync location with the CLI

1. Open a terminal.
2. Make sure that your CLI profile is configured to use Account A.
3. Copy the following command:

```
aws datasync create-location-s3 --s3-bucket-arn arn:aws:s3:::account-b-bucket --s3-
config '{"BucketAccessRoleArn":"arn:aws:iam::account-a-id:role/name-of-role"}'
```

4. Replace *`account-b-bucket`* with the name of the S3 bucket in Account B.
5. Replace *`account-a-id`* with the AWS account number of Account A.
6. Replace *`name-of-role`* with the IAM role that you created for DataSync in Account A (back in Step 1).
7. If the bucket in Account B is in a different Region than the bucket in Account A, add the `--region` option at the end of the command to specify the Region where there Account B bucket resides. For example, `--region` *`us-west-1`*.
8. Run the command.

   If the command returns a DataSync location ARN similar to this, you successfully created the location:

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:123456789012:location/loc-
abcdef01234567890"
}
```

9. Switch back to Account A in the AWS Management Console.
10. Open the DataSync console at https://console.aws.amazon.com/datasync/.
11. In the left navigation pane, choose **Locations**.

    You can see the location of the S3 bucket in Account B that you just created with the CLI.

Step 5: Create and start a DataSync task

Before you move your data, let's recap what you've done so far:

- Created an IAM role in Account A so that DataSync can write data to the S3 bucket in Account B.
- Configured your S3 bucket in Account B to ensure that your DataSync task works.
- Created your DataSync source and destination locations in Account A.

To create and start the DataSync task

1. While still using the DataSync console in Account A, choose **Tasks** in the left navigation pane then **Create task**.

   Note
   You must be logged in to the console with the same IAM user name (or role) for Account A that you specified in the S3 bucket policy in Step 3.
2. If the bucket in Account B is in a different Region than the bucket in Account A, choose the Account B bucket's Region in the navigation pane.

   You must start the DataSync task from the Region of the destination location (in this case, the Account B bucket) to avoid a connection error.
3. On the **Configure source location** page, choose **Choose an existing location**.
4. For transfers across Regions, choose the Region where the Account A bucket resides.
5. Choose the source location that you're copying data from (the S3 bucket in Account A) then **Next**.
6. On the **Configure destination location** page, choose **Choose an existing location**. Choose the destination location that you're copying data to (the S3 bucket in Account B) then **Next**.
7. On the **Configure settings** page, give the task a name. As needed, configure additional settings, such as specifying an Amazon CloudWatch log group. Choose **Next**.
8. On the **Review** page, review your settings and choose **Create task**.
9. On the task's details page, choose **Start**, and then choose one of the following:
   - To run the task without modification, choose **Start with defaults**.
   - To modify the task before running it, choose **Start with overriding options**.

When your task finishes, check the S3 bucket in Account B. You should see the data from your Account A bucket.

Related resources

For more information about what you did in this tutorial, see the following topics:

- Creating a role for an AWS service (console)
- Modifying a role trust policy (console)
- Adding a bucket policy by using the Amazon S3 console
- Create an S3 location with the AWS CLI

Tutorial: Transferring data from Google Cloud Storage to Amazon S3

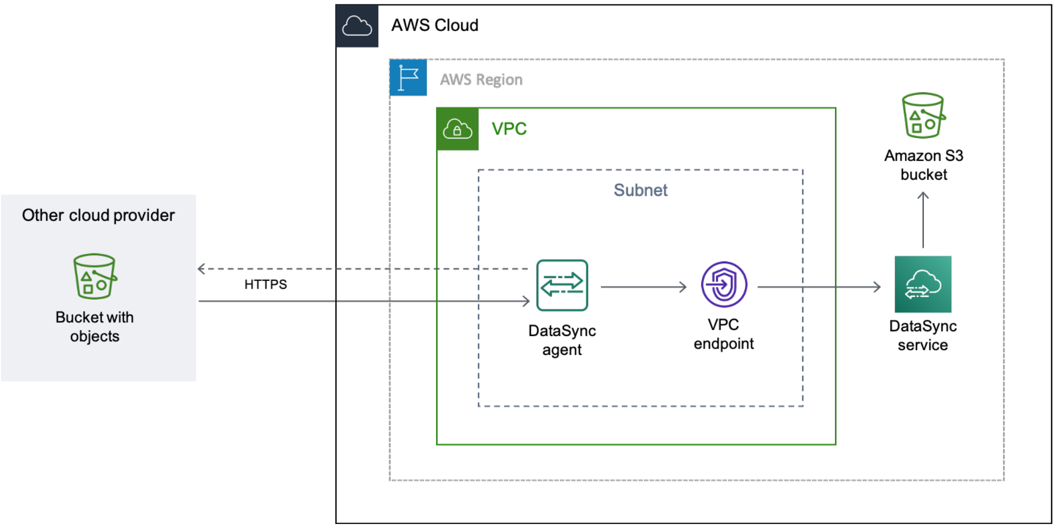
The following tutorial shows how you can use AWS DataSync to migrate objects from a Google Cloud Storage bucket to an Amazon S3 bucket.

Overview

Because DataSync integrates with the Google Cloud Storage XML API, you can copy objects into Amazon S3 without writing code. Let's get an idea how this transfer works:

1. You deploy a DataSync agent in a virtual private cloud (VPC) in your AWS environment.
2. The agent reads your Google Cloud Storage bucket using a Hash-based Message Authentication Code (HMAC) key.
3. The objects from your Google Cloud Storage bucket move securely through TLS 1.2 into the AWS Cloud by using a private VPC endpoint.
4. The DataSync service writes the data to your S3 bucket.

The following diagram illustrates the transfer.



Costs

The fees associated with this migration include:

- Running an Amazon EC2 instance (for your DataSync agent)
- Transferring the data using DataSync
- Transferring data out of Google Cloud Storage
- Storing data in Amazon S3

Prerequisites

Before you begin, do the following if you haven't already:

- Create a Google Cloud Storage bucket with the objects you want to transfer to AWS.
- Create an AWS account.
- Set up the AWS CLI.
- Create an Amazon S3 bucket where you want the objects to go once they're in AWS.

Step 1: Create an HMAC key for your Google Cloud Storage bucket

DataSync uses an HMAC key associated with your Google service account to authenticate with and read the bucket you're transferring data from. (If you need detailed instructions on how to do this, see the Google Cloud Storage documentation.)

To create an HMAC key

1. Create an HMAC key for your Google service account.
2. Make sure that your Google service account has at least `Storage Object Viewer` permissions.
3. Save your HMAC key's access ID and secret in a secure location.

   You need these later to configure your DataSync source location.

Step 2: Configure your network

You need a VPC in AWS to host your DataSync agent. The VPC must also have an interface endpoint (using AWS PrivateLink) to facilitate the transfer.

To configure your network

1. If you don't have one, create a VPC in the same AWS Region as your S3 bucket.
2. Create a private subnet for your VPC.
3. Create a VPC endpoint for DataSync.
4. Configure your network to allow a DataSync agent to transfer data through a VPC endpoint (p. 12).

   You can do this modifying the security group associated with your VPC endpoint.

Step 3: Create a DataSync agent

You need a DataSync agent to connect to your Google Cloud Storage bucket. In this scenario, the agent runs in a VPC associated with your AWS account.

Deploy your agent

In this tutorial, you use the AWS CLI and AWS Management Console to deploy your agent as an Amazon EC2 instance.

To deploy the DataSync agent

1. Open a terminal. Make sure to configure your AWS CLI profile to use the account with your S3 bucket.
2. Copy the following command. Replace *`vpc-region`* with the AWS Region where your VPC resides (for example, `us-east-1`).

```
aws ssm get-parameter --name /aws/service/datasync/ami --region vpc-region
```

3. Run the command. In the output, take note of the `"Value"` property.

   This value is the DataSync Amazon Machine Image (AMI) ID of the Region you specified (an AMI ID looks like `ami-1234567890abcdef0`).

4. Copy the following URL. Again, replace *vpc-region* with the AWS Region where your VPC resides. Then, replace *ami-id* with AMI ID you noted in the previous step.

   ```
   https://console.aws.amazon.com/ec2/v2/home?region=vpc-
   region#LaunchInstanceWizard:ami=ami-id
   ```
5. Paste the URL into a browser.

   This opens the Amazon EC2 instance launch page in the AWS Management Console.
6. For **Instance type**, choose one of the recommended Amazon EC2 instances for DataSync agents (p. 10).
7. For **Key pair**, choose an existing key pair or create a new one.
8. For **Network settings**, choose the VPC and subnet where you want to deploy the agent.
9. Choose **Launch instance**.

Specify the agent for your DataSync task

With your network set up and agent running, you can start configuring your transfer.

1. Choose your VPC endpoint (p. 26).
2. Activate your agent (p. 27).

Step 4: Create a DataSync source location for your Google Cloud Storage bucket

To set up a DataSync location for your Google Cloud Storage bucket, you need the access ID and secret for the HMAC key you created in Step 1.

To create the DataSync source location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the left navigation pane, choose **Locations**, then choose **Create location**.
3. For **Location type**, choose **Object storage**.
4. For **Agents**, choose the agent you created in Step 3.
5. For **Server**, enter **`storage.googleapis.com`**.
6. For **Bucket name**, enter the name of your Google Cloud Storage bucket.
7. Expand **Additional settings**. For **Server protocol**, choose **HTTPS**. For **Server port**, choose **443**.
8. Scroll down to the **Authentication** section. Make sure the **Requires credentials** check box is selected, and then do the following:

   - For **Access key**, enter your HMAC key’s access ID.
   - For **Secret key**, enter your HMAC key’s secret.
9. Choose **Create location**.

Step 5: Create a DataSync destination location for your S3 bucket

You need a DataSync location for where you want your data to end up.

To create the DataSync destination location

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the left navigation pane, choose **Locations**, then choose **Create location**.
3. Create a DataSync location for the S3 bucket (p. 88).

   You can configure the location's settings however you like, though this tutorial assumes the S3 bucket is in the same AWS Region as your VPC and DataSync agent.

Step 6: Create and start a DataSync task

With your source and destinations locations configured, you can start moving your data into AWS.

To create and start the DataSync task

1. Open the AWS DataSync console at https://console.aws.amazon.com/datasync/.
2. In the left navigation pane, choose **Tasks**, then choose **Create task**.
3. On the **Configure source location** page, do the following:

   a. Select **Choose an existing location**.

   b. Choose the source location you created in Step 4, then choose **Next**.
4. On the **Configure destination location** page, do the following:

   a. Select **Choose an existing location**.

   b. Choose the destination location you created in Step 5, then choose **Next**.
5. On the **Configure settings** page, do the following:

   a. Under **Data transfer configuration**, expand **Additional settings** and deselect the **Copy object tags** option.

      Important
      Your DataSync task may fail if you try to copy object tags. For more information, see Considerations when migrating to or from a Google Cloud Storage bucket (p. 80).

   b. Configure any other task settings you want and choose **Next**.
6. On the **Review** page, review your settings, and then choose **Create task**.
7. On the task's details page, choose **Start**, and then choose one of the following:

   - To run the task without modification, choose **Start with defaults**.
   - To modify the task before running it, choose **Start with overriding options**.

When your task finishes, you'll see the objects from your Google Cloud Storage bucket in your S3 bucket.

Additional resources for AWS DataSync

In this section, you can find additional information about and resources for AWS DataSync.

Topics

Transferring data from a self-managed storage array

You might want to transfer data from a self-managed enterprise storage array to Amazon EFS. In this case, files in the source file system might be modified by another application while the files are being transferred from Network File System (NFS) or Server Message Block (SMB) file share to Amazon EFS.

To ensure that DataSync successfully performs a transfer with full consistency verification, we recommend that the source location point to a read-only snapshot. This setup ensures that files at the source location can't be modified while the files are being transferred, and makes sure that verification works.

For information about how to take a snapshot in an enterprise storage array, see one of the following:

- EMC VNX: How to create a VNX snapshot and attach it to a server
- NetApp: Snapshot management
- HPE 3PAR: Creating virtual volume snapshots
- HDS: Hitachi Copy-on-Write Snapshot User Guide

Additional AWS DataSync use cases

In this section, you can find information about use cases in AWS DataSync that are not common to most users.

Topics

Transferring files in opposite directions

Transferring data in opposite directions allows for workflows where the active application moves between locations. AWS DataSync doesn't support workflows where multiple active applications write to both locations at the same time. Use the steps in the following procedure to configure DataSync to transfer data in opposite directions.

To configure DataSync to data transfers in opposite directions

1. Create a location and name it **Location A**.
2. Create a second location and name it **Location B**.
3. Create a task, name it **Task A-B**, and then configure **Location A** as the source location and **Location B** as the destination location.
4. Create a second task, name it **Task B-A**, and then configure **Location B** as the source location and **Location A** as the destination location.
5. To update **Location B** with data from **Location A**, run **Task A-B**.

   To update **Location A** with data from **Location B**, run **Task B-A**.

   Don't run these two tasks concurrently. DataSync can transfer files in opposite directions periodically. However, it doesn't support workflows where multiple active applications write to both **Location A** and **Location B** simultaneously.

Using multiple tasks to write to the same Amazon S3 bucket

In certain use cases, you might want different tasks to write to the same Amazon S3 bucket. In this case, you create different folders in the S3 bucket for each of the task. This approach prevents file name conflicts between the tasks, and also means that you can set different permissions for each of folders.

For example, you might have three tasks: `task1`, `task2`, and `task3` write to an S3 bucket named `MyBucket`.

You create three folders in the bucket:

`s3://MyBucket/task1`

`s3://MyBucket/task2`

`s3://MyBucket/task3`

For each task, you choose the folder in `MyBucket` that corresponds to the task as the destination, and set different permissions for each of the three folders.

Allowing DataSync to access a restricted Amazon S3 bucket

In some cases, you might want to limit access to your Amazon S3 bucket. You can edit the S3 bucket policy so that DataSync can still access the bucket when you run a task.

To allow DataSync to access a restricted S3 bucket

1. Copy the following sample policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:userid": [
            "datasync-role-id:*",
            "datasync-user-id"
          ]
        }
      }
    }
  ]
}
```

2. In the sample policy, replace these values:

   - *bucket-name*: The name of the S3 bucket that you're restricting access to.
   - *datasync-role-id*: The ID of the DataSync IAM role that needs access to the S3 bucket. Get the IAM role ID by running the following AWS CLI command:

     ```
     aws iam get-role --role-name datasync-iam-role-name
     ```

     In the output, look for the `RoleId` value:

     ```
     "RoleId": "ANPAJ2UCCR6DPCEXAMPLE"
     ```
   - *datasync-user-id*: The ID of the IAM user creating the DataSync location for the S3 bucket. (If you're using a role to create the location, specify that role ID instead.) Get the IAM user ID by running the following AWS CLI command:

     ```
     aws iam get-user
     ```

     In the output, look for the `UserId` value:

     ```
     "UserId": "AIDACKCEVSQ6C2EXAMPLE"
     ```
3. Add this policy to your S3 bucket policy. For more information, see how to edit a bucket policy in the *Amazon S3 User Guide*.

Once you've updated the S3 bucket policy, you must add additional IAM roles or users to the policy for those who need to access the S3 bucket.

DataSync API

In addition to using the console, you can use the AWS DataSync API to programmatically configure and manage DataSync and its resources. This section describes the AWS DataSync operations and data types and contains the API Reference documentation for AWS DataSync.

Topics

Actions

The following actions are supported:

- CancelTaskExecution (p. 168)
- CreateAgent (p. 170)
- CreateLocationEfs (p. 174)
- CreateLocationFsxLustre (p. 178)
- CreateLocationFsxOpenZfs (p. 181)
- CreateLocationFsxWindows (p. 184)
- CreateLocationHdfs (p. 187)
- CreateLocationNfs (p. 192)
- CreateLocationObjectStorage (p. 196)
- CreateLocationS3 (p. 200)
- CreateLocationSmb (p. 205)
- CreateTask (p. 209)
- DeleteAgent (p. 214)
- DeleteLocation (p. 216)
- DeleteTask (p. 218)
- DescribeAgent (p. 220)
- DescribeLocationEfs (p. 223)
- DescribeLocationFsxLustre (p. 227)
- DescribeLocationFsxOpenZfs (p. 230)
- DescribeLocationFsxWindows (p. 233)
- DescribeLocationHdfs (p. 236)
- DescribeLocationNfs (p. 240)
- DescribeLocationObjectStorage (p. 243)
- DescribeLocationS3 (p. 246)
- DescribeLocationSmb (p. 249)
- DescribeTask (p. 252)
- DescribeTaskExecution (p. 258)
- ListAgents (p. 263)

CancelTaskExecution

Cancels execution of a task.

When you cancel a task execution, the transfer of some files is abruptly interrupted. The contents of files that are transferred to the destination might be incomplete or inconsistent with the source files. However, if you start a new task execution on the same task and you allow the task execution to complete, file content on the destination is complete and consistent. This applies to other unexpected failures that interrupt a task execution. In all of these cases, AWS DataSync successfully complete the transfer when you start the next task execution.

Request Syntax

```
{
   "TaskExecutionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

TaskExecutionArn (p. 168)

The Amazon Resource Name (ARN) of the task execution to cancel.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateAgent

Activates an AWS DataSync agent that you have deployed on your host. The activation process associates your agent with your account. In the activation process, you specify information such as the AWS Region that you want to activate the agent in. You activate the agent in the AWS Region where your target locations (in Amazon S3 or Amazon EFS) reside. Your tasks are created in this AWS Region.

You can activate the agent in a VPC (virtual private cloud) or provide the agent access to a VPC endpoint so you can run tasks without going over the public internet.

You can use an agent for more than one location. If a task uses multiple agents, all of them need to have status AVAILABLE for the task to run. If you use multiple agents for a source location, the status of all the agents must be AVAILABLE for the task to run.

For more information, see Creating and activating an agent in the *AWS DataSync User Guide*.

Agents are automatically updated by AWS on a regular basis, using a mechanism that ensures minimal interruption to your tasks.

Request Syntax

```
{
   "ActivationKey": "string",
   "AgentName": "string",
   "SecurityGroupArns": [ "string" ],
   "SubnetArns": [ "string" ],
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "VpcEndpointId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

ActivationKey (p. 170)

Your agent activation key. You can get the activation key either by sending an HTTP GET request with redirects that enable you to get the agent IP address (port 80). Alternatively, you can get it from the DataSync console.

The redirect URL returned in the response provides you the activation key for your agent in the query string parameter `activationKey`. It might also include other activation-related parameters; however, these are merely defaults. The arguments you pass to this API call determine the actual configuration of your agent.

For more information, see Creating and activating an agent in the *AWS DataSync User Guide.*

Type: String

Length Constraints: Maximum length of 29.

Pattern: `[A-Z0-9]{5}(-[A-Z0-9]{5}){4}`

Required: Yes

AgentName (p. 170)

The name you configured for your agent. This value is a text reference that is used to identify the agent in the console.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

SecurityGroupArns (p. 170)

The ARNs of the security groups used to protect your data transfer task subnets. See SecurityGroupArns.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: No

SubnetArns (p. 170)

The Amazon Resource Names (ARNs) of the subnets in which DataSync will create elastic network interfaces for each data transfer task. The agent that runs a task must be private. When you start a task that is associated with an agent created in a VPC, or one that has access to an IP address in a VPC, then the task is also private. In this case, DataSync creates four network interfaces for each task in your subnet. For a data transfer to work, the agent must be able to route to all these four network interfaces.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:subnet/.*$`

Required: No

Tags (p. 170)

The key-value pair that represents the tag that you want to associate with the agent. The value can be an empty string. This value helps you manage, filter, and search for your agents.

> **Note**
> Valid characters for key and value are letters, spaces, and numbers representable in UTF-8 format, and the following special characters: + - = . _ : / @.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

VpcEndpointId (p. 170)

The ID of the VPC (virtual private cloud) endpoint that the agent has access to. This is the client-side VPC endpoint, also called a PrivateLink. If you don't have a PrivateLink VPC endpoint, see Creating a VPC Endpoint Service Configuration in the Amazon VPC User Guide.

VPC endpoint ID looks like this: `vpce-01234d5aff67890e1`.

Type: String

Pattern: `^vpce-[0-9a-f]{17}$`

Required: No

Response Syntax

```
{
   "AgentArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AgentArn (p. 172)

The Amazon Resource Name (ARN) of the agent. Use the `ListAgents` operation to return a list of agents for your account and AWS Region.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example creates an agent and activates it using an activation key.

Sample Request

```
{
  "ActivationKey": "AAAAA-7AAAA-GG7MC-3I9R3-27COD",
  "AgentName": "MyAgent",
  "Tags": [
  {
    "Key": "Job",
    "Value": "TransferJob-1"
 }
  ]
}
```

Example

The response returns the Amazon Resource Name (ARN) of the activated agent.

Sample Response

```
{
  "AgentArn": "arn:aws:datasync:us-east-2:111222333444:agent/agent-0b0addbeef44baca3"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationEfs

Creates an endpoint for an Amazon EFS file system that AWS DataSync can access for a transfer. For more information, see Creating a location for Amazon EFS.

Request Syntax

```
{
   "AccessPointArn": "string",
   "Ec2Config": {
      "SecurityGroupArns": [ "string" ],
      "SubnetArn": "string"
   },
   "EfsFilesystemArn": "string",
   "FileSystemAccessRoleArn": "string",
   "InTransitEncryption": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AccessPointArn (p. 174)

Specifies the Amazon Resource Name (ARN) of the access point that DataSync uses to access the Amazon EFS file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):elasticfilesystem:[a-z\-0-9]+:[0-9]{12}:access-point/fsap-[0-9a-f]{8,40}$`

Required: No

Ec2Config (p. 174)

Specifies the subnet and security groups DataSync uses to access your Amazon EFS file system.

Type: Ec2Config (p. 307) object

Required: Yes

EfsFilesystemArn (p. 174)

Specifies the ARN for the Amazon EFS file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):elasticfilesystem:[a-z\-0-9]*:[0-9]{12}:file-system/fs-.*$`

Required: Yes

FileSystemAccessRoleArn (p. 174)

Specifies an AWS Identity and Access Management (IAM) role that DataSync assumes when mounting the Amazon EFS file system.

Type: String

Length Constraints: Maximum length of 2048.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`

Required: No

InTransitEncryption (p. 174)

Specifies whether you want DataSync to use Transport Layer Security (TLS) 1.2 encryption when it copies data to or from the Amazon EFS file system.

If you specify an access point using `AccessPointArn` or an IAM role using `FileSystemAccessRoleArn`, you must set this parameter to `TLS1_2`.

Type: String

Valid Values: `NONE | TLS1_2`

Required: No

Subdirectory (p. 174)

Specifies a mount path for your Amazon EFS file system. This is where DataSync reads or writes data (depending on if this is a source or destination location). By default, DataSync uses the root directory, but you can also include subdirectories.

> **Note**
> You must specify a value with forward slashes (for example, `/path/to/folder`).

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]*$`

Required: No

Tags (p. 174)

Specifies the key-value pair that represents a tag that you want to add to the resource. The value can be an empty string. This value helps you manage, filter, and search for your resources. We recommend that you create a name tag for your location.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 176)

The Amazon Resource Name (ARN) of the Amazon EFS file system location that you create.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Sample Request

The following example creates a location for an Amazon EFS file system.

```
{
    "Ec2Config": {
        "SubnetArn": "arn:aws:ec2:us-east-2:11122233344:subnet/subnet-1234567890abcdef1",
        "SecurityGroupArns": [
            "arn:aws:ec2:us-east-2:11122233344:security-group/sg-1234567890abcdef2"
        ]
    },
    "EfsFilesystemArn": "arn:aws:elasticfilesystem:us-east-2:111222333444:file-system/
fs-021345abcdef6789",
    "Subdirectory": "/mount/path",
    "Tags": [{
        "Key": "Name",
        "Value": "ElasticFileSystem-1"
    }]
}
```

Sample Request: Creating a location for a restricted Amazon EFS file system

The following example creates a location for an Amazon EFS file system with restricted access. In this kind of scenario, you might have to specify values for `AccessPointArn`, `FileSystemAccessRoleArn`, and `InTransitEncryption` in your request.

```
{
    "AccessPointArn": "arn:aws:elasticfilesystem:us-east-2:111222333444:access-point/
fsap-1234567890abcdef0",
    "Ec2Config": {
        "SubnetArn": "arn:aws:ec2:us-east-2:111222333444:subnet/subnet-1234567890abcdef1",
        "SecurityGroupArns": [
            "arn:aws:ec2:us-east-2:111222333444:security-group/sg-1234567890abcdef2"
        ]
    },
    "FileSystemAccessRoleArn": "arn:aws:iam::111222333444:role/AwsDataSyncFullAccessNew",
    "InTransitEncryption": "TLS1_2",
    "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-
abcdef01234567890",
    "LocationUri": "efs://us-east-2.fs-021345abcdef6789/",
    "Subdirectory": "/mount/path",
    "Tags": [{
        "Key": "Name",
        "Value": "ElasticFileSystem-1"
    }]
}
```

Sample Response

A response returns the location ARN of the Amazon EFS file system.

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-12abcdef012345678"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationFsxLustre

Creates an endpoint for an Amazon FSx for Lustre file system.

Request Syntax

```
{
   "FsxFilesystemArn": "string",
   "SecurityGroupArns": [ "string" ],
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

FsxFilesystemArn (p. 178)

The Amazon Resource Name (ARN) for the FSx for Lustre file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):fsx:[a-z\-0-9]*:[0-9]{12}:file-system/fs-.*$`

Required: Yes

SecurityGroupArns (p. 178)

The Amazon Resource Names (ARNs) of the security groups that are used to configure the FSx for Lustre file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: Yes

Subdirectory (p. 178)

A subdirectory in the location's path. This subdirectory in the FSx for Lustre file system is used to read data from the FSx for Lustre source location or write data to the FSx for Lustre destination.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: No

Tags (p. 178)

The key-value pair that represents a tag that you want to add to the resource. The value can be an empty string. This value helps you manage, filter, and search for your resources. We recommend that you create a name tag for your location.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 179)

The Amazon Resource Name (ARN) of the FSx for Lustre file system location that's created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationFsxOpenZfs

Creates an endpoint for an Amazon FSx for OpenZFS file system.

Request Syntax

```
{
   "FsxFilesystemArn": "string",
   "Protocol": {
      "NFS": {
         "MountOptions": {
            "Version": "string"
         }
      }
   },
   "SecurityGroupArns": [ "string" ],
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

FsxFilesystemArn (p. 181)

The Amazon Resource Name (ARN) of the FSx for OpenZFS file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):fsx:[a-z\-0-9]*:[0-9]{12}:file-system/fs-.*$`

Required: Yes

Protocol (p. 181)

The type of protocol that AWS DataSync uses to access your file system.

Type: FsxProtocol (p. 309) object

Required: Yes

SecurityGroupArns (p. 181)

The ARNs of the security groups that are used to configure the FSx for OpenZFS file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: Yes

Subdirectory (p. 181)

A subdirectory in the location's path that must begin with `/fsx`. DataSync uses this subdirectory to read or write data (depending on whether the file system is a source or destination location).

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

Required: No

Tags (p. 181)

The key-value pair that represents a tag that you want to add to the resource. The value can be an empty string. This value helps you manage, filter, and search for your resources. We recommend that you create a name tag for your location.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 182)

The ARN of the FSx for OpenZFS file system location that you created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationFsxWindows

Creates an endpoint for an Amazon FSx for Windows File Server file system.

Request Syntax

```
{
   "Domain": "string",
   "FsxFilesystemArn": "string",
   "Password": "string",
   "SecurityGroupArns": [ "string" ],
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "User": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Domain (p. 184)

The name of the Windows domain that the FSx for Windows File Server belongs to.

Type: String

Length Constraints: Maximum length of 253.

Pattern: `^([A-Za-z0-9]+[A-Za-z0-9-.]*)*[A-Za-z0-9-]*[A-Za-z0-9]$`

Required: No

FsxFilesystemArn (p. 184)

The Amazon Resource Name (ARN) for the FSx for Windows File Server file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):fsx:[a-z\-0-9]*:[0-9]{12}:file-system/fs-.*$`

Required: Yes

Password (p. 184)

The password of the user who has the permissions to access files and folders in the FSx for Windows File Server file system.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^.{0,104}$`

Required: Yes

SecurityGroupArns (p. 184)

The ARNs of the security groups that are used to configure the FSx for Windows File Server file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: Yes

Subdirectory (p. 184)

A subdirectory in the location's path. This subdirectory in the Amazon FSx for Windows File Server file system is used to read data from the Amazon FSx for Windows File Server source location or write data to the FSx for Windows File Server destination.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: No

Tags (p. 184)

The key-value pair that represents a tag that you want to add to the resource. The value can be an empty string. This value helps you manage, filter, and search for your resources. We recommend that you create a name tag for your location.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

User (p. 184)

The user who has the permissions to access files and folders in the FSx for Windows File Server file system.

For information about choosing a user name that ensures sufficient permissions to files, folders, and metadata, see user.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^[^\x5B\x5D\\/:;|=,+*?]{1,104}$`

Required: Yes

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 186)

The Amazon Resource Name (ARN) of the FSx for Windows File Server file system location you created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationHdfs

Creates an endpoint for a Hadoop Distributed File System (HDFS).

Request Syntax

```
{
   "AgentArns": [ "string" ],
   "AuthenticationType": "string",
   "BlockSize": number,
   "KerberosKeytab": blob,
   "KerberosKrb5Conf": blob,
   "KerberosPrincipal": "string",
   "KmsKeyProviderUri": "string",
   "NameNodes": [
      {
         "Hostname": "string",
         "Port": number
      }
   ],
   "QopConfiguration": {
      "DataTransferProtection": "string",
      "RpcProtection": "string"
   },
   "ReplicationFactor": number,
   "SimpleUser": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArns (p. 187)

The Amazon Resource Names (ARNs) of the agents that are used to connect to the HDFS cluster.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

AuthenticationType (p. 187)

The type of authentication used to determine the identity of the user.

Type: String

Valid Values: `SIMPLE | KERBEROS`

Required: Yes

BlockSize (p. 187)

The size of data blocks to write into the HDFS cluster. The block size must be a multiple of 512 bytes. The default block size is 128 mebibytes (MiB).

Type: Integer

Valid Range: Minimum value of 1048576. Maximum value of 1073741824.

Required: No

KerberosKeytab (p. 187)

The Kerberos key table (keytab) that contains mappings between the defined Kerberos principal and the encrypted keys. You can load the keytab from a file by providing the file's address. If you're using the AWS CLI, it performs base64 encoding for you. Otherwise, provide the base64-encoded text.

Note
If `KERBEROS` is specified for `AuthenticationType`, this parameter is required.

Type: Base64-encoded binary data object

Length Constraints: Maximum length of 65536.

Required: No

KerberosKrb5Conf (p. 187)

The `krb5.conf` file that contains the Kerberos configuration information. You can load the `krb5.conf` file by providing the file's address. If you're using the AWS CLI, it performs the base64 encoding for you. Otherwise, provide the base64-encoded text.

Note
If `KERBEROS` is specified for `AuthenticationType`, this parameter is required.

Type: Base64-encoded binary data object

Length Constraints: Maximum length of 131072.

Required: No

KerberosPrincipal (p. 187)

The Kerberos principal with access to the files and folders on the HDFS cluster.

Note
If `KERBEROS` is specified for `AuthenticationType`, this parameter is required.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^.+$`

Required: No

KmsKeyProviderUri (p. 187)

The URI of the HDFS cluster's Key Management Server (KMS).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^kms:\/\/http[s]?@(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])(;(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9]))*:[0-9]{1,5}\/kms$`

Required: No

NameNodes (p. 187)

The NameNode that manages the HDFS namespace. The NameNode performs operations such as opening, closing, and renaming files and directories. The NameNode contains the information to map blocks of data to the DataNodes. You can use only one NameNode.

Type: Array of HdfsNameNode (p. 311) objects

Array Members: Minimum number of 1 item.

Required: Yes

QopConfiguration (p. 187)

The Quality of Protection (QOP) configuration specifies the Remote Procedure Call (RPC) and data transfer protection settings configured on the Hadoop Distributed File System (HDFS) cluster. If `QopConfiguration` isn't specified, `RpcProtection` and `DataTransferProtection` default to `PRIVACY`. If you set `RpcProtection` or `DataTransferProtection`, the other parameter assumes the same value.

Type: QopConfiguration (p. 325) object

Required: No

ReplicationFactor (p. 187)

The number of DataNodes to replicate the data to when writing to the HDFS cluster. By default, data is replicated to three DataNodes.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 512.

Required: No

SimpleUser (p. 187)

The user name used to identify the client on the host operating system.

> **Note**
> If `SIMPLE` is specified for `AuthenticationType`, this parameter is required.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[_.A-Za-z0-9][-_.A-Za-z0-9]*$`

Required: No

Subdirectory (p. 187)

A subdirectory in the HDFS cluster. This subdirectory is used to read data from or write data to the HDFS cluster. If the subdirectory isn't specified, it will default to `/`.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: No

Tags (p. 187)

The key-value pair that represents the tag that you want to add to the location. The value can be an empty string. We recommend using tags to name your resources.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 190)

The ARN of the source HDFS cluster location that's created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationNfs

Defines a file system on a Network File System (NFS) server that can be read from or written to.

Request Syntax

```
{
   "MountOptions": {
      "Version": "string"
   },
   "OnPremConfig": {
      "AgentArns": [ "string" ]
   },
   "ServerHostname": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

MountOptions (p. 192)

The NFS mount options that DataSync can use to mount your NFS share.

Type: NfsMountOptions (p. 315) object

Required: No

OnPremConfig (p. 192)

Contains a list of Amazon Resource Names (ARNs) of agents that are used to connect to an NFS server.

If you are copying data to or from your AWS Snowcone device, see NFS Server on AWS Snowcone for more information.

Type: OnPremConfig (p. 316) object

Required: Yes

ServerHostname (p. 192)

The name of the NFS server. This value is the IP address or Domain Name Service (DNS) name of the NFS server. An agent that is installed on-premises uses this hostname to mount the NFS server in a network.

If you are copying data to or from your AWS Snowcone device, see NFS Server on AWS Snowcone for more information.

Note
This name must either be DNS-compliant or must be an IP version 4 (IPv4) address.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])$`

Required: Yes

Subdirectory (p. 192)

The subdirectory in the NFS file system that is used to read data from the NFS source location or write data to the NFS destination. The NFS path should be a path that's exported by the NFS server, or a subdirectory of that path. The path should be such that it can be mounted by other NFS clients in your network.

To see all the paths exported by your NFS server, run "`showmount -e nfs-server-name`" from an NFS client that has access to your server. You can specify any directory that appears in the results, and any subdirectory of that directory. Ensure that the NFS export is accessible without Kerberos authentication.

To transfer all the data in the folder you specified, DataSync needs to have permissions to read all the data. To ensure this, either configure the NFS export with `no_root_squash,` or ensure that the permissions for all of the files that you want DataSync allow read access for all users. Doing either enables the agent to read the files. For the agent to access directories, you must additionally enable all execute access.

If you are copying data to or from your AWS Snowcone device, see NFS Server on AWS Snowcone for more information.

For information about NFS export configuration, see 18.7. The /etc/exports Configuration File in the Red Hat Enterprise Linux documentation.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]+$`

Required: Yes

Tags (p. 192)

The key-value pair that represents the tag that you want to add to the location. The value can be an empty string. We recommend using tags to name your resources.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 193)

The Amazon Resource Name (ARN) of the source NFS file system location that is created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example creates an endpoint for an NFS file system using the specified NFS version as a mount option.

Sample Request

```
{
  "MountOptions": {
     "Version": : "NFS4_0"
     },
  "OnPremConfig": {
    "AgentArn": [ "arn:aws:datasync:us-east-2:111222333444:agent/agent-0b0addbeef44b3nfs" ]
          },

           "ServerHostname": "MyServer@amazon.com",
           "Subdirectory": "/MyFolder",
           "Tags": [
              {
                "Key": "Name",
                "Value": "ElasticFileSystem-1"
              }
           ]
}
```

Example

The response returns the Amazon Resource Name (ARN) of the NFS location.

Sample Response

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50aa"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationObjectStorage

Creates an endpoint for an object storage system that AWS DataSync can access for a transfer. For more information, see Creating a location for object storage.

Request Syntax

```
{
   "AccessKey": "string",
   "AgentArns": [ "string" ],
   "BucketName": "string",
   "SecretKey": "string",
   "ServerHostname": "string",
   "ServerPort": number,
   "ServerProtocol": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AccessKey (p. 196)

Specifies the access key (for example, a user name) if credentials are required to authenticate with the object storage server.

Type: String

Length Constraints: Minimum length of 8. Maximum length of 200.

Pattern: `^.+$`

Required: No

AgentArns (p. 196)

Specifies the Amazon Resource Names (ARNs) of the DataSync agents that can securely connect with your location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

BucketName (p. 196)

Specifies the name of the object storage bucket involved in the transfer.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: Yes

SecretKey (p. 196)

Specifies the secret key (for example, a password) if credentials are required to authenticate with the object storage server.

Type: String

Length Constraints: Minimum length of 8. Maximum length of 200.

Pattern: `^.+$`

Required: No

ServerHostname (p. 196)

Specifies the domain name or IP address of the object storage server. A DataSync agent uses this hostname to mount the object storage server in a network.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])$`

Required: Yes

ServerPort (p. 196)

Specifies the port that your object storage server accepts inbound network traffic on (for example, port 443).

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65536.

Required: No

ServerProtocol (p. 196)

Specifies the protocol that your object storage server uses to communicate.

Type: String

Valid Values: `HTTPS | HTTP`

Required: No

Subdirectory (p. 196)

Specifies the object prefix for your object storage server. If this is a source location, DataSync only copies objects with this prefix. If this is a destination location, DataSync writes all objects with this prefix.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]*$`

Required: No

Tags (p. 196)

Specifies the key-value pair that represents a tag that you want to add to the resource. Tags can help you manage, filter, and search for your resources. We recommend creating a name tag for your location.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 198)

Specifies the ARN of the object storage system location that you create.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationS3

Creates an endpoint for an Amazon S3 bucket.

For more information, see Create an Amazon S3 location in the *AWS DataSync User Guide*.

Request Syntax

```
{
   "AgentArns": [ "string" ],
   "S3BucketArn": "string",
   "S3Config": {
      "BucketAccessRoleArn": "string"
   },
   "S3StorageClass": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArns (p. 200)

If you're using DataSync on an AWS Outpost, specify the Amazon Resource Names (ARNs) of the DataSync agents deployed on your Outpost. For more information about launching a DataSync agent on an AWS Outpost, see Deploy your DataSync agent on AWS Outposts.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: No

S3BucketArn (p. 200)

The ARN of the Amazon S3 bucket. If the bucket is on an AWS Outpost, this must be an access point ARN.

Type: String

Length Constraints: Maximum length of 156.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):(s3|s3-outposts):[a-z\-0-9]*:[0-9]*:.*$`

Required: Yes

S3Config (p. 200)

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role used to access an Amazon S3 bucket.

For detailed information about using such a role, see Creating a Location for Amazon S3 in the *AWS DataSync User Guide*.

Type: S3Config (p. 326) object

Required: Yes

S3StorageClass (p. 200)

The Amazon S3 storage class that you want to store your files in when this location is used as a task destination. For buckets in AWS Regions, the storage class defaults to Standard. For buckets on AWS Outposts, the storage class defaults to AWS S3 Outposts.

For more information about S3 storage classes, see Amazon S3 Storage Classes. Some storage classes have behaviors that can affect your S3 storage cost. For detailed information, see Considerations when working with S3 storage classes in DataSync.

Type: String

Valid Values: `STANDARD | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE | OUTPOSTS`

Required: No

Subdirectory (p. 200)

A subdirectory in the Amazon S3 bucket. This subdirectory in Amazon S3 is used to read data from the S3 source location or write data to the S3 destination.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]*$`

Required: No

Tags (p. 200)

The key-value pair that represents the tag that you want to add to the location. The value can be an empty string. We recommend using tags to name your resources.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 201)

The Amazon Resource Name (ARN) of the source Amazon S3 bucket location that is created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Step 1. Allow to assume the IAM role required to write to the bucket

The following example shows the simplest policy that grants the required permissions for AWS DataSync to access a destination Amazon S3 bucket, followed by an IAM role to which the `create-location-s3-iam-role` policy has been attached.

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

```
"Role": {
        "Path": "/",
        "RoleName": "MyBucketAccessRole",
        "RoleId": "role-id",
        "Arn": "arn:aws:iam::account-id:role/MyBucketAccessRole",
        "CreateDate": "2018-07-27T02:49:23.117Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "datasync.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Step 2. Allow the created IAM role to write to the bucket

Attach a policy that has sufficient permissions to access the bucket to the role. An example of such policy is the `AWSDataSyncFullAccess` managed policy.

For more information, see AWSDataSyncFullAccess in the IAM console.

You don't need to create this policy. It's managed by AWS, so all that you need to do is specify its ARN in the `attach-role-policy` command.

```
IAM_POLICY_ARN='arn:aws:iam::aws:policy/AWSDataSyncFullAccess'
```

Step 3. Create an endpoint for an Amazon S3 bucket

The following example creates an endpoint for an Amazon S3 bucket.

When the S3 endpoint is created, a response similar to the second example following returns the Amazon Resource Name (ARN) for the new Amazon S3 location.

Sample Request

```
{
  "S3BucketArn": "arn:aws:s3:::MyBucket",
  "S3Config": {
     "BucketAccessRoleArn": "arn:aws:iam::111222333444:role/MyBucketAccessRole",
    },
    "S3StorageClass": "STANDARD",
    "Subdirectory": "/MyFolder",
    "Tags": [
       {
          "Key": "Name",
          "Value": "s3Bucket-1"
       }
     ]
}
```

Sample Response

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50s3"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateLocationSmb

Defines a file system on a Server Message Block (SMB) server that can be read from or written to.

Request Syntax

```
{
   "AgentArns": [ "string" ],
   "Domain": "string",
   "MountOptions": {
      "Version": "string"
   },
   "Password": "string",
   "ServerHostname": "string",
   "Subdirectory": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "User": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArns (p. 205)

The Amazon Resource Names (ARNs) of agents to use for a Simple Message Block (SMB) location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

Domain (p. 205)

The name of the Windows domain that the SMB server belongs to.

Type: String

Length Constraints: Maximum length of 253.

Pattern: `^([A-Za-z0-9]+[A-Za-z0-9-.]*)*[A-Za-z0-9-]*[A-Za-z0-9]$`

Required: No

MountOptions (p. 205)

The mount options used by DataSync to access the SMB server.

Type: SmbMountOptions (p. 327) object

Required: No

Password (p. 205)

The password of the user who can mount the share, has the permissions to access files and folders in the SMB share.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^.{0,104}$`

Required: Yes

ServerHostname (p. 205)

The name of the SMB server. This value is the IP address or Domain Name Service (DNS) name of the SMB server. An agent that is installed on-premises uses this hostname to mount the SMB server in a network.

> **Note**
> This name must either be DNS-compliant or must be an IP version 4 (IPv4) address.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])$`

Required: Yes

Subdirectory (p. 205)

The subdirectory in the SMB file system that is used to read data from the SMB source location or write data to the SMB destination. The SMB path should be a path that's exported by the SMB server, or a subdirectory of that path. The path should be such that it can be mounted by other SMB clients in your network.

> **Note**
> `Subdirectory` must be specified with forward slashes. For example, `/path/to/folder`.

To transfer all the data in the folder you specified, DataSync needs to have permissions to mount the SMB share, as well as to access all the data in that share. To ensure this, either ensure that the user/password specified belongs to the user who can mount the share, and who has the appropriate permissions for all of the files and directories that you want DataSync to access, or use credentials of a member of the Backup Operators group to mount the share. Doing either enables the agent to access the data. For the agent to access directories, you must additionally enable all execute access.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: Yes

Tags (p. 205)

The key-value pair that represents the tag that you want to add to the location. The value can be an empty string. We recommend using tags to name your resources.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

User (p. 205)

The user who can mount the share, has the permissions to access files and folders in the SMB share.

For information about choosing a user name that ensures sufficient permissions to files, folders, and metadata, see the User setting for SMB locations.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^[^\x5B\x5D\\/:;|=,+*?]{1,104}$`

Required: Yes

Response Syntax

```
{
   "LocationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LocationArn (p. 207)

The Amazon Resource Name (ARN) of the source SMB file system location that is created.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

This example illustrates one usage of CreateLocationSmb.

Sample Request

```
{
   "AgentArns":[
      "arn:aws:datasync:us-east-2:111222333444:agent/agent-0b0addbeef44b3nfs",
      "arn:aws:datasync:us-east-2:111222333444:agent/agent-2345noo35nnee1123ovo3"
   ],
   "Domain":"AMAZON",
   "MountOptions":{
      "Version":"SMB3"
   },
   "Password":"string",
   "ServerHostname":"MyServer.amazon.com",
   "Subdirectory":"share",
   "Tags":[
      {
         "Key":"department",
         "Value":"finance"
      }
   ],
   "User":"user-1"
}
```

Example

This example illustrates one usage of CreateLocationSmb.

Sample Response

```
{"arn:aws:datasync:us-east-1:111222333444:location/loc-0f01451b140b2af49"}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateTask

Creates a task.

A task includes a source location and a destination location, and a configuration that specifies how data is transferred. A task always transfers data from the source location to the destination location. The configuration specifies options such as task scheduling, bandwidth limits, etc. A task is the complete definition of a data transfer.

When you create a task that transfers data between AWS services in different AWS Regions, one of the two locations that you specify must reside in the Region where DataSync is being used. The other location must be specified in a different Region.

You can transfer data between commercial AWS Regions except for China, or between AWS GovCloud (US) Regions.

> **Important**
> When you use DataSync to copy files or objects between AWS Regions, you pay for data transfer between Regions. This is billed as data transfer OUT from your source Region to your destination Region. For more information, see Data Transfer pricing.

Request Syntax

```
{
   "CloudWatchLogGroupArn": "string",
   "DestinationLocationArn": "string",
   "Excludes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Includes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Name": "string",
   "Options": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "Schedule": {
      "ScheduleExpression": "string"
   },
   "SourceLocationArn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

CloudWatchLogGroupArn (p. 209)

The Amazon Resource Name (ARN) of the Amazon CloudWatch log group that is used to monitor and log events in the task.

For more information about how to use CloudWatch Logs with DataSync, see Monitoring Your Task in the *AWS DataSync User Guide.*

For more information about these groups, see Working with Log Groups and Log Streams in the *Amazon CloudWatch Logs User Guide*.

Type: String

Length Constraints: Maximum length of 562.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):logs:[a-z\-0-9]*:[0-9]{12}:log-group:([^:\*]*)(:\*)?$`

Required: No

DestinationLocationArn (p. 209)

The Amazon Resource Name (ARN) of an AWS storage resource's location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Excludes (p. 209)

A list of filter rules that determines which files to exclude from a task. The list should contain a single filter string that consists of the patterns to exclude. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

Includes (p. 209)

A list of filter rules that determines which files to include when running a task. The pattern contains a single filter string that consists of the patterns to include. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

Name (p. 209)

The name of a task. This value is a text reference that is used to identify the task in the console.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

Options (p. 209)

The set of configuration options that control the behavior of a single execution of the task that occurs when you call `StartTaskExecution`. You can configure these options to preserve metadata such as user ID (UID) and group ID (GID), file permissions, data integrity verification, and so on.

For each individual task execution, you can override these options by specifying the `OverrideOptions` before starting the task execution. For more information, see the StartTaskExecution operation.

Type: Options (p. 317) object

Required: No

Schedule (p. 209)

Specifies a schedule used to periodically transfer files from a source to a destination location. The schedule should be specified in UTC time. For more information, see Scheduling your task.

Type: TaskSchedule (p. 334) object

Required: No

SourceLocationArn (p. 209)

The Amazon Resource Name (ARN) of the source location for the task.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Tags (p. 209)

The key-value pair that represents the tag that you want to add to the resource. The value can be an empty string.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "TaskArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TaskArn (p. 212)

The Amazon Resource Name (ARN) of the task.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example creates a task using a source and destination locations.

Sample Request

```
{
  "Options": {
      "Atime": "BEST_EFFORT",
      "Gid": "NONE",
      "Mtime": "PRESERVE",
      "PosixPermissions": "PRESERVE",
      "PreserveDevices": "NONE",
      "PreserveDeletedFiles": "PRESERVE",
      "Uid": "NONE",
      "VerifyMode": "POINT_IN_TIME_CONSISTENT",
   },
   "Schedule": {
      "ScheduleExpression": "0 12 ? * SUN,WED *"
   },
    "CloudWatchLogGroupArn": "arn:aws:logs:us-east-2:111222333444:log-group",
   "DestinationLocationArn": "arn:aws:datasync:us-east-2:111222333444:location/
loc-07db7abfc326c50fb",
   "Name": "MyTask",
   "SourceLocationArn": "arn:aws:datasync:us-east-2:111222333444:location/
loc-0f01451b140b2af49",
   "Tags": [
       {
          "Key": "Name",
          "Value": "Task-1"
       }
    ]
}
```

Example

The following response returns the Amazon Resource Name (ARN) of the task.

Sample Response

```
{
  "TaskArn": "arn:aws:datasync:us-east-2:111222333444:task/task-08de6e6697796f026"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteAgent

Deletes an agent. To specify which agent to delete, use the Amazon Resource Name (ARN) of the agent in your request. The operation disassociates the agent from your AWS account. However, it doesn't delete the agent virtual machine (VM) from your on-premises environment.

Request Syntax

```
{
   "AgentArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArn (p. 214)

The Amazon Resource Name (ARN) of the agent to delete. Use the `ListAgents` operation to return a list of agents for your account and AWS Region.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteLocation

Deletes the configuration of a location used by AWS DataSync.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 216)

The Amazon Resource Name (ARN) of the location to delete.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteTask

Deletes a task.

Request Syntax

```
{
   "TaskArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

TaskArn (p. 218)

The Amazon Resource Name (ARN) of the task to delete.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeAgent

Returns metadata such as the name, the network interfaces, and the status (that is, whether the agent is running or not) for an agent. To specify which agent to describe, use the Amazon Resource Name (ARN) of the agent in your request.

Request Syntax

```
{
   "AgentArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArn (p. 220)

The Amazon Resource Name (ARN) of the agent to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AgentArn": "string",
   "CreationTime": number,
   "EndpointType": "string",
   "LastConnectionTime": number,
   "Name": "string",
   "PrivateLinkConfig": {
      "PrivateLinkEndpoint": "string",
      "SecurityGroupArns": [ "string" ],
      "SubnetArns": [ "string" ],
      "VpcEndpointId": "string"
   },
   "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AgentArn (p. 220)

The Amazon Resource Name (ARN) of the agent.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

CreationTime (p. 220)

The time that the agent was activated (that is, created in your account).

Type: Timestamp

EndpointType (p. 220)

The type of endpoint that your agent is connected to. If the endpoint is a VPC endpoint, the agent is not accessible over the public internet.

Type: String

Valid Values: `PUBLIC | PRIVATE_LINK | FIPS`

LastConnectionTime (p. 220)

The time that the agent last connected to DataSync.

Type: Timestamp

Name (p. 220)

The name of the agent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

PrivateLinkConfig (p. 220)

The subnet and the security group that DataSync used to access a VPC endpoint.

Type: PrivateLinkConfig (p. 323) object

Status (p. 220)

The status of the agent. If the status is ONLINE, then the agent is configured properly and is available to use. The Running status is the normal running status for an agent. If the status is OFFLINE, the agent's VM is turned off or the agent is in an unhealthy state. When the issue that caused the unhealthy state is resolved, the agent returns to ONLINE status.

Type: String

Valid Values: `ONLINE | OFFLINE`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example returns information about the agent specified in the sample request.

Sample Request

```
{
  "AgentArn": "arn:aws:datasync:us-east-2:111222333444:agent/agent-0b0addbeef44baca3"
}
```

Example

This example illustrates one usage of DescribeAgent.

Sample Response

```
{
  "AgentArn": "arn:aws:datasync:us-east-2:111222333444:agent/agent-0b0addbeef44baca3",
  "CreationTime": "1532660733.39",
  "LastConnectionTime": "1532660733.39",
  "Name": "MyAgent",
  "Status": "ONLINE"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationEfs

Returns metadata about your AWS DataSync location for an Amazon EFS file system.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 223)

The Amazon Resource Name (ARN) of the Amazon EFS file system location that you want information about.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AccessPointArn": "string",
   "CreationTime": number,
   "Ec2Config": {
      "SecurityGroupArns": [ "string" ],
      "SubnetArn": "string"
   },
   "FileSystemAccessRoleArn": "string",
   "InTransitEncryption": "string",
   "LocationArn": "string",
   "LocationUri": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AccessPointArn (p. 223)

The ARN of the access point that DataSync uses to access the Amazon EFS file system.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):elasticfilesystem:[a-z\-0-9]+:[0-9]{12}:access-point/fsap-[0-9a-f]{8,40}$`

CreationTime (p. 223)

The time that the location was created.

Type: Timestamp

Ec2Config (p. 223)

The subnet and security groups that AWS DataSync uses to access your Amazon EFS file system.

Type: Ec2Config (p. 307) object

FileSystemAccessRoleArn (p. 223)

The AWS Identity and Access Management (IAM) role that DataSync assumes when mounting the Amazon EFS file system.

Type: String

Length Constraints: Maximum length of 2048.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`

InTransitEncryption (p. 223)

Describes whether DataSync uses Transport Layer Security (TLS) encryption when copying data to or from the Amazon EFS file system.

Type: String

Valid Values: `NONE | TLS1_2`

LocationArn (p. 223)

The ARN of the Amazon EFS file system location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 223)

The URL of the Amazon EFS file system location.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Sample Request

The following example shows how to get information about a specific Amazon EFS file system location.

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-12abcdef012345678"
}
```

Sample Response

The following example returns location details about an Amazon EFS file system.

```
{
    "CreationTime": 1653319021.353,
    "Ec2Config": {
        "SubnetArn": "arn:aws:ec2:us-east-2:111222333444:subnet/subnet-1234567890abcdef1",
        "SecurityGroupArns": [
            "arn:aws:ec2:us-east-2:111222333444:security-group/sg-1234567890abcdef2"
        ]
    },
    "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-
abcdef01234567890",
    "LocationUri": "efs://us-east-2.fs-021345abcdef6789/"
}
```

Sample Response: Describing a location for a restricted Amazon EFS file system

The following example returns location details about an Amazon EFS file system with restricted access, including the `AccessPointArn`, `FileSystemAccessRoleArn`, and `InTransitEncryption` elements.

```
{
    "CreationTime": 1653319021.353,
    "AccessPointArn": "arn:aws:elasticfilesystem:us-east-2:111222333444:access-point/
fsap-1234567890abcdef0",
    "Ec2Config": {
        "SubnetArn": "arn:aws:ec2:us-east-2:111222333444:subnet/subnet-1234567890abcdef1",
        "SecurityGroupArns": [
            "arn:aws:ec2:us-east-2:111222333444:security-group/sg-1234567890abcdef2"
        ]
    },
    "FileSystemAccessRoleArn": "arn:aws:iam::111222333444:role/AwsDataSyncFullAccessNew",
    "InTransitEncryption": "TLS1_2",
    "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-
abcdef01234567890",
    "LocationUri": "efs://us-east-2.fs-021345abcdef6789/",
    "Subdirectory": "/mount/path",
    "Tags": [{
        "Key": "Name",
        "Value": "ElasticFileSystem-1"
    }]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationFsxLustre

Returns metadata about an Amazon FSx for Lustre location, such as information about its path.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 227)

The Amazon Resource Name (ARN) of the FSx for Lustre location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "CreationTime": number,
   "LocationArn": "string",
   "LocationUri": "string",
   "SecurityGroupArns": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreationTime (p. 227)

The time that the FSx for Lustre location was created.

Type: Timestamp

LocationArn (p. 227)

The Amazon Resource Name (ARN) of the FSx for Lustre location that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 227)

The URI of the FSx for Lustre location that was described.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

SecurityGroupArns (p. 227)

The Amazon Resource Names (ARNs) of the security groups that are configured for the FSx for Lustre file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationFsxOpenZfs

Returns metadata about an Amazon FSx for OpenZFS location, such as information about its path.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 230)

The Amazon Resource Name (ARN) of the FSx for OpenZFS location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "CreationTime": number,
   "LocationArn": "string",
   "LocationUri": "string",
   "Protocol": {
      "NFS": {
         "MountOptions": {
            "Version": "string"
         }
      }
   },
   "SecurityGroupArns": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreationTime (p. 230)

The time that the FSx for OpenZFS location was created.

Type: Timestamp

LocationArn (p. 230)

The ARN of the FSx for OpenZFS location that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 230)

The uniform resource identifier (URI) of the FSx for OpenZFS location that was described.

Example: `fsxz://us-west-2.fs-1234567890abcdef02/fsx/folderA/folder`

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

Protocol (p. 230)

The type of protocol that AWS DataSync uses to access your file system.

Type: FsxProtocol (p. 309) object

SecurityGroupArns (p. 230)

The ARNs of the security groups that are configured for the FSx for OpenZFS file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationFsxWindows

Returns metadata about an Amazon FSx for Windows File Server location, such as information about its path.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 233)

The Amazon Resource Name (ARN) of the FSx for Windows File Server location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "CreationTime": number,
   "Domain": "string",
   "LocationArn": "string",
   "LocationUri": "string",
   "SecurityGroupArns": [ "string" ],
   "User": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreationTime (p. 233)

The time that the FSx for Windows File Server location was created.

Type: Timestamp

Domain (p. 233)

The name of the Windows domain that the FSx for Windows File Server belongs to.

Type: String

Length Constraints: Maximum length of 253.

Pattern: `^([A-Za-z0-9]+[A-Za-z0-9-.]*)*[A-Za-z0-9-]*[A-Za-z0-9]$`

LocationArn (p. 233)

The Amazon Resource Name (ARN) of the FSx for Windows File Server location that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 233)

The URL of the FSx for Windows File Server location that was described.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

SecurityGroupArns (p. 233)

The Amazon Resource Names (ARNs) of the security groups that are configured for the FSx for Windows File Server file system.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

User (p. 233)

The user who has the permissions to access files and folders in the FSx for Windows File Server file system.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^[^\x5B\x5D\\/:;|=,+*?]{1,104}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationHdfs

Returns metadata, such as the authentication information about the Hadoop Distributed File System (HDFS) location.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 236)

The Amazon Resource Name (ARN) of the HDFS cluster location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AgentArns": [ "string" ],
   "AuthenticationType": "string",
   "BlockSize": number,
   "CreationTime": number,
   "KerberosPrincipal": "string",
   "KmsKeyProviderUri": "string",
   "LocationArn": "string",
   "LocationUri": "string",
   "NameNodes": [
      {
         "Hostname": "string",
         "Port": number
      }
   ],
   "QopConfiguration": {
      "DataTransferProtection": "string",
      "RpcProtection": "string"
   },
   "ReplicationFactor": number,
   "SimpleUser": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AgentArns (p. 236)

The ARNs of the agents that are used to connect to the HDFS cluster.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

AuthenticationType (p. 236)

The type of authentication used to determine the identity of the user.

Type: String

Valid Values: `SIMPLE | KERBEROS`

BlockSize (p. 236)

The size of the data blocks to write into the HDFS cluster.

Type: Integer

Valid Range: Minimum value of 1048576. Maximum value of 1073741824.

CreationTime (p. 236)

The time that the HDFS location was created.

Type: Timestamp

KerberosPrincipal (p. 236)

The Kerberos principal with access to the files and folders on the HDFS cluster. This parameter is used if the `AuthenticationType` is defined as `KERBEROS`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^.+$`

KmsKeyProviderUri (p. 236)

The URI of the HDFS cluster's Key Management Server (KMS).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^kms:\/\/http[s]?@(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])(;(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9]))*:[0-9]{1,5}\/kms$`

LocationArn (p. 236)

The ARN of the HDFS cluster location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 236)

The URI of the HDFS cluster location.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

NameNodes (p. 236)

The NameNode that manage the HDFS namespace.

Type: Array of HdfsNameNode (p. 311) objects

Array Members: Minimum number of 1 item.

QopConfiguration (p. 236)

The Quality of Protection (QOP) configuration specifies the Remote Procedure Call (RPC) and data transfer protection settings configured on the Hadoop Distributed File System (HDFS) cluster.

Type: QopConfiguration (p. 325) object

ReplicationFactor (p. 236)

The number of DataNodes to replicate the data to when writing to the HDFS cluster.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 512.

SimpleUser (p. 236)

The user name used to identify the client on the host operating system. This parameter is used if the `AuthenticationType` is defined as `SIMPLE`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[_.A-Za-z0-9][-_.A-Za-z0-9]*$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationNfs

Returns metadata, such as the path information, about an NFS location.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 240)

The Amazon Resource Name (ARN) of the NFS location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "CreationTime": number,
   "LocationArn": "string",
   "LocationUri": "string",
   "MountOptions": {
      "Version": "string"
   },
   "OnPremConfig": {
      "AgentArns": [ "string" ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreationTime (p. 240)

The time that the NFS location was created.

Type: Timestamp

LocationArn (p. 240)

The Amazon Resource Name (ARN) of the NFS location that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 240)

The URL of the source NFS location that was described.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

MountOptions (p. 240)

The NFS mount options that DataSync used to mount your NFS share.

Type: NfsMountOptions (p. 315) object

OnPremConfig (p. 240)

A list of Amazon Resource Names (ARNs) of agents to use for a Network File System (NFS) location.

Type: OnPremConfig (p. 316) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example returns information about the NFS location specified in the sample request.

Sample Request

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50aa"
}
```

Example

This example illustrates one usage of DescribeLocationNfs.

Sample Response

```
{
   "CreationTime": 1532660733.39,
   "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50aa",
   "LocationUri": "hostname.amazon.com",
   "OnPremConfig": {
      "AgentArns": [ "arn:aws:datasync:us-east-2:111222333444:agent/
agent-0b0addbeef44b3nfs" ]
   }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationObjectStorage

Returns metadata about your AWS DataSync location for an object storage system.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 243)

The Amazon Resource Name (ARN) of the object storage system location that you want information about.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AccessKey": "string",
   "AgentArns": [ "string" ],
   "CreationTime": number,
   "LocationArn": "string",
   "LocationUri": "string",
   "ServerPort": number,
   "ServerProtocol": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AccessKey (p. 243)

The access key (for example, a user name) required to authenticate with the object storage server.

Type: String

Length Constraints: Minimum length of 8. Maximum length of 200.

Pattern: `^.+$`

AgentArns (p. 243)

The ARNs of the DataSync agents that can securely connect with your location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

CreationTime (p. 243)

The time that the location was created.

Type: Timestamp

LocationArn (p. 243)

The ARN of the object storage system location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 243)

The URL of the object storage system location.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

ServerPort (p. 243)

The port that your object storage server accepts inbound network traffic on (for example, port 443).

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65536.

ServerProtocol (p. 243)

The protocol that your object storage server uses to communicate.

Type: String

Valid Values: `HTTPS | HTTP`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationS3

Returns metadata, such as bucket name, about an Amazon S3 bucket location.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 246)

The Amazon Resource Name (ARN) of the Amazon S3 bucket location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AgentArns": [ "string" ],
   "CreationTime": number,
   "LocationArn": "string",
   "LocationUri": "string",
   "S3Config": {
      "BucketAccessRoleArn": "string"
   },
   "S3StorageClass": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AgentArns (p. 246)

If you are using DataSync on an AWS Outpost, the Amazon Resource Name (ARNs) of the EC2 agents deployed on your Outpost. For more information about launching a DataSync agent on an AWS Outpost, see Deploy your DataSync agent on AWS Outposts.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

CreationTime (p. 246)

The time that the Amazon S3 bucket location was created.

Type: Timestamp

LocationArn (p. 246)

The Amazon Resource Name (ARN) of the Amazon S3 bucket or access point.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 246)

The URL of the Amazon S3 location that was described.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

S3Config (p. 246)

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role used to access an Amazon S3 bucket.

For detailed information about using such a role, see Creating a Location for Amazon S3 in the *AWS DataSync User Guide*.

Type: S3Config (p. 326) object

S3StorageClass (p. 246)

The Amazon S3 storage class that you chose to store your files in when this location is used as a task destination. For more information about S3 storage classes, see Amazon S3 Storage Classes. Some storage classes have behaviors that can affect your S3 storage cost. For detailed information, see Considerations when working with S3 storage classes in DataSync.

Type: String

Valid Values: `STANDARD | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE | OUTPOSTS`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example returns information about the S3 location specified in the sample request.

Sample Request

```
{
  "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50s3"
}
```

Example

This example illustrates one usage of DescribeLocationS3.

Sample Response

```
{
   "CreationTime": 1532660733.39,
   "LocationArn": "arn:aws:datasync:us-east-2:111222333444:location/loc-07db7abfc326c50s3",
   "LocationUri": "MyBucket.",
   "S3Config": {
      "BucketAccessRoleArn": "arn:aws:iam::111222333444:role/MyBucketAccessRole",
   }
    "S3StorageClass": "STANDARD"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLocationSmb

Returns metadata, such as the path and user information about an SMB location.

Request Syntax

```
{
   "LocationArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 249)

The Amazon Resource Name (ARN) of the SMB location to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "AgentArns": [ "string" ],
   "CreationTime": number,
   "Domain": "string",
   "LocationArn": "string",
   "LocationUri": "string",
   "MountOptions": {
      "Version": "string"
   },
   "User": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AgentArns (p. 249)

The Amazon Resource Name (ARN) of the source SMB file system location that is created.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

CreationTime (p. 249)

The time that the SMB location was created.

Type: Timestamp

Domain (p. 249)

The name of the Windows domain that the SMB server belongs to.

Type: String

Length Constraints: Maximum length of 253.

Pattern: `^([A-Za-z0-9]+[A-Za-z0-9-.]*)*[A-Za-z0-9-]*[A-Za-z0-9]$`

LocationArn (p. 249)

The Amazon Resource Name (ARN) of the SMB location that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

LocationUri (p. 249)

The URL of the source SMB location that was described.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

MountOptions (p. 249)

The mount options that are available for DataSync to use to access an SMB location.

Type: SmbMountOptions (p. 327) object

User (p. 249)

The user who can mount the share, has the permissions to access files and folders in the SMB share.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^[^\x5B\x5D\\/:;|=,+*?]{1,104}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

This example illustrates one usage of DescribeLocationSmb.

Sample Request

```
{
  "arn:aws:datasync:us-east-1:111222333444:location/loc-0f01451b140b2af49"
}
```

Example

This example illustrates one usage of DescribeLocationSmb.

Sample Response

```
{
   "AgentArns":[
      "arn:aws:datasync:us-east-2:111222333444:agent/agent-0bc3b3dc9bbc15145",
      "arn:aws:datasync:us-east-2:111222333444:agent/agent-04b3fe3d261a18c8f"
   ],
   "CreationTime":"1532660733.39",
   "Domain":"AMAZON",
   "LocationArn":"arn:aws:datasync:us-east-1:111222333444:location/loc-0f01451b140b2af49",
   "LocationUri":"smb://hostname.amazon.com/share",
   "MountOptions":{
      "Version":"SMB3"
   },
   "User":"user-1"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTask

Returns metadata about a task.

Request Syntax

```
{
   "TaskArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

TaskArn (p. 252)

The Amazon Resource Name (ARN) of the task to describe.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: Yes

Response Syntax

```
{
   "CloudWatchLogGroupArn": "string",
   "CreationTime": number,
   "CurrentTaskExecutionArn": "string",
   "DestinationLocationArn": "string",
   "DestinationNetworkInterfaceArns": [ "string" ],
   "ErrorCode": "string",
   "ErrorDetail": "string",
   "Excludes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Includes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Name": "string",
   "Options": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "Schedule": {
      "ScheduleExpression": "string"
   },
   "SourceLocationArn": "string",
   "SourceNetworkInterfaceArns": [ "string" ],
   "Status": "string",
   "TaskArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CloudWatchLogGroupArn (p. 252)

The Amazon Resource Name (ARN) of the Amazon CloudWatch log group that was used to monitor and log events in the task.

For more information on these groups, see Working with Log Groups and Log Streams in the *Amazon CloudWatch User Guide*.

Type: String

Length Constraints: Maximum length of 562.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):logs:[a-z\-0-9]*:[0-9]{12}:log-group:([^:\*]*)(:\*)?$`

CreationTime (p. 252)

The time that the task was created.

Type: Timestamp

CurrentTaskExecutionArn (p. 252)

The Amazon Resource Name (ARN) of the task execution that is syncing files.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

DestinationLocationArn (p. 252)

The Amazon Resource Name (ARN) of the AWS storage resource's location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

DestinationNetworkInterfaceArns (p. 252)

The Amazon Resource Names (ARNs) of the destination elastic network interfaces (ENIs) that were created for your subnet.

Type: Array of strings

Length Constraints: Maximum length of 128.

Pattern: `^arn:aws[\-a-z]{0,}:ec2:[a-z\-0-9]*:[0-9]{12}:network-interface/eni-[0-9a-f]+$`

ErrorCode (p. 252)

Errors that AWS DataSync encountered during execution of the task. You can use this error code to help troubleshoot issues.

Type: String

ErrorDetail (p. 252)

Detailed description of an error that was encountered during the task execution. You can use this information to help troubleshoot issues.

Type: String

Excludes (p. 252)

A list of filter rules that determines which files to exclude from a task. The list should contain a single filter string that consists of the patterns to exclude. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Includes (p. 252)

A list of filter rules that determines which files to include when running a task. The pattern contains a single filter string that consists of the patterns to include. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2`".

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Name (p. 252)

The name of the task that was described.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Options (p. 252)

The set of configuration options that control the behavior of a single execution of the task that occurs when you call `StartTaskExecution`. You can configure these options to preserve metadata such as user ID (UID) and group (GID), file permissions, data integrity verification, and so on.

For each individual task execution, you can override these options by specifying the overriding `OverrideOptions` value to StartTaskExecution operation.

Type: Options (p. 317) object

Schedule (p. 252)

The schedule used to periodically transfer files from a source to a destination location.

Type: TaskSchedule (p. 334) object

SourceLocationArn (p. 252)

The Amazon Resource Name (ARN) of the source file system's location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

SourceNetworkInterfaceArns (p. 252)

The Amazon Resource Names (ARNs) of the source elastic network interfaces (ENIs) that were created for your subnet.

Type: Array of strings

Length Constraints: Maximum length of 128.

Pattern: `^arn:aws[\-a-z]{0,}:ec2:[a-z\-0-9]*:[0-9]{12}:network-interface/eni-[0-9a-f]+$`

Status (p. 252)

The status of the task that was described.

For detailed information about task execution statuses, see Understanding Task Statuses in the *AWS DataSync User Guide.*

Type: String

Valid Values: `AVAILABLE | CREATING | QUEUED | RUNNING | UNAVAILABLE`

TaskArn (p. 252)

The Amazon Resource Name (ARN) of the task that was described.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example returns information about the task specified in the sample request.

Sample Request

```
{
  "TaskArn": "arn:aws:datasync:us-east-2:111222333444:task/task-08de6e6697796f026"
}
```

Example

This example illustrates one usage of DescribeTask.

Sample Response

```
{
   "CloudWatchLogGroupArn": "arn:aws:logs:us-east-2:111222333444:log-group"
   "CreationTime": 1532660733.39,
   "CurrentTaskExecutionArn": "arn:aws:datasync:us-east-2:111222333444:task/
task-08de6e6697796f026/execution/exec-04ce9d516d69bd52f",
   "Options": {
      "Atime": "BEST_EFFORT",
      "BytesPerSecond": 1000,
      "Gid": "NONE",
      "Mtime": "PRESERVE",
      "PosixPermissions": "PRESERVE",
      "PreserveDevices": "NONE",
      "PreserveDeletedFiles": "PRESERVE",
      "Uid": "NONE",
      "VerifyMode": "POINT_IN_TIME_CONSISTENT"
   },
   "DestinationLocationArn": "arn:aws:datasync:us-east-2:111222333444:location/
loc-07db7abfc326c50fb",
   "ErrorCode": "???????",
   "ErrorDetail": "??????",
   "Name": "MyTask",
   "SourceLocationArn": "arn:aws:datasync:us-east-2:111222333444:location/
loc-07db7abfc326c50aa",
   "Status": "CREATING",
   "TaskArn": "arn:aws:datasync:us-east-2:111222333444:task/task-08de6e6697796f026"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTaskExecution

Returns detailed metadata about a task that is being executed.

Request Syntax

```
{
   "TaskExecutionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

TaskExecutionArn (p. 258)

The Amazon Resource Name (ARN) of the task that is being executed.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Required: Yes

Response Syntax

```
{
   "BytesTransferred": number,
   "BytesWritten": number,
   "EstimatedBytesToTransfer": number,
   "EstimatedFilesToTransfer": number,
   "Excludes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "FilesTransferred": number,
   "Includes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Options": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "Result": {
      "ErrorCode": "string",
      "ErrorDetail": "string",
      "PrepareDuration": number,
      "PrepareStatus": "string",
      "TotalDuration": number,
      "TransferDuration": number,
      "TransferStatus": "string",
      "VerifyDuration": number,
      "VerifyStatus": "string"
   },
   "StartTime": number,
   "Status": "string",
   "TaskExecutionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BytesTransferred (p. 258)

The physical number of bytes transferred over the network.

Type: Long

BytesWritten (p. 258)

The number of logical bytes written to the destination AWS storage resource.

Type: Long

EstimatedBytesToTransfer (p. 258)

The estimated physical number of bytes that is to be transferred over the network.

Type: Long

EstimatedFilesToTransfer (p. 258)

The expected number of files that is to be transferred over the network. This value is calculated during the PREPARING phase, before the TRANSFERRING phase. This value is the expected number of files to be transferred. It's calculated based on comparing the content of the source and destination locations and finding the delta that needs to be transferred.

Type: Long

Excludes (p. 258)

A list of filter rules that determines which files to exclude from a task. The list should contain a single filter string that consists of the patterns to exclude. The patterns are delimited by "|" (that is, a pipe), for example: `"/folder1|/folder2"`

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

FilesTransferred (p. 258)

The actual number of files that was transferred over the network. This value is calculated and updated on an ongoing basis during the TRANSFERRING phase. It's updated periodically when each file is read from the source and sent over the network.

If failures occur during a transfer, this value can be less than `EstimatedFilesToTransfer`. This value can also be greater than `EstimatedFilesTransferred` in some cases. This element is implementation-specific for some location types, so don't use it as an indicator for a correct file number or to monitor your task execution.

Type: Long

Includes (p. 258)

A list of filter rules that determines which files to include when running a task. The list should contain a single filter string that consists of the patterns to include. The patterns are delimited by "|" (that is, a pipe), for example: `"/folder1|/folder2"`

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Options (p. 258)

Represents the options that are available to control the behavior of a StartTaskExecution operation. Behavior includes preserving metadata such as user ID (UID), group ID (GID), and file permissions, and also overwriting files in the destination, data integrity verification, and so on.

A task has a set of default options associated with it. If you don't specify an option in StartTaskExecution, the default value is used. You can override the defaults options on each task execution by specifying an overriding `Options` value to StartTaskExecution.

Type: Options (p. 317) object

Result (p. 258)

The result of the task execution.

Type: TaskExecutionResultDetail (p. 330) object

StartTime (p. 258)

The time that the task execution was started.

Type: Timestamp

Status (p. 258)

The status of the task execution.

For detailed information about task execution statuses, see Understanding Task Statuses.

Type: String

Valid Values: `QUEUED | LAUNCHING | PREPARING | TRANSFERRING | VERIFYING | SUCCESS | ERROR`

TaskExecutionArn (p. 258)

The Amazon Resource Name (ARN) of the task execution that was described. `TaskExecutionArn` is hierarchical and includes `TaskArn` for the task that was executed.

For example, a `TaskExecution` value with the ARN `arn:aws:datasync:us-east-1:111222333444:task/task-0208075f79cedf4a2/execution/exec-08ef1e88ec491019b` executed the task with the ARN `arn:aws:datasync:us-east-1:111222333444:task/task-0208075f79cedf4a2`.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example returns information about the `TaskExecution` value specified in the sample request.

Sample Request

```
{
  "TaskExecutionArn": "arn:aws:datasync:us-east-1:111222333444:task/task-08de6e6697796f026/
execution/exec-04ce9d516d69bd52f"
}
```

Example

This example illustrates one usage of DescribeTaskExecution.

Sample Response

```
{
   "BytesTransferred": "5000",
   "BytesWritten": "5000",
   "EstimatedBytesToTransfer": "5000",
   "EstimatedFilesToTransfer": "100",
   "FilesTransferred": "100",
   "Result": {
      "ErrorCode": "??????",
      "ErrorDetail": "??????",
      "PrepareDuration": "100",
      "PrepareStatus": "SUCCESS",
      "TransferDuration": "60",
      "TransferStatus": "AVAILABLE",
      "VerifyDuration": "30",
      "VerifyStatus": "SUCCESS"
   },
   "StartTime": "1532660733.39",
   "Status": "SUCCESS",
   "OverrideOptions": {
      "Atime": "BEST_EFFORT",
      "BytesPerSecond": "1000",
      "Gid": "NONE",
      "Mtime": "PRESERVE",
      "PosixPermissions": "PRESERVE",
      "PreserveDevices": "NONE",
      "PreserveDeletedFiles": "PRESERVE",
      "Uid": "NONE",
      "VerifyMode": "POINT_IN_TIME_CONSISTENT"
   },
   "TaskExecutionArn": "arn:aws:datasync:us-east-2:111222333444:task/
task-08de6e6697796f026/execution/exec-04ce9d516d69bd52f"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListAgents

Returns a list of agents owned by an AWS account in the AWS Region specified in the request. The returned list is ordered by agent Amazon Resource Name (ARN).

By default, this operation returns a maximum of 100 agents. This operation supports pagination that enables you to optionally reduce the number of agents returned in a response.

If you have more agents than are returned in a response (that is, the response returns only a truncated list of your agents), the response contains a marker that you can specify in your next request to fetch the next page of agents.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

MaxResults (p. 263)

The maximum number of agents to list.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NextToken (p. 263)

An opaque string that indicates the position at which to begin the next list of agents.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Required: No

Response Syntax

```
{
   "Agents": [
      {
         "AgentArn": "string",
         "Name": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Agents (p. 263)

A list of agents in your account.

Type: Array of AgentListEntry (p. 306) objects

NextToken (p. 263)

An opaque string that indicates the position at which to begin returning the next list of agents.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListLocations

Returns a list of source and destination locations.

If you have more locations than are returned in a response (that is, the response returns only a truncated list of your agents), the response contains a token that you can specify in your next request to fetch the next page of locations.

Request Syntax

```
{
   "Filters": [
      {
         "Name": "string",
         "Operator": "string",
         "Values": [ "string" ]
      }
   ],
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Filters (p. 265)

You can use API filters to narrow down the list of resources returned by `ListLocations`. For example, to retrieve all tasks on a specific source location, you can use `ListLocations` with filter name `LocationType S3` and `Operator Equals`.

Type: Array of LocationFilter (p. 312) objects

Required: No

MaxResults (p. 265)

The maximum number of locations to return.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NextToken (p. 265)

An opaque string that indicates the position at which to begin the next list of locations.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Required: No

Response Syntax

```
{
   "Locations": [
      {
         "LocationArn": "string",
         "LocationUri": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Locations (p. 266)

An array that contains a list of locations.

Type: Array of LocationListEntry (p. 313) objects

NextToken (p. 266)

An opaque string that indicates the position at which to begin returning the next list of locations.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go

- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Returns all the tags associated with a specified resource.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ResourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

MaxResults (p. 268)

The maximum number of locations to return.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NextToken (p. 268)

An opaque string that indicates the position at which to begin the next list of locations.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Required: No

ResourceArn (p. 268)

The Amazon Resource Name (ARN) of the resource whose tags to list.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:(agent|task|location)/(agent|task|loc)-[0-9a-z]{17}$`

Required: Yes

Response Syntax

```
{
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 268)

An opaque string that indicates the position at which to begin returning the next list of resource tags.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Tags (p. 268)

Array of resource tags.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 55 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2

- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTaskExecutions

Returns a list of executed tasks.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "TaskArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

MaxResults (p. 271)

The maximum number of executed tasks to list.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NextToken (p. 271)

An opaque string that indicates the position at which to begin the next list of the executed tasks.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Required: No

TaskArn (p. 271)

The Amazon Resource Name (ARN) of the task whose tasks you want to list.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "TaskExecutions": [
      {
         "Status": "string",
         "TaskExecutionArn": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 271)

An opaque string that indicates the position at which to begin returning the next list of executed tasks.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

TaskExecutions (p. 271)

A list of executed tasks.

Type: Array of TaskExecutionListEntry (p. 329) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

ListTasks

Returns a list of all the tasks.

Request Syntax

```
{
   "Filters": [
      {
         "Name": "string",
         "Operator": "string",
         "Values": [ "string" ]
      }
   ],
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Filters (p. 274)

You can use API filters to narrow down the list of resources returned by `ListTasks`. For example, to retrieve all tasks on a specific source location, you can use `ListTasks` with filter name `LocationId` and `Operator Equals` with the ARN for the location.

Type: Array of TaskFilter (p. 332) objects

Required: No

MaxResults (p. 274)

The maximum number of tasks to return.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NextToken (p. 274)

An opaque string that indicates the position at which to begin the next list of tasks.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Tasks": [
      {
         "Name": "string",
         "Status": "string",
         "TaskArn": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 274)

An opaque string that indicates the position at which to begin returning the next list of tasks.

Type: String

Length Constraints: Maximum length of 65535.

Pattern: `[a-zA-Z0-9=_-]+`

Tasks (p. 274)

A list of all the tasks that are returned.

Type: Array of TaskListEntry (p. 333) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartTaskExecution

Starts a specific invocation of a task. A `TaskExecution` value represents an individual run of a task. Each task can have at most one `TaskExecution` at a time.

`TaskExecution` has the following transition phases: INITIALIZING | PREPARING | TRANSFERRING | VERIFYING | SUCCESS/FAILURE.

For detailed information, see Task Execution in Components and Terminology in the *AWS DataSync User Guide*.

Request Syntax

```
{
   "Excludes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Includes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "OverrideOptions": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "TaskArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Excludes (p. 277)

A list of filter rules that determines which files to exclude from a task. The list contains a single filter string that consists of the patterns to exclude. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

Includes (p. 277)

A list of filter rules that determines which files to include when running a task. The pattern should contain a single filter string that consists of the patterns to include. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

OverrideOptions (p. 277)

Represents the options that are available to control the behavior of a StartTaskExecution operation. Behavior includes preserving metadata such as user ID (UID), group ID (GID), and file permissions, and also overwriting files in the destination, data integrity verification, and so on.

A task has a set of default options associated with it. If you don't specify an option in StartTaskExecution, the default value is used. You can override the defaults options on each task execution by specifying an overriding `Options` value to StartTaskExecution.

Type: Options (p. 317) object

Required: No

TaskArn (p. 277)

The Amazon Resource Name (ARN) of the task to start.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: Yes

Response Syntax

```
{
   "TaskExecutionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TaskExecutionArn (p. 278)

The Amazon Resource Name (ARN) of the specific task execution that was started.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

Examples

Example

The following example starts a task execution using the default options and the specified task.

Sample Request

```
{
   "OverrideOptions": {
      "Atime": "BEST_EFFORT",
      "BytesPerSecond": 1000,
      "Gid": "NONE",
      "Mtime": "PRESERVE",
      "PosixPermissions": "PRESERVE",
      "PreserveDevices": "NONE",
      "PreserveDeletedFiles": "PRESERVE",
      "Uid": "NONE",
      "VerifyMode": "POINT_IN_TIME_CONSISTENT"
   },
   "TaskArn": "arn:aws:datasync:us-east-2:111222333444:task/task-08de6e6697796f026"
}
```

Example

This example illustrates one usage of StartTaskExecution.

Sample Response

```
{
  "TaskExecutionArn": "arn:aws:datasync:us-east-2:111222333444:task/task-08de6e6697796f026/
execution/exec-04ce9d516d69bd52f"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Applies a key-value pair to an AWS resource.

Request Syntax

```
{
   "ResourceArn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

ResourceArn (p. 281)

The Amazon Resource Name (ARN) of the resource to apply the tag to.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:(agent|task|location)/(agent|task|loc)-[0-9a-z]{17}$`

Required: Yes

Tags (p. 281)

The tags to apply.

Type: Array of TagListEntry (p. 328) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Removes a tag from an AWS resource.

Request Syntax

```
{
   "Keys": [ "string" ],
   "ResourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Keys (p. 283)

The keys in the key-value pair in the tag to remove.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 50 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:/-]+$`

Required: Yes

ResourceArn (p. 283)

The Amazon Resource Name (ARN) of the resource to remove the tag from.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:(agent|task|location)/(agent|task|loc)-[0-9a-z]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateAgent

Updates the name of an agent.

Request Syntax

```
{
   "AgentArn": "string",
   "Name": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArn (p. 285)

The Amazon Resource Name (ARN) of the agent to update.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

Name (p. 285)

The name that you want to use to configure the agent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateLocationHdfs

Updates some parameters of a previously created location for a Hadoop Distributed File System cluster.

Request Syntax

```
{
   "AgentArns": [ "string" ],
   "AuthenticationType": "string",
   "BlockSize": number,
   "KerberosKeytab": blob,
   "KerberosKrb5Conf": blob,
   "KerberosPrincipal": "string",
   "KmsKeyProviderUri": "string",
   "LocationArn": "string",
   "NameNodes": [
      {
         "Hostname": "string",
         "Port": number
      }
   ],
   "QopConfiguration": {
      "DataTransferProtection": "string",
      "RpcProtection": "string"
   },
   "ReplicationFactor": number,
   "SimpleUser": "string",
   "Subdirectory": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArns (p. 287)

The ARNs of the agents that are used to connect to the HDFS cluster.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: No

AuthenticationType (p. 287)

The type of authentication used to determine the identity of the user.

Type: String

Valid Values: `SIMPLE | KERBEROS`

Required: No

BlockSize (p. 287)

The size of the data blocks to write into the HDFS cluster.

Type: Integer

Valid Range: Minimum value of 1048576. Maximum value of 1073741824.

Required: No

KerberosKeytab (p. 287)

The Kerberos key table (keytab) that contains mappings between the defined Kerberos principal and the encrypted keys. You can load the keytab from a file by providing the file's address. If you use the AWS CLI, it performs base64 encoding for you. Otherwise, provide the base64-encoded text.

Type: Base64-encoded binary data object

Length Constraints: Maximum length of 65536.

Required: No

KerberosKrb5Conf (p. 287)

The `krb5.conf` file that contains the Kerberos configuration information. You can load the `krb5.conf` file by providing the file's address. If you're using the AWS CLI, it performs the base64 encoding for you. Otherwise, provide the base64-encoded text.

Type: Base64-encoded binary data object

Length Constraints: Maximum length of 131072.

Required: No

KerberosPrincipal (p. 287)

The Kerberos principal with access to the files and folders on the HDFS cluster.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^.+$`

Required: No

KmsKeyProviderUri (p. 287)

The URI of the HDFS cluster's Key Management Server (KMS).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^kms:\/\/http[s]?@(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])(;(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9]))*:[0-9]{1,5}\/kms$`

Required: No

LocationArn (p. 287)

The Amazon Resource Name (ARN) of the source HDFS cluster location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

NameNodes (p. 287)

The NameNode that manages the HDFS namespace. The NameNode performs operations such as opening, closing, and renaming files and directories. The NameNode contains the information to map blocks of data to the DataNodes. You can use only one NameNode.

Type: Array of HdfsNameNode (p. 311) objects

Array Members: Minimum number of 1 item.

Required: No

QopConfiguration (p. 287)

The Quality of Protection (QOP) configuration specifies the Remote Procedure Call (RPC) and data transfer privacy settings configured on the Hadoop Distributed File System (HDFS) cluster.

Type: QopConfiguration (p. 325) object

Required: No

ReplicationFactor (p. 287)

The number of DataNodes to replicate the data to when writing to the HDFS cluster.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 512.

Required: No

SimpleUser (p. 287)

The user name used to identify the client on the host operating system.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[_.A-Za-z0-9][-_.A-Za-z0-9]*$`

Required: No

Subdirectory (p. 287)

A subdirectory in the HDFS cluster. This subdirectory is used to read data from or write data to the HDFS cluster.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateLocationNfs

Updates some of the parameters of a previously created location for Network File System (NFS) access. For information about creating an NFS location, see Creating a location for NFS.

Request Syntax

```
{
   "LocationArn": "string",
   "MountOptions": { 
      "Version": "string"
   },
   "OnPremConfig": { 
      "AgentArns": [ "string" ]
   },
   "Subdirectory": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

LocationArn (p. 291)

The Amazon Resource Name (ARN) of the NFS location to update.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

MountOptions (p. 291)

Represents the mount options that are available for DataSync to access an NFS location.

Type: NfsMountOptions (p. 315) object

Required: No

OnPremConfig (p. 291)

A list of Amazon Resource Names (ARNs) of agents to use for a Network File System (NFS) location.

Type: OnPremConfig (p. 316) object

Required: No

Subdirectory (p. 291)

The subdirectory in the NFS file system that is used to read data from the NFS source location or write data to the NFS destination. The NFS path should be a path that's exported by the NFS server, or a subdirectory of that path. The path should be such that it can be mounted by other NFS clients in your network.

To see all the paths exported by your NFS server, run "`showmount -e nfs-server-name`" from an NFS client that has access to your server. You can specify any directory that appears in the results, and any subdirectory of that directory. Ensure that the NFS export is accessible without Kerberos authentication.

To transfer all the data in the folder that you specified, DataSync must have permissions to read all the data. To ensure this, either configure the NFS export with `no_root_squash`, or ensure that the files you want DataSync to access have permissions that allow read access for all users. Doing either option enables the agent to read the files. For the agent to access directories, you must additionally enable all execute access.

If you are copying data to or from your AWS Snowcone device, see NFS Server on AWS Snowcone for more information.

For information about NFS export configuration, see 18.7. The /etc/exports Configuration File in the Red Hat Enterprise Linux documentation.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]+$`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateLocationObjectStorage

Updates some of the parameters of a previously created location for self-managed object storage server access. For information about creating a self-managed object storage location, see Creating a location for object storage.

Request Syntax

```
{
   "AccessKey": "string",
   "AgentArns": [ "string" ],
   "LocationArn": "string",
   "SecretKey": "string",
   "ServerPort": number,
   "ServerProtocol": "string",
   "Subdirectory": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AccessKey (p. 294)

Optional. The access key is used if credentials are required to access the self-managed object storage server. If your object storage requires a user name and password to authenticate, use `AccessKey` and `SecretKey` to provide the user name and password, respectively.

Type: String

Length Constraints: Minimum length of 8. Maximum length of 200.

Pattern: `^.+$`

Required: No

AgentArns (p. 294)

The Amazon Resource Name (ARN) of the agents associated with the self-managed object storage server location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: No

LocationArn (p. 294)

The Amazon Resource Name (ARN) of the self-managed object storage server location to be updated.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

SecretKey (p. 294)

Optional. The secret key is used if credentials are required to access the self-managed object storage server. If your object storage requires a user name and password to authenticate, use `AccessKey` and `SecretKey` to provide the user name and password, respectively.

Type: String

Length Constraints: Minimum length of 8. Maximum length of 200.

Pattern: `^.+$`

Required: No

ServerPort (p. 294)

The port that your self-managed object storage server accepts inbound network traffic on. The server port is set by default to TCP 80 (HTTP) or TCP 443 (HTTPS). You can specify a custom port if your self-managed object storage server requires one.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65536.

Required: No

ServerProtocol (p. 294)

The protocol that the object storage server uses to communicate. Valid values are `HTTP` or `HTTPS`.

Type: String

Valid Values: `HTTPS | HTTP`

Required: No

Subdirectory (p. 294)

The subdirectory in the self-managed object storage server that is used to read data from.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\p{Zs}]*$`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateLocationSmb

Updates some of the parameters of a previously created location for Server Message Block (SMB) file system access. For information about creating an SMB location, see Creating a location for SMB.

Request Syntax

```
{
   "AgentArns": [ "string" ],
   "Domain": "string",
   "LocationArn": "string",
   "MountOptions": {
      "Version": "string"
   },
   "Password": "string",
   "Subdirectory": "string",
   "User": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

AgentArns (p. 297)

The Amazon Resource Names (ARNs) of agents to use for a Simple Message Block (SMB) location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: No

Domain (p. 297)

The name of the Windows domain that the SMB server belongs to.

Type: String

Length Constraints: Maximum length of 253.

Pattern: `^([A-Za-z0-9]+[A-Za-z0-9-.]*)*[A-Za-z0-9-]*[A-Za-z0-9]$`

Required: No

LocationArn (p. 297)

The Amazon Resource Name (ARN) of the SMB location to update.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: Yes

MountOptions (p. 297)

Represents the mount options that are available for DataSync to access an SMB location.

Type: SmbMountOptions (p. 327) object

Required: No

Password (p. 297)

The password of the user who can mount the share has the permissions to access files and folders in the SMB share.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^.{0,104}$`

Required: No

Subdirectory (p. 297)

The subdirectory in the SMB file system that is used to read data from the SMB source location or write data to the SMB destination. The SMB path should be a path that's exported by the SMB server, or a subdirectory of that path. The path should be such that it can be mounted by other SMB clients in your network.

> **Note**
> `Subdirectory` must be specified with forward slashes. For example, `/path/to/folder`.

To transfer all the data in the folder that you specified, DataSync must have permissions to mount the SMB share and to access all the data in that share. To ensure this, do either of the following:

- Ensure that the user/password specified belongs to the user who can mount the share and who has the appropriate permissions for all of the files and directories that you want DataSync to access.
- Use credentials of a member of the Backup Operators group to mount the share.

Doing either of these options enables the agent to access the data. For the agent to access directories, you must also enable all execute access.

Type: String

Length Constraints: Maximum length of 4096.

Pattern: `^[a-zA-Z0-9_\-\+\./\(\)\$\p{Zs}]+$`

Required: No

User (p. 297)

The user who can mount the share has the permissions to access files and folders in the SMB share.

Type: String

Length Constraints: Maximum length of 104.

Pattern: `^[^\x5B\x5D\\/:;|=,+*?]{1,104}$`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTask

Updates the metadata associated with a task.

Request Syntax

```
{
   "CloudWatchLogGroupArn": "string",
   "Excludes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Includes": [
      {
         "FilterType": "string",
         "Value": "string"
      }
   ],
   "Name": "string",
   "Options": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "Schedule": {
      "ScheduleExpression": "string"
   },
   "TaskArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

CloudWatchLogGroupArn (p. 300)

The Amazon Resource Name (ARN) of the resource name of the Amazon CloudWatch log group.

Type: String

Length Constraints: Maximum length of 562.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):logs:[a-z\-0-9]*:[0-9]{12}:log-group:([^:\*]*)(:\*)?$`

Required: No

Excludes (p. 300)

A list of filter rules that determines which files to exclude from a task. The list should contain a single filter string that consists of the patterns to exclude. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

Includes (p. 300)

A list of filter rules that determines which files to include when running a task. The pattern contains a single filter string that consists of the patterns to include. The patterns are delimited by "|" (that is, a pipe), for example, `"/folder1|/folder2"`.

Type: Array of FilterRule (p. 308) objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Required: No

Name (p. 300)

The name of the task to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

Options (p. 300)

Represents the options that are available to control the behavior of a StartTaskExecution operation. Behavior includes preserving metadata such as user ID (UID), group ID (GID), and file permissions, and also overwriting files in the destination, data integrity verification, and so on.

A task has a set of default options associated with it. If you don't specify an option in StartTaskExecution, the default value is used. You can override the defaults options on each task execution by specifying an overriding `Options` value to StartTaskExecution.

Type: Options (p. 317) object

Required: No

Schedule (p. 300)

Specifies a schedule used to periodically transfer files from a source to a destination location. You can configure your task to execute hourly, daily, weekly or on specific days of the week. You control when in the day or hour you want the task to execute. The time you specify is UTC time. For more information, see Scheduling your task.

Type: TaskSchedule (p. 334) object

Required: No

TaskArn (p. 300)

The Amazon Resource Name (ARN) of the resource name of the task to update.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTaskExecution

Updates execution of a task.

You can modify bandwidth throttling for a task execution that is running or queued. For more information, see Adjusting Bandwidth Throttling for a Task Execution.

> **Note**
> The only `Option` that can be modified by `UpdateTaskExecution` is `BytesPerSecond` .

Request Syntax

```
{
   "Options": {
      "Atime": "string",
      "BytesPerSecond": number,
      "Gid": "string",
      "LogLevel": "string",
      "Mtime": "string",
      "ObjectTags": "string",
      "OverwriteMode": "string",
      "PosixPermissions": "string",
      "PreserveDeletedFiles": "string",
      "PreserveDevices": "string",
      "SecurityDescriptorCopyFlags": "string",
      "TaskQueueing": "string",
      "TransferMode": "string",
      "Uid": "string",
      "VerifyMode": "string"
   },
   "TaskExecutionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 336).

The request accepts the following data in JSON format.

Options (p. 303)

Represents the options that are available to control the behavior of a StartTaskExecution operation. Behavior includes preserving metadata such as user ID (UID), group ID (GID), and file permissions, and also overwriting files in the destination, data integrity verification, and so on.

A task has a set of default options associated with it. If you don't specify an option in StartTaskExecution, the default value is used. You can override the defaults options on each task execution by specifying an overriding `Options` value to StartTaskExecution.

Type: Options (p. 317) object

Required: Yes

TaskExecutionArn (p. 303)

The Amazon Resource Name (ARN) of the specific task execution that is being updated.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 334).

InternalException

This exception is thrown when an error occurs in the AWS DataSync service.

HTTP Status Code: 500

InvalidRequestException

This exception is thrown when the client submits a malformed request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The following data types are supported:

- AgentListEntry (p. 306)
- Ec2Config (p. 307)
- FilterRule (p. 308)
- FsxProtocol (p. 309)
- FsxProtocolNfs (p. 310)
- HdfsNameNode (p. 311)
- LocationFilter (p. 312)
- LocationListEntry (p. 313)

- NfsMountOptions (p. 315)
- OnPremConfig (p. 316)
- Options (p. 317)
- PrivateLinkConfig (p. 323)
- QopConfiguration (p. 325)
- S3Config (p. 326)
- SmbMountOptions (p. 327)
- TagListEntry (p. 328)
- TaskExecutionListEntry (p. 329)
- TaskExecutionResultDetail (p. 330)
- TaskFilter (p. 332)
- TaskListEntry (p. 333)
- TaskSchedule (p. 334)

AgentListEntry

Represents a single entry in a list of agents. `AgentListEntry` returns an array that contains a list of agents when the ListAgents operation is called.

Contents

AgentArn

The Amazon Resource Name (ARN) of the agent.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: No

Name

The name of the agent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

Status

The status of the agent.

Type: String

Valid Values: `ONLINE | OFFLINE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Ec2Config

The subnet and security groups that AWS DataSync uses to access your Amazon EFS file system.

Contents

SecurityGroupArns

Specifies the Amazon Resource Names (ARNs) of the security groups associated with an Amazon EFS file system's mount target.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: Yes

SubnetArn

Specifies the ARN of a subnet where DataSync creates the network interfaces for managing traffic during your transfer.

The subnet must be located:

- In the same virtual private cloud (VPC) as the Amazon EFS file system.
- In the same Availability Zone as at least one mount target for the Amazon EFS file system.

Note
You don't need to specify a subnet that includes a file system mount target.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:subnet/.*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FilterRule

Specifies which files, folders, and objects to include or exclude when transferring files from source to destination.

Contents

FilterType

The type of filter rule to apply. AWS DataSync only supports the SIMPLE_PATTERN rule type.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^[A-Z0-9_]+$`

Valid Values: `SIMPLE_PATTERN`

Required: No

Value

A single filter string that consists of the patterns to include or exclude. The patterns are delimited by "|" (that is, a pipe), for example: `/folder1|/folder2`

Type: String

Length Constraints: Maximum length of 102400.

Pattern: `^[^\x00]+$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FsxProtocol

Represents the protocol that AWS DataSync uses to access your Amazon FSx for OpenZFS file system.

Contents

NFS

> Represents the Network File System (NFS) protocol that DataSync uses to access your FSx for OpenZFS file system.
>
> Type: FsxProtocolNfs (p. 310) object
>
> Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FsxProtocolNfs

Represents the Network File System (NFS) protocol that AWS DataSync uses to access your Amazon FSx for OpenZFS file system.

Contents

MountOptions

Represents the mount options that are available for DataSync to access an NFS location.

Type: NfsMountOptions (p. 315) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HdfsNameNode

The NameNode of the Hadoop Distributed File System (HDFS). The NameNode manages the file system's namespace. The NameNode performs operations such as opening, closing, and renaming files and directories. The NameNode contains the information to map blocks of data to the DataNodes.

Contents

Hostname

The hostname of the NameNode in the HDFS cluster. This value is the IP address or Domain Name Service (DNS) name of the NameNode. An agent that's installed on-premises uses this hostname to communicate with the NameNode in the network.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^(([a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9\-]*[A-Za-z0-9])$`

Required: Yes

Port

The port that the NameNode uses to listen to client requests.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65536.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocationFilter

You can use API filters to narrow down the list of resources returned by `ListLocations`. For example, to retrieve all your Amazon S3 locations, you can use `ListLocations` with filter name `LocationType S3` and `Operator Equals`.

Contents

Name

The name of the filter being used. Each API call supports a list of filters that are available for it (for example, `LocationType` for `ListLocations`).

Type: String

Valid Values: `LocationUri | LocationType | CreationTime`

Required: Yes

Operator

The operator that is used to compare filter values (for example, `Equals` or `Contains`). For more about API filtering operators, see API filters for ListTasks and ListLocations.

Type: String

Valid Values: `Equals | NotEquals | In | LessThanOrEqual | LessThan | GreaterThanOrEqual | GreaterThan | Contains | NotContains | BeginsWith`

Required: Yes

Values

The values that you want to filter for. For example, you might want to display only Amazon S3 locations.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[0-9a-zA-Z_\ \-\:\*\.\\/\?-]*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocationListEntry

Represents a single entry in a list of locations. `LocationListEntry` returns an array that contains a list of locations when the ListLocations operation is called.

Contents

LocationArn

The Amazon Resource Name (ARN) of the location. For Network File System (NFS) or Amazon EFS, the location is the export path. For Amazon S3, the location is the prefix path that you want to mount and use as the root of the location.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:location/loc-[0-9a-z]{17}$`

Required: No

LocationUri

Represents a list of URIs of a location. `LocationUri` returns an array that contains a list of locations when the ListLocations operation is called.

Format: `TYPE://GLOBAL_ID/SUBDIR`.

TYPE designates the type of location (for example, `nfs` or `s3`).

GLOBAL_ID is the globally unique identifier of the resource that backs the location. An example for EFS is `us-east-2.fs-abcd1234`. An example for Amazon S3 is the bucket name, such as `myBucket`. An example for NFS is a valid IPv4 address or a hostname that is compliant with Domain Name Service (DNS).

SUBDIR is a valid file system path, delimited by forward slashes as is the *nix convention. For NFS and Amazon EFS, it's the export path to mount the location. For Amazon S3, it's the prefix path that you mount to and treat as the root of the location.

Type: String

Length Constraints: Maximum length of 4356.

Pattern: `^(efs|nfs|s3|smb|hdfs|fsx[a-z0-9]+)://[a-zA-Z0-9.:/\-]+$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NfsMountOptions

Represents the mount options that are available for DataSync to access an NFS location.

Contents

Version

The specific NFS version that you want DataSync to use to mount your NFS share. If the server refuses to use the version specified, the sync will fail. If you don't specify a version, DataSync defaults to `AUTOMATIC`. That is, DataSync automatically selects a version based on negotiation with the NFS server.

You can specify the following NFS versions:

- **NFSv3** - stateless protocol version that allows for asynchronous writes on the server.
- **NFSv4.0** - stateful, firewall-friendly protocol version that supports delegations and pseudo file systems.
- **NFSv4.1** - stateful protocol version that supports sessions, directory delegations, and parallel data processing. Version 4.1 also includes all features available in version 4.0.

Type: String

Valid Values: `AUTOMATIC | NFS3 | NFS4_0 | NFS4_1`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OnPremConfig

A list of Amazon Resource Names (ARNs) of agents to use for a Network File System (NFS) location.

Contents

AgentArns

ARNs of the agents to use for an NFS location.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]+:[0-9]{12}:agent/agent-[0-9a-z]{17}$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Options

Represents the options that are available to control the behavior of a StartTaskExecution operation. Behavior includes preserving metadata such as user ID (UID), group ID (GID), and file permissions, and also overwriting files in the destination, data integrity verification, and so on.

A task has a set of default options associated with it. If you don't specify an option in StartTaskExecution, the default value is used. You can override the defaults options on each task execution by specifying an overriding `Options` value to StartTaskExecution.

Contents

Atime

A file metadata value that shows the last time a file was accessed (that is, when the file was read or written to). If you set `Atime` to BEST_EFFORT, DataSync attempts to preserve the original `Atime` attribute on all source files (that is, the version before the PREPARING phase). However, `Atime`'s behavior is not fully standard across platforms, so AWS DataSync can only do this on a best-effort basis.

Default value: BEST_EFFORT.

BEST_EFFORT: Attempt to preserve the per-file `Atime` value (recommended).

NONE: Ignore `Atime`.

> **Note**
> If `Atime` is set to BEST_EFFORT, `Mtime` must be set to PRESERVE.
> If `Atime` is set to NONE, `Mtime` must also be NONE.

Type: String

Valid Values: `NONE | BEST_EFFORT`

Required: No

BytesPerSecond

A value that limits the bandwidth used by AWS DataSync. For example, if you want AWS DataSync to use a maximum of 1 MB, set this value to `1048576` (`=1024*1024`).

Type: Long

Valid Range: Minimum value of -1.

Required: No

Gid

The POSIX group ID (GID) of the file's owners.

You can set this option for the following location types:

- Network File System (NFS)
- Hadoop Distributed File System (HDFS)
- Amazon EFS
- Amazon S3
- Amazon FSx for Lustre
- Amazon FSx for OpenZFS

For more information, see Metadata copied by DataSync.

Default value: INT_VALUE. This preserves the integer value of the ID.

INT_VALUE: Preserve the integer value of user ID (UID) and GID (recommended).

NONE: Ignore UID and GID.

Type: String

Valid Values: `NONE | INT_VALUE | NAME | BOTH`

Required: No

LogLevel

A value that determines the type of logs that DataSync publishes to a log stream in the Amazon CloudWatch log group that you provide. For more information about providing a log group for DataSync, see CloudWatchLogGroupArn. If set to `OFF`, no logs are published. `BASIC` publishes logs on errors for individual files transferred, and `TRANSFER` publishes logs for every file or object that is transferred and integrity checked.

Type: String

Valid Values: `OFF | BASIC | TRANSFER`

Required: No

Mtime

A value that indicates the last time that a file was modified (that is, a file was written to) before the PREPARING phase. This option is required for cases when you need to run the same task more than one time.

Default Value: `PRESERVE`

PRESERVE: Preserve original `Mtime` (recommended)

NONE: Ignore `Mtime`.

> **Note**
> If `Mtime` is set to PRESERVE, `Atime` must be set to BEST_EFFORT.
> If `Mtime` is set to NONE, `Atime` must also be set to NONE.

Type: String

Valid Values: `NONE | PRESERVE`

Required: No

ObjectTags

Specifies whether object tags are maintained when transferring between object storage systems. If you want your DataSync task to ignore object tags, specify the `NONE` value.

Default Value: `PRESERVE`

Type: String

Valid Values: `PRESERVE | NONE`

Required: No

OverwriteMode

A value that determines whether files at the destination should be overwritten or preserved when copying files. If set to `NEVER` a destination file will not be replaced by a source file, even if the destination file differs from the source file. If you modify files in the destination and you sync the files, you can use this value to protect against overwriting those changes.

Some storage classes have specific behaviors that can affect your S3 storage cost. For detailed information, see Considerations when working with Amazon S3 storage classes in DataSync in the *AWS DataSync User Guide*.

Type: String

Valid Values: `ALWAYS | NEVER`

Required: No

PosixPermissions

A value that determines which users or groups can access a file for a specific purpose such as reading, writing, or execution of the file.

You can set this option for the following location types:

- Network File System (NFS)
- Hadoop Distributed File System (HDFS)
- Amazon EFS
- Amazon S3
- Amazon FSx for Lustre
- Amazon FSx for OpenZFS

Default value: PRESERVE

PRESERVE: Preserve POSIX-style permissions (recommended).

NONE: Ignore permissions.

> **Note**
> AWS DataSync can preserve extant permissions of a source location.

Type: String

Valid Values: `NONE | PRESERVE`

Required: No

PreserveDeletedFiles

A value that specifies whether files in the destination that don't exist in the source file system should be preserved. This option can affect your storage cost. If your task deletes objects, you might incur minimum storage duration charges for certain storage classes. For detailed information, see Considerations when working with Amazon S3 storage classes in DataSync in the *AWS DataSync User Guide*.

Default value: PRESERVE.

PRESERVE: Ignore such destination files (recommended).

REMOVE: Delete destination files that aren’t present in the source.

Type: String

Valid Values: `PRESERVE | REMOVE`

Required: No

PreserveDevices

A value that determines whether AWS DataSync should preserve the metadata of block and character devices in the source file system, and re-create the files with that device name and metadata on the destination. DataSync does not copy the contents of such devices, only the name and metadata.

> **Note**
> AWS DataSync can't sync the actual contents of such devices, because they are nonterminal and don't return an end-of-file (EOF) marker.

Default value: NONE.

NONE: Ignore special devices (recommended).

PRESERVE: Preserve character and block device metadata. This option isn't currently supported for Amazon EFS.

Type: String

Valid Values: `NONE | PRESERVE`

Required: No

SecurityDescriptorCopyFlags

A value that determines which components of the SMB security descriptor are copied from source to destination objects.

This value is only used for transfers between SMB and Amazon FSx for Windows File Server locations, or between two Amazon FSx for Windows File Server locations. For more information about how DataSync handles metadata, see How DataSync Handles Metadata and Special Files.

Default value: OWNER_DACL.

OWNER_DACL: For each copied object, DataSync copies the following metadata:

- Object owner.
- NTFS discretionary access control lists (DACLs), which determine whether to grant access to an object.

When choosing this option, DataSync does NOT copy the NTFS system access control lists (SACLs), which are used by administrators to log attempts to access a secured object.

OWNER_DACL_SACL: For each copied object, DataSync copies the following metadata:

- Object owner.
- NTFS discretionary access control lists (DACLs), which determine whether to grant access to an object.
- NTFS system access control lists (SACLs), which are used by administrators to log attempts to access a secured object.

Copying SACLs requires granting additional permissions to the Windows user that DataSync uses to access your SMB location. For information about choosing a user that ensures sufficient permissions to files, folders, and metadata, see user.

NONE: None of the SMB security descriptor components are copied. Destination objects are owned by the user that was provided for accessing the destination location. DACLs and SACLs are set based on the destination server’s configuration.

Type: String

Valid Values: `NONE | OWNER_DACL | OWNER_DACL_SACL`

Required: No

TaskQueueing

A value that determines whether tasks should be queued before executing the tasks. If set to `ENABLED`, the tasks will be queued. The default is `ENABLED`.

If you use the same agent to run multiple tasks, you can enable the tasks to run in series. For more information, see Queueing task executions.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

TransferMode

A value that determines whether DataSync transfers only the data and metadata that differ between the source and the destination location, or whether DataSync transfers all the content from the source, without comparing to the destination location.

CHANGED: DataSync copies only data or metadata that is new or different content from the source location to the destination location.

ALL: DataSync copies all source location content to the destination, without comparing to existing content on the destination.

Type: String

Valid Values: `CHANGED | ALL`

Required: No

Uid

The POSIX user ID (UID) of the file's owner.

You can set this option for the following location types:

- Network File System (NFS)
- Hadoop Distributed File System (HDFS)
- Amazon EFS
- Amazon S3
- Amazon FSx for Lustre
- Amazon FSx for OpenZFS

Default value: INT_VALUE. This preserves the integer value of the ID.

INT_VALUE: Preserve the integer value of UID and group ID (GID) (recommended).

NONE: Ignore UID and GID.

Type: String

Valid Values: `NONE | INT_VALUE | NAME | BOTH`

Required: No

VerifyMode

A value that determines whether a data integrity verification should be performed at the end of a task execution after all data and metadata have been transferred. For more information, see Configure task settings.

Default value: POINT_IN_TIME_CONSISTENT.

ONLY_FILES_TRANSFERRED (recommended): Perform verification only on files that were transferred.

POINT_IN_TIME_CONSISTENT: Scan the entire source and entire destination at the end of the transfer to verify that source and destination are fully synchronized. This option isn't supported when transferring to S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes.

NONE: No additional verification is done at the end of the transfer, but all data transmissions are integrity-checked with checksum verification during the transfer.

Type: String

Valid Values: `POINT_IN_TIME_CONSISTENT | ONLY_FILES_TRANSFERRED | NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PrivateLinkConfig

The VPC endpoint, subnet, and security group that an agent uses to access IP addresses in a VPC (Virtual Private Cloud).

Contents

PrivateLinkEndpoint

The private endpoint that is configured for an agent that has access to IP addresses in a PrivateLink. An agent that is configured with this endpoint will not be accessible over the public internet.

Type: String

Length Constraints: Minimum length of 7. Maximum length of 15.

Pattern: `\A(25[0-5]|2[0-4]\d|[0-1]?\d?\d)(\.(25[0-5]|2[0-4]\d|[0-1]?\d?\d)){3}\z`

Required: No

SecurityGroupArns

The Amazon Resource Names (ARNs) of the security groups that are configured for the EC2 resource that hosts an agent activated in a VPC or an agent that has access to a VPC endpoint.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:security-group/.*$`

Required: No

SubnetArns

The Amazon Resource Names (ARNs) of the subnets that are configured for an agent activated in a VPC or an agent that has access to a VPC endpoint.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):ec2:[a-z\-0-9]*:[0-9]{12}:subnet/.*$`

Required: No

VpcEndpointId

The ID of the VPC endpoint that is configured for an agent. An agent that is configured with a VPC endpoint will not be accessible over the public internet.

Type: String

Pattern: `^vpce-[0-9a-f]{17}$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

QopConfiguration

The Quality of Protection (QOP) configuration specifies the Remote Procedure Call (RPC) and data transfer privacy settings configured on the Hadoop Distributed File System (HDFS) cluster.

Contents

DataTransferProtection

The data transfer protection setting configured on the HDFS cluster. This setting corresponds to your `dfs.data.transfer.protection` setting in the `hdfs-site.xml` file on your Hadoop cluster.

Type: String

Valid Values: `DISABLED | AUTHENTICATION | INTEGRITY | PRIVACY`

Required: No

RpcProtection

The RPC protection setting configured on the HDFS cluster. This setting corresponds to your `hadoop.rpc.protection` setting in your `core-site.xml` file on your Hadoop cluster.

Type: String

Valid Values: `DISABLED | AUTHENTICATION | INTEGRITY | PRIVACY`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

S3Config

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role used to access an Amazon S3 bucket.

For detailed information about using such a role, see Creating a Location for Amazon S3 in the *AWS DataSync User Guide*.

Contents

BucketAccessRoleArn

The ARN of the IAM role for accessing the S3 bucket.

Type: String

Length Constraints: Maximum length of 2048.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SmbMountOptions

Represents the mount options that are available for DataSync to access an SMB location.

Contents

Version

The specific SMB version that you want DataSync to use to mount your SMB share. If you don't specify a version, DataSync defaults to `AUTOMATIC`. That is, DataSync automatically selects a version based on negotiation with the SMB server.

Type: String

Valid Values: `AUTOMATIC | SMB2 | SMB3`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TagListEntry

Represents a single entry in a list of AWS resource tags. `TagListEntry` returns an array that contains a list of tasks when the ListTagsForResource operation is called.

Contents

Key

The key for an AWS resource tag.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:/-]+$`

Required: Yes

Value

The value for an AWS resource tag.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskExecutionListEntry

Represents a single entry in a list of task executions. `TaskExecutionListEntry` returns an array that contains a list of specific invocations of a task when the ListTaskExecutions operation is called.

Contents

Status

The status of a task execution.

Type: String

Valid Values: `QUEUED | LAUNCHING | PREPARING | TRANSFERRING | VERIFYING | SUCCESS | ERROR`

Required: No

TaskExecutionArn

The Amazon Resource Name (ARN) of the task that was executed.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}/execution/exec-[0-9a-f]{17}$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskExecutionResultDetail

Describes the detailed result of a `TaskExecution` operation. This result includes the time in milliseconds spent in each phase, the status of the task execution, and the errors encountered.

Contents

ErrorCode

Errors that AWS DataSync encountered during execution of the task. You can use this error code to help troubleshoot issues.

Type: String

Required: No

ErrorDetail

Detailed description of an error that was encountered during the task execution. You can use this information to help troubleshoot issues.

Type: String

Required: No

PrepareDuration

The total time in milliseconds that AWS DataSync spent in the PREPARING phase.

Type: Long

Valid Range: Minimum value of 0.

Required: No

PrepareStatus

The status of the PREPARING phase.

Type: String

Valid Values: `PENDING | SUCCESS | ERROR`

Required: No

TotalDuration

The total time in milliseconds that AWS DataSync took to transfer the file from the source to the destination location.

Type: Long

Valid Range: Minimum value of 0.

Required: No

TransferDuration

The total time in milliseconds that AWS DataSync spent in the TRANSFERRING phase.

Type: Long

Valid Range: Minimum value of 0.

Required: No

TransferStatus

The status of the TRANSFERRING phase.

Type: String

Valid Values: `PENDING | SUCCESS | ERROR`

Required: No

VerifyDuration

The total time in milliseconds that AWS DataSync spent in the VERIFYING phase.

Type: Long

Valid Range: Minimum value of 0.

Required: No

VerifyStatus

The status of the VERIFYING phase.

Type: String

Valid Values: `PENDING | SUCCESS | ERROR`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskFilter

You can use API filters to narrow down the list of resources returned by `ListTasks`. For example, to retrieve all tasks on a source location, you can use `ListTasks` with filter name `LocationId` and `Operator Equals` with the ARN for the location.

Contents

Name

The name of the filter being used. Each API call supports a list of filters that are available for it. For example, `LocationId` for `ListTasks`.

Type: String

Valid Values: `LocationId | CreationTime`

Required: Yes

Operator

The operator that is used to compare filter values (for example, `Equals` or `Contains`). For more about API filtering operators, see API filters for ListTasks and ListLocations.

Type: String

Valid Values: `Equals | NotEquals | In | LessThanOrEqual | LessThan | GreaterThanOrEqual | GreaterThan | Contains | NotContains | BeginsWith`

Required: Yes

Values

The values that you want to filter for. For example, you might want to display only tasks for a specific destination location.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[0-9a-zA-Z_\ \-\:\*\.\\/\?-]*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskListEntry

Represents a single entry in a list of tasks. `TaskListEntry` returns an array that contains a list of tasks when the ListTasks operation is called. A task includes the source and destination file systems to sync and the options to use for the tasks.

Contents

Name

The name of the task.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9\s+=._:@/-]+$`

Required: No

Status

The status of the task.

Type: String

Valid Values: `AVAILABLE | CREATING | QUEUED | RUNNING | UNAVAILABLE`

Required: No

TaskArn

The Amazon Resource Name (ARN) of the task.

Type: String

Length Constraints: Maximum length of 128.

Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):datasync:[a-z\-0-9]*:[0-9]{12}:task/task-[0-9a-f]{17}$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskSchedule

Specifies the schedule you want your task to use for repeated executions. For more information, see Schedule Expressions for Rules.

Contents

ScheduleExpression

A cron expression that specifies when AWS DataSync initiates a scheduled transfer from a source to a destination location.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `^[a-zA-Z0-9\ \_\*\?\,\|\^\-\/\#\s\(\)\+]*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Document history

The following table describes important additions to the AWS DataSync documentation. We also update the documentation frequently to address feedback that you send us.

To get notified about updates to this documentation, subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
New security options for Amazon EFS locations	AWS DataSync can access Amazon EFS file systems using TLS, access points, and IAM roles.	May 31, 2022
Migrating data to or from Google Cloud Storage and Azure Files (p. 338)	With AWS DataSync, you can transfer data to or from Google Cloud Storage and Azure Files. For more information, see Creating a location for object storage and Creating a location for SMB.	May 24, 2022
New AWS DataSync task setting	With the **Copy object tags** option, you can specify whether to maintain object tags when transferring between object storage systems.	May 5, 2022
New AWS Region	AWS DataSync is now available in the Asia Pacific (Jakarta) Region.	April 19, 2022
Support for Amazon FSx for OpenZFS file systems	AWS DataSync can now transfer files and folders to and from FSx for OpenZFS file systems.	April 5, 2022
Support for Amazon FSx for Lustre file systems	AWS DataSync can now transfer files and folders to and from FSx for Lustre file systems.	December 10, 2021
Support for Hadoop Distributed File Systems (HDFS)	AWS DataSync now supports transferring files and folders to and from HDFS clusters.	November 3, 2021
New AWS Region	AWS DataSync is now available in the Asia Pacific (Osaka) Region.	July 28, 2021
Fully automated transfers between AWS storage services	AWS DataSync can now transfer files or objects between Amazon S3, Amazon EFS, or FSx for Windows File Server with just a few clicks in the DataSync console.	November 9, 2020

Adjusting the network bandwidth used by a running task	AWS DataSync now enables customers to adjust the network bandwidth used by a running DataSync task. This helps to minimize impact on other users or applications when a task spans multiple days.	November 9, 2020
Enhanced support for on-premises DataSync virtual machine (VM) functions	The AWS DataSync agent VM host console now supports enhanced functions, including activating an agent from the local console.	October 19, 2020
AWS DataSync can now transfer data to and from AWS Outposts	DataSync now supports transferring objects to and from Amazon S3 on AWS Outposts.	September 30, 2020
Support for API filtering	AWS DataSync now supports filtering for the `ListTasks` and `ListLocations` API calls, enabling you to easily retrieve configuration of data transfer tasks by using filters such as the source or destination for the data transfer.	August 18, 2020
Support for copying data from your self-managed object storage	AWS DataSync now supports data transfer between self-managed object storage and Amazon S3, Amazon Elastic File System, or FSx for Windows File Server.	July 27, 2020
Support for Linux Kernel-based Virtual Machine (KVM) and Microsoft Hyper-V hypervisors	AWS DataSync now provides the ability to deploy on-premises agents on the KVM and Microsoft Hyper-V virtualization platforms, in addition to the existing VMware and Amazon EC2 options.	July 1, 2020
AWS DataSync can now automatically configure your Amazon CloudWatch Logs configuration (p. 338)	When using DataSync, you now have the option of automatically generating the CloudWatch log group and resource policy required to publish logs for your data transfer, simplifying task creation and monitoring setup.	July 1, 2020

AWS DataSync can now transfer data to and from AWS Snowcone (p. 338)	DataSync now supports transferring files to and from AWS Snowcone, the smallest member of the AWS Snow Family of edge computing and data transfer devices. Snowcone is portable, ruggedized, and secure—small and light enough to fit in a backpack and able to withstand harsh environments.	June 17, 2020
New AWS Region	AWS DataSync is now available in the Africa (Cape Town) Region and the Europe (Milan) Region.	June 16, 2020
Enhanced monitoring capabilities with file-level logging	You can now enable detailed logging for files and objects copied between your NFS servers, SMB servers, Amazon S3 buckets, Amazon EFS file systems, and FSx for Windows File Server file systems.	April 24, 2020
Support for copying data between your SMB share and Amazon FSx for Windows File Server	You can now copy data between your SMB share and FSx for Windows File Server.	January 24, 2020
Support for scheduling tasks	You can now run tasks manually or schedule them to run based on a specified schedule.	November 20, 2019
New AWS Region	AWS DataSync is now available in the Asia Pacific (Hong Kong) Region, Asia Pacific (Mumbai) Region, Europe (Stockholm) Region, South America (São Paulo) Region, and AWS GovCloud (US-East) Region.	November 20, 2019
New AWS Region	AWS DataSync is now available in the Canada (Central) Region, Europe (London) Region, and Europe (Paris) Region.	October 2, 2019
Support for Amazon S3 storage classes	You can now transfer objects directly into Amazon S3 storage classes.	September 24, 2019
New AWS Region	AWS DataSync is now available in the Middle East (Bahrain) Region.	August 28, 2019
Support for copying data between your Server Message Block (SMB) share and Amazon S3 or Amazon EFS	You can now copy data between your SMB file share and Amazon S3 or Amazon EFS.	August 22, 2019

Support for using virtual private cloud (VPC) endpoints	You can now create a private connection between your agent and AWS and run tasks in a private network. Doing this increases the security of your data as it's copied over the network.	August 5, 2019
Support for Federal Information Processing Standard (FIPS) endpoints	You can now use FIPS endpoints to create agents and run tasks.	August 5, 2019
New AWS Region	AWS DataSync is now available in the AWS GovCloud (US-West) Region.	June 11, 2019
Support for filtering	You can now apply filters to transfer only a subset of the files in your source location when you transfer data from your source to your destination location.	May 22, 2019
First release of AWS DataSync (p. 338)	General release of the AWS DataSync service.	November 26, 2018

AWS glossary

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