User Guide

AWS Data Transfer Terminal

AWS Data Transfer Terminal: User Guide

Table of Contents

What is Data Transfer Terminal? 1
Features 1
Key concepts 2
Transfer team 2
Personnel 2
Facilities 3
Scheduling considerations 3
Use cases 4
Related services 4
Technical requirements 6
Equipment 6
Network requirements 6
Performance optimization 6
More information 8
Getting started 9
Sign up for an AWS account 9
Create a user with administrative access 10
Schedule a reservation 12
Create a Transfer team 12
Updating Transfer teams on your Data Transfer Terminal account 13
Add personnel 13
Updating personnel on your Data Transfer Terminal account 14
Specify reservation details 15
Review and confirm your reservation 16
Making changes to your reservation 16
Make a data transfer 18
What to bring 18
The Data Transfer Terminal facility physical address 18
Accessing the building 19
Expected equipment in the Data Transfer Terminal suite. 19
Troubleshooting network connections 20
Equipment connection issues 20
Troubleshooting connectivity 20
Linux/Unix 21

Windows ... 22
Network throughput ... 22
Security ... 23
Data protection ... 24
Data encryption ... 25
Encryption in transit ... 25
Key management ... 26
Inter-network traffic privacy ... 26
Identity and access management ... 26
Audience ... 27
Authenticating with identities ... 27
Managing access using policies ... 31
How Data Transfer Terminal works with IAM ... 33
Identity-based policy examples ... 39
Troubleshooting ... 42
API references ... 43
Compliance validation ... 47
Resilience ... 48
CloudTrail logs ... 49
Data Transfer Terminal information in CloudTrail ... 49
Understanding Data Transfer Terminal log file entries ... 50
Infrastructure Security ... 50
Document history ... 52

What is Data Transfer Terminal?

AWS Data Transfer Terminal is a network-ready, physical location you can bring your data storage devices for fast data transfer to and from your AWS Cloud service. Upload data captured remotely to making it easier to access data captured remotely.

Schedule a reservation at one of our physical Data Transfer Terminal facilities from the AWS Management Console, arrive at your scheduled time, and upload your data to your AWS Cloud services with your own devices. After your scheduled reservation is complete and you leave, the facility is re-secured and made ready for the next scheduled reservation.

> **Note**
>
> AWS Data Transfer Terminal is only available to AWS Enterprise customers at this time.

To access Data Transfer Terminal:

- **AWS Data Transfer Terminal console**: https://console.aws.amazon.com/datatransferterminal
- **Data Transfer Terminal facilities**: The location of Data Transfer Terminal facilities are provided once a reservation is made in the console. For more information, see Make a data transfer.

Features

Using AWS Data Transfer Terminal makes getting your data into your AWS Cloud service easier from remote locations. The following are some of the advantages of Data Transfer Terminal for your remote data upload needs:

Secure, private, and exclusive

Each Data Transfer Terminal facility is a secure, private location for you to make large data transfers between your data storage device and your AWS services over a fast network connection.

A dedicated reservation console

Add approved personnel to your Transfer team and schedule a Data Transfer Terminal reservation using the AWS Data Transfer Terminal console.

Fiber optic network connections

Each Data Transfer Terminal facility includes two 100 Gigabit (Gbps) fiber optic (LR4) connections for fast data uploads and redundancy.

Control of your data storage devices

No need to ship your Snowball device and wait for your data to be uploaded to your AWS Cloud services. You control your physical data storage devices throughout the entire data transfer process, getting your data where it needs to go faster.

Key concepts

Using AWS Data Transfer Terminal requires a Process owner to schedule a reservation for a Data transfer specialist to access a Data Transfer Terminal facility. Refer to the following sections to learn more about Data Transfer Terminal terminology.

Topics

- Transfer team
- Personnel
- Facilities

Transfer team

A Transfer team is a grouping of personnel determined by an AWS account owner that may be selected for conducting data transfers on behalf of your organization. Setting up a Transfer team includes giving the Transfer team a name and specifying personnel for the team. We recommend groups of four or fewer Data transfer specialists for a single reservation.

For more information, see Schedule a Data Transfer Terminal reservation.

Personnel

Personnel refers to the individuals who can either make and manage reservations or can go to and use Data Transfer Terminal facilities. Personnel may be either an Process owner or Data transfer specialist or both.

Process owner

A Process owner is an AWS account owner who can add, edit, and remove personnel from their AWS Data Transfer Terminal account.

Data transfer specialist

A Data transfer specialist is an individual who can go to Data Transfer Terminal facilities for data upload transactions. These personnel must be authorized by the Process owner and added to your AWS Data Transfer Terminal account. When accessing a Data Transfer Terminal facility, a government-issued ID will be required.

Facilities

Data Transfer Terminal facilities are data hubs, co-owned and managed by one or more service providers. Each facility requires Data Transfer Terminal Data transfer specialists to provide a government issued, proof of identity that must match their reservation records to access the Data Transfer Terminal suite.

Scheduling considerations

Reservations can be made in Data Transfer Terminal console for one to six hours in length, for any day of the week, throughout the year. Individual reservations can be scheduled consecutively, with a minimum of one hour separation between reservations. All reservations must be made at least 24 hours in advance.

The amount of time required to make a data transfer varies depending upon upload performance speeds. Consider the following factors that impact upload performance as you plan for and schedule your Data Transfer Terminal reservation.

Equipment

Some equipment may include settings that may impact upload performance. Refer to your equipment specifications for suggested upload performance speeds.

Network conditions

Times of heavy network traffic will impact data upload speeds and should be taken into consideration when selecting a time for your data transfer session. Planning your data transfer session for off-peak hours or during times of less network activity may improve your upload speed.

Data transfer size

The Data Transfer Terminal network connectivity is designed for large data transfers. However, the size of the data being transfered will impact how long the session takes.

Use cases

While any AWS Enterprise customer can access the Data Transfer Terminal system, certain use case scenarios may find greater benefit from it.

Autonomous Driving and Advanced Driver Assistance Systems (AD/ADAS): Automotive Original Equipment Manufacturers (OEM) and suppliers generate large data sets from their fleets of autonomous vehicles operating and collecting data in numerous metros within North America, Europe, and ASEAN. With Data Transfer Terminal, the data collected by these fleet vehicles can be uploaded to the AWS Cloud service and used to train AD/ADAS models.

Media and Entertainment: Studios and other content creators often generate digital video and audio (AV) files in remote locations. It is important these AV files are uploaded to the cloud in a timely manner so that geographically dispersed production and editing teams can begin workflows in parallel and in real-time. By using Data Transfer Terminal to remotely upload data, production timelines can be shortened, translating into reduced production costs.

Maps, Photogrammetry, and 3D imagery: Organizations that work with mapping or imagery applications collect data in remote locations and need to upload these visual files to the AWS Cloud for analysis or training. Data Transfer Terminal minimizes the time between collecting and analyzing these large data sets, which helps keep geospacial data up-to-date for drivers, farmers, and other users of that information.

Related services

The following AWS services provide an optimal experience while using Data Transfer Terminal.

AWS service	Description
AWS Snowball Edge	AWS Data Transfer Terminal complements Snowball products by providing a location for faster upload to your AWS cloud, minimizing wait times to access your data.

AWS service	Description
Amazon S3	Bring your own device to a Data Transfer Terminal to quickly and securely upload your data to your Amazon S3 service.

Technical requirements for using Data Transfer Terminal

Before scheduling a reservation at a Data Transfer Terminal, you'll need to ensure you have the equipment and configurations necessary to connect to the network. Refer to the following guidelines for optimal network connectivity and experience.

Equipment

You must bring portable devices for connectivity including monitors, a keyboard, a mouse, and computer or laptop to the Data Transfer Terminal facility for your scheduled reservation.

Your hardware must be able to work with fiber optic (L4) connections

> **Note**
>
> As a data security best practice, ensure that your data is encrypted and secured on the storage devices you bring to the Data Transfer Terminal, and that you apply data encryption policies while using the Data Transfer Terminal facility. For more information, see Security of AWS Data Transfer Terminal

Network requirements

Ensure your uploading device, server, or appliance (laptop) is prepared to connect to the network and that it supports DHCP. You should have the following for an optimal data upload experience:

- A 100G QSFP28 LR4 (100GBASE-LR4) optical QSFP transceiver, compatible with the NIC and LC connectors for the fiber cable connections provided in the Data Transfer Terminal facility.
- IP address auto-configuration DHCP enabled. DNS servers are automatically assigned by DHCP.
- Up-to-date software and NIC drivers.

Performance optimization

To maximize the throughput while using the AWS Data Transfer Terminal consider the following recommendations.

- **Recommended hardware**:

  - 100 Gbps network interface card
  - 16-core CPU
  - 128 GB RAM
  - multiple NVME SSD drives in a RAID array

- Use the AWS Common Runtime (AWS CRT) library for uploads using the AWS Command Line Interface or AWS SDK.

  Optimize Amazon S3 transfer settings by configuring the parameters below. Set these values under the top level `s3` key in the AWS config file, default location `~/.aws/config`.

  ```
  [default]
  s3 =
      preferred_transfer_client = crt
      target_bandwidth = 100Gb/s
      max_concurrent_requests = 20
      multipart_chunksize = 16MB
  ```

  Note that all Amazon S3 configuration values are indented and nested under the top level `s3` key.

  - Optional: You can set the above values programmatically using the `aws configure set` command. For example, to set the above values for the default profile, you can run the following commands instead:

    ```
    aws configure set default.s3.preferred_transfer_client crt
    aws configure set default.s3.target_bandwidth 100Gb/s
    aws configure set default.s3.max_concurrent_requests 20
    aws configure set default.s3.multipart_chunksize 16MB
    ```

- To programmatically set these values for a profile other than default, provide the `--profile` flag. For example, to set configuration for a profile named `test-profile`, runa command like the example below.

  ```
  aws configure set s3.max_concurrent_requests 20 --profile test-profile
  ```

- Enable BBR (Linux) on the device for better throughput.

  ```
  sysctl -w net.core.default_qdisc=fq
  sysctl -w net.ipv4.tcp_congestion_control=bbr
  ```

More information

For more information about AWS command line Amazon S3 configurations to optimize your network connectivity and performance, refer to the following resources.

- AWS CLI Amazon S3 Configuration in the **AWS CLI Command Reference**
- Use a performant Amazon S3 client: AWS CRT-based client in the **Amazon S3Amazon AppStream SDK for Java**
- How do I optimize performance when I use AWS CLI to upload large files to Amazon S3? in the **AWS Knowledge Center**

Getting started

Get started with making remote data transfers to your AWS Cloud services by making a reservation at one of the Data Transfer Terminal facilities. To begin, you'll need equipment that is supported by the Data Transfer Terminal facility and an AWS Enterprise account.

Review the Technical requirements for using Data Transfer Terminal section of this guide before scheduling a Data Transfer Terminal reservation to ensure you have equipment with the optimal configurations for the data transfer. Not all data storage devices and network connection equipment is compatible with the fiber optic network connections available in the suites.

When you sign up for AWS, your AWS account is automatically signed up for all services in AWS, including Data Transfer Terminal. You are charged only for the services that you use.

To set up Data Transfer Terminal, use the steps in the following sections.

When you sign up for AWS and set up Data Transfer Terminal, you can optionally change the display language in the AWS Management Console. For more information, see Changing the language of the AWS Management Console in the *AWS Management Console Getting Started Guide*.

Once you have an AWS account you can access Data Transfer Terminal. For more information about setting up and using AWS Data Transfer Terminal, see Schedule a Data Transfer Terminal reservation.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

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 or text message and entering a verification code on the phone keypad.

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

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Schedule a Data Transfer Terminal reservation

To get started using AWS Data Transfer Terminal, you need to have an AWS account and logged into your Data Transfer Terminal console at https://console.aws.amazon.com/datatransferterminal. Once you're logged into your Data Transfer Terminal console, you can see existing reservations or make a new one. To schedule a reservation, you'll need to do the following:

1. Create a Transfer team. You will need to create a designated group of users to create a reservation and access the Data Transfer Terminal facility to make a data transfer. To learn more about this topic, see Create a Transfer team.
2. Once your team is set up, you will need to add personnel to it. To learn more about adding personnel to your Transfer team, see Add personnel.
3. The Process owner can schedule the data transfer with teams on the account. For more information about how to schedule the reservation, see Specify reservation details.
4. Ensure that the details of the reservation are correct before submitting your request. Once submitted, a reservation request cannot be modified for at least 24 hours. For more information, see Review and confirm your reservation.

Once your reservation is processed and confirmed, your Transfer team will be able to access the Data Transfer Terminal facility at the scheduled time. For more information, see Make a data transfer at the Data Transfer Terminal facility.

Create a Transfer team

To access a Data Transfer Terminal facility you'll need to schedule a reservation in the AWS Management Console. Log into your AWS account to access the Data Transfer Terminal console and complete the following steps to schedule your reservation.

1. From the Data Transfer Terminal home page, select the **Get started** button.
2. If you don't already have a Transfer team set up in your account, the **Create reservation** button will be disabled. You will need to create and name a Transfer team to begin.
   a. Select the **Create Transfer team** button.
   b. Give the team a name.

  - The name must be between two and 64 characters long, starting with a letter or number.
  - Only use letters, numbers, periods, and dashes. Special characters are not recognized.
  - Do not include any sensitive identifying information.

c. Create a Transfer team description.

  - Provide a description that helps identify the team, such as describing the purpose of the team for a specific time period, campaign, or project.

d. Select the **Create Transfer team** button.

   You'll be returned to the Transfer team page and your newly created team will appear under the **Transfer teams** section.

Updating Transfer teams on your Data Transfer Terminal account

To set up a new Transfer team, refer to the Schedule a Data Transfer Terminal reservation section of this guide.

To modify or remove a Transfer team, do the following:

1. On the **Transfer teams** page, select the Transfer team you would like to modify.
2. To modify the Transfer team name and description, select the **Edit** button.
3. To add or remove personnel, select the **personnel** tab and complete the steps described in the *How do I modify, add, or remove personnel from my account?* section of this FAQ.
4. To add or cancel a reservation for the selected Transfer team, refer to the Updating personnel on your Data Transfer Terminal account section of this FAQ.

Add personnel

Add Process owners and Data transfer specialists to your Transfer team to set up the data transfer and access the Data Transfer Terminal facility. To add personnel to your Transfer team, do the following:

1. On the **Transfer teams** page, select the desired Transfer team card from those listed in the **Transfer teams** section. The Transfer team's summary page will appear.

2. Choose the **Personnel** tab, and then **Register person** button to add personnel to the Transfer team.
3. Complete the fields with the necessary information about the person you're adding to the Transfer team on the **Register personnel** page.
    a. **Personnel alias**: Create a unique alias to identify the person.
        - The alias is used for identifying personnel while protecting their identity.
        - It can be up to 64 characters long and include letters, numbers, and dashes.
        - Special characters are not permitted.

    b. **First name**: Provide the person's first name as it appears on their government issued identification.

    c. **Last name**: Provide the person's last name or surname s it appears on their government issued identification.

    d. **Email address**: Include a good email address for the person to receive reservation information and instructions for accessing the Data Transfer Terminal facility.
4. Select the **Register person** button to finish adding the person to your Transfer team.

Updating personnel on your Data Transfer Terminal account

Modifying existing personnel on your account in the Data Transfer Terminal console is not currently supported. AWS Data Transfer Terminal Process owners are only able to add or delete personnel at this time.

To remove personnel from your Data Transfer Terminal account, do the following:

1. On the **Transfer teams** page, select the Transfer team associated with the personnel you would like to remove.
2. On the selected Transfer team's summary page, select the **personnel** tab.
3. Click the radio button next to the alias you would like to remove. Note that you will only be able to see the person's alias when deleting their profile.
4. Select **Delete** button. A warning will appear to confirm the intended action for the selected personnel. Click the **Delete** button to continue. A banner will appear at the top of the console confirming the personnel was deleted successfully.

Specify reservation details

The following instructions walk you through how to schedule your Data Transfer Terminal reservation in the AWS Management Console. For information about using the Data Transfer Terminal facility, see Make a data transfer.

1. Select the **Make reservation** button in the **Upcoming reservations** tab.
2. Complete the fields on the **Specify reservation details** page.

   a. **Transfer team selection**: The Transfer team selected as the default appears first. If you would like to choose a different team, click the drop-down arrow to select from the list of Transfer teams available.

   b. **Process owner**: Select the personnel alias you would like to be responsible for managing the reservation.

      - Only one Process owner is allowed for a reservation and they need to be an authorized personnel on your AWS account.

        The Process owner can be included as one of the Data transfer specialists to perform the data transfer activity as well.

   c. **Data transfer specialist**: Select the personnel you want to have access to the Data Transfer Terminal facility to complete the data transfer activity. You may select more than one personnel, as needed.

      - Best practice is to limit your Transfer team to no more than four (4) Data transfer specialists.

   d. **Data Transfer Terminal information**: Specify the Data Transfer Terminal facility, the desired date, and specific time for the data transfer session.

      i. **Data Transfer Terminal facility**: Click the drop-down arrow to select a Data Transfer Terminal facility.

         > **Note**
         >
         > Only facility descriptions will be provided while making a reservation. Additional location information will be provided in the reservation confirmation email.

ii. **Data Transfer Terminal date and time**: Click into the **Search a date and time for your reservation** field to view the calendar and schedule your reservation.

- Reservations must be made a minimum of 24 hours in advance and no more than six (6) months out and can only be a maximum of six (6) hours long. A single reservation may span more than one day to account for overnight scenarios, if necessary.
- Time is indicated using a 24-hour clock and can only be reserved in whole hour increments.
- To make consecutive reservations, you must create separate reservations with at least one hour between each data transfer session.
- For more information, see Scheduling considerations.

3. Confirm the reservation details are correct and then select the **Create** button to continue. This will take you to the confirmation page, which provides a summary of your reservation.

Review and confirm your reservation

After specifying the details of your reservation, select the **Next** button to continue to see the overview page. Review the details of your Data Transfer Terminal reservation request on the **Review and create** page.

- If you are satisfied with the request, select the **Create** button.
- If you need to change your reservation, select the **Previous** button.

Once the reservation request is submitted, the Process owner will receive an email confirming that the request has been received and is being processed. Once the request is approved, another email will confirm the reservation and provide instructions for locating and accessing the Data Transfer Terminal facility. For information about accessing the Data Transfer Terminal facility, see Make a data transfer.

Making changes to your reservation

There is a 24-hour processing period before any changes can be made to your Data Transfer Terminal reservation request.

After the processing period, to view, edit, or delete your reservation, navigate to the Transfer teams page in the console.

1. Locate and select the desired reservation on the team's card.
2. Click the **Actions** menu and select the desired action.
   - **View**: Selecting the view option allows you to view the details of your reservation including the date, time, location, and assigned personnel.
   - **Edit**: You can revise details of the reservation including date, time, location, and assigned personnel. Note that changes must be made 24 hours before the desired reservation date and that the revisions are not immediately accepted and applied. Your Process owner will receive confirmation of the updated request.
   - **Delete**: The delete option allows you to cancel your reservation. The cancellation request must be made a minimum of 24 hours before the scheduled reservation date. The Process owner will receive confirmation of the canceled reservation when the request is approved.

Make a data transfer at the Data Transfer Terminal facility

The Data Transfer Terminal is a secure, co-owned location that provides secure access to the AWS network. To access the Data Transfer Terminal facility, ensure that you have a confirmation email with the location description and access instructions. Refer to the topics below for more information about accessing and using the Data Transfer Terminal facility.

Topics

- What to bring
- The Data Transfer Terminal facility physical address
- Accessing the building
- Expected equipment in the Data Transfer Terminal suite.

What to bring

Data transfer specialists should bring in the items necessary to perform a data transfer, such as a laptop computer, flash drives, Solid State Drives (SSDs), and AWS Snowball Edge. Ensure that your equipment is optimized to use the fiber network cables at the Data Transfer Terminal facility. For more information about optimal equipment and configurations, see Technical requirements for using Data Transfer Terminal.

You are responsible for the installation, use, and removal of the equipment and items you and accompanying Data transfer specialists bring into the Data Transfer Terminal facility. Anything brought into the suite must be removed when leaving. AWS Data Transfer Terminal is not responsible for forgotten or lost items.

The Data Transfer Terminal facility physical address

The physical address for the Data Transfer Terminal facility will not be provided. Instead, the Process owner and Data transfer specialists specified in the reservation will receive an email with the searchable public name of the Data Transfer Terminal facility. AWS Data Transfer Terminal uses the same location identification system as AWS Direct Connect so you can search for the public name on the internet to locate the Data Transfer Terminal facility. If you do not have an email with this information, confirm with your AWS Data Transfer Terminal account manager that you are included in the Transfer team and that your email information is correct.

Accessing the building

To access the Data Transfer Terminal facility, each Data transfer specialist must provide proof of identity or a government-issued ID. Once admitted to the building, security will escort you to your Data Transfer Terminal suite.

Expected equipment in the Data Transfer Terminal suite.

Each Data Transfer Terminal facility should only have two (2) fiber optic cables, a table or desk, and chairs. If there is any other equipment or items in the room, report it to Support immediately.

Troubleshooting network connection issues

If you experience issues connecting to the network while using AWS Data Transfer Terminal, such as not being able to connect the internet or slow connection speeds, consider the following troubleshooting tips.

Topics

- Equipment connection issues
- Troubleshooting connectivity
- Network throughput

Equipment connection issues

If you are having difficulty establishing a physical connection while in the Data Transfer Terminal suite, consider the following:

- Each Data Transfer Terminal facility will have two (2) single-mode, LC fiber cables. If one or both of these cables are missing, contact AWS Support immediately.
- If one fiber optic cable is not working, try rolling the cable first. If you are still unable to connect with the first cable, try using the other cable.

If you're still unable to use the cables to connect, contact AWS Support immediately.

Troubleshooting connectivity

If you're able to connect your equipment but are not able to connect to the network, try the following troubleshooting suggestions.

- Confirm that your equipment configuration meets the specified network requirements. For more information, see Technical requirements for using Data Transfer Terminal
- Switch to the other fiber optic cable to connect.
- Reboot your device while keeping the fiber optic cables connected.
- Perform basic network diagnostics on the device to ensure the following:
  - DHCP is enabled

  - An IP address is assigned to the connected network interface
  - DNS servers are configured
  - The system clock is synchronized with NTP

If you're still unable connect, contact AWS Support and provide them with the following outputs depending on what operating system (OS) is running on your device.

Linux/Unix

- Get IP address and routing information in a terminal or command-line interface (CLI). Verify that an IP address is assigned to the network interface, and a default route with a default gateway address is added in the route table.

  ```
  ip address show
  ip route show
  ```

  - Alternatively, if `iproute2` is not installed on the device and `ip` commands are not available, use the following commands:

    ```
    ifconfig
    netstat -rn
    ```

- Collect DNS server information. This should show two IP addresses starting with the `nameserver` keyword.

  ```
  cat /etc/resolv.conf
  ```

- Collect the output of basic connectivity tests. Replace the `default_gateway_address` with the IP address of the default gateway assigned.

  ```
  ping -c 5 <default_gateway_address>
  ping -c 5 s3.amazonaws.com
  traceroute s3.amazonaws.com
  ```

- Collect the output of the HTTPS connectivity test. The following command should show a HTTP `200 OK` response from Amazon S3.

  ```
  curl -i https://s3.amazonaws.com/ping
  ```

Windows

- Get the IP address, routing and DNS server information in the command prompt. Verify that an IP address is assigned to the network interface, two DNS servers assigned, and a default route with a default gateway address is added in the route table.

  ```
  ipconfig /all
  route print
  ```

- Collect the output of the basic connectivity tests in the command prompt. Replace the `default_gateway_address` with the IP address of the assigned default gateway.

  ```
  ping <default_gateway_address>
  ping s3.amazonaws.com
  tracert s3.amazonaws.com
  ```

- Collect the output of the HTTPS connectivity test in PowerShell. The following command should show a `HTTP 200 OK` response.

  ```
  Invoke-WebRequest -Uri "https://s3.amazonaws.com/ping"
  ```

Network throughput

Network throughput, which measures the actual data transfer rate in a network, can be influenced by various factors. The following may impact your data transfer speeds:

- **Hardware**: The hardware components of the device may cause reduced connection speeds when uploading data. The CPU and disks used in the device could be reaching their performance limits. Consider using NVME SSDs in a RAID array. Make sure you use the AWS CRT library for better performance and to lower CPU usage.
- **Encryption overhead**: Secure transmissions, such as HTTPS, increase processing time due to encryption overhead.
- **Latency**: Latency refers to the time taken for a data packet to travel from source to destination. High latency can be observed when uploading to an Amazon S3 bucket in a different geographic region, which can lead to delays in data transfer and lower throughput. Best practice is to make data transfers within the same region, whenever possible.
- **Packet loss**: Lost packets require retransmission, slowing the data transfer.

Security of AWS Data Transfer Terminal

AWS Data Transfer Terminal provides a secure environment for making data transfers to and from the AWS Cloud. Like any other physical network fiber connection, the Data Transfer Terminal connection does not provide default encryption. Therefore, you will be responsible for enforcing data encryption best practices to ensure your data transfer is secure.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are 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 Data Transfer Terminal, 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 Data Transfer Terminal. The following topics show you how to secure your data while using the Data Transfer Terminal service. You also learn how to use other AWS services that help you to monitor and secure your Data Transfer Terminal resources.

Topics

- Data protection in AWS Data Transfer Terminal
- Identity and access management for Data Transfer Terminal
- Compliance validation for AWS Data Transfer Terminal
- Resilience in AWS Data Transfer Terminal
- Logging and monitoring in Data Transfer Terminal
- Infrastructure Security in AWS Data Transfer Terminal

Data protection in AWS Data Transfer Terminal

The AWS shared responsibility model applies to data protection in AWS Data Transfer Terminal. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Data Transfer Terminal or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

AWS Data Transfer Terminal provides access to a high-speed network connection for you to securely transfer data between self-managed storage systems and AWS storage services. How your storage data is encrypted in transit depends in part on the policies enabled on your devices and the services your data is transfered to. Management of data and it's encryption in transit are the responsibility of the individual using Data Transfer Terminal.

Encryption at rest

AWS Data Transfer Terminal encrypts all data at rest.

Data Transfer Terminal only captures data necessary for reservations including the first and last names and email addresses of the individuals specified to both attend and schedule the reservation. The purpose for this data collection is to confirm reservation details and ensure access to the room to perform the data transfer. This transactional information is backed up no more than 35 days, however, AWS account information is retained for 10 years.

Encryption in transit

AWS Data Transfer Terminal does not encrypt data in transit. Data is encrypted-in-transit when you interact with Data Transfer Terminal API endpoints for setting up Transfer teams, adding personnel, and scheduling reservations in the console. As part of the AWS shared responsibility model, you have choices about how you connect to AWS services through Data Transfer Terminal. We strongly recommend you choose to connect to AWS services using strong encryption-in-transit, such as TLS 1.2 and 1.3.

For example, use only encrypted connections over HTTPS (TLS) by using the `aws:SecureTransport` condition in your Amazon S3 bucket policies, as illustrated in the bucket policy below.

```
{
 "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:",
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
    }]
}
```

To learn more about data encryption in transit with other AWS services, such as Amazon S3, see Protecting data with server-side encryption in the Amazon S3 User Guide.

Key management

AWS Data Transfer Terminal does not directly support Customer managed keys. Use the Customer managed key support available for the AWS services you connect to during your Data Transfer Terminal reservation. Learn more about Customer managed keys and how to encrypt your data at rest in the AWS KMS keys section of the AWS Key Management Service Developer Guide.

Inter-network traffic privacy

Access to the Data Transfer Terminal console is through published service APIs. Data Transfer Terminal resources are independent of virtual private cloud (VPC).

Identity and access management for Data Transfer Terminal

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Data Transfer Terminal resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How Data Transfer Terminal works with IAM

- Identity-based policy examples for AWS Data Transfer Terminal
- Troubleshooting AWS Data Transfer Terminal identity and access
- Data Transfer Terminal API references: Actions and resources

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Data Transfer Terminal.

Service user – If you use the Data Transfer Terminal service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Data Transfer Terminal features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Data Transfer Terminal, see Troubleshooting AWS Data Transfer Terminal identity and access.

Service administrator – If you're in charge of Data Transfer Terminal resources at your company, you probably have full access to Data Transfer Terminal. It's your job to determine which Data Transfer Terminal features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Data Transfer Terminal, see How Data Transfer Terminal works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Data Transfer Terminal. To view example Data Transfer Terminal identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Data Transfer Terminal.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one 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 don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – 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 Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **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 Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to

any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Data Transfer Terminal works with IAM

Before you use IAM to manage access to Data Transfer Terminal, learn what IAM features are available to use with Data Transfer Terminal.

IAM feature	Data Transfer Terminal support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes

IAM feature	Data Transfer Terminal support
ACLs	No
ABAC (tags in policies)	No
Temporary credentials	Yes
Principal permissions	No
Service roles	No
Service-linked roles	No

To get a high-level view of how Data Transfer Terminal and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Data Transfer Terminal

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Data Transfer Terminal

To view examples of Data Transfer Terminal identity-based policies, see Identity-based policy examples for AWS Data Transfer Terminal.

Resource-based policies within Data Transfer Terminal

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Data Transfer Terminal

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Data Transfer Terminal actions, see Actions Defined by AWS Data Transfer Terminal in the *Service Authorization Reference*.

Policy actions in Data Transfer Terminal use the following prefix before the action:

```
datatransferterminal
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "datatransferterminal:action1",
      "datatransferterminal:action2"
         ]
```

To view examples of Data Transfer Terminal identity-based policies, see Identity-based policy examples for AWS Data Transfer Terminal.

Policy resources for Data Transfer Terminal

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Data Transfer Terminal resource types and their ARNs, see Resources Defined by AWS Data Transfer Terminal in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Data Transfer Terminal .

To view examples of Data Transfer Terminal identity-based policies, see Identity-based policy examples for AWS Data Transfer Terminal.

Policy condition keys for Data Transfer Terminal

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Condition element (or Condition *block*) lets you specify conditions in which a statement is in effect. The Condition element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple Condition elements in a statement, or multiple keys in a single Condition element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Data Transfer Terminal condition keys, see Condition Keys for AWS Data Transfer Terminal in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Data Transfer Terminal .

To view examples of Data Transfer Terminal identity-based policies, see Identity-based policy examples for AWS Data Transfer Terminal.

ACLs in Data Transfer Terminal

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Data Transfer Terminal

Supports ABAC (tags in policies): No

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Data Transfer Terminal

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Data Transfer Terminal

Supports forward access sessions (FAS): No

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Data Transfer Terminal

Supports service roles: No

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 Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> ⚠ **Warning**
>
> Changing the permissions for a service role might break Data Transfer Terminal functionality. Edit service roles only when Data Transfer Terminal provides guidance to do so.

Service-linked roles for Data Transfer Terminal

Supports service-linked roles: No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Data Transfer Terminal

By default, users and roles don't have permission to create or modify Data Transfer Terminal resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by , including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for AWS Data Transfer Terminal in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the Data Transfer Terminal console
- Allow users to view their own permissions

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Data Transfer Terminal resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Data Transfer Terminal console

To access the AWS Data Transfer Terminal console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Data Transfer Terminal resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Data Transfer Terminal console, also attach the Data Transfer Terminal `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Troubleshooting AWS Data Transfer Terminal identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Data Transfer Terminal and IAM.

Topics

- I am not authorized to perform an action in Data Transfer Terminal
- I want to allow people outside of my AWS account to access my Data Transfer Terminal resources

I am not authorized to perform an action in Data Transfer Terminal

If you're unable to view or schedule reservations in the AWS Data Transfer Terminal console, you may not have the required permissions. Contact your account administrator to configure an IAM identity policy that grants you access and appropriate permissions.

I want to allow people outside of my AWS account to access my Data Transfer Terminal resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Data Transfer Terminal supports these features, see How Data Transfer Terminal works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Data Transfer Terminal API references: Actions and resources

When creating AWS Identity and Access Management (IAM) policies, this page can help you understand the relationship between AWS Data Transfer Terminal 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 Data Transfer Terminal permissions to your policy:

- Specify an action in the `Action` element. The value includes a `datatransferterminal:` prefix and the API operation name. For example, `datatransferterminal:CreateTask`.

- Specify an AWS resource related to the action in the `Resource` element.

You can also use AWS condition keys in your Data Transfer Terminal policies. For a complete list of AWS keys, see Available keys in the *IAM User Guide*.

Data Transfer Terminal API operations and corresponding actions

CreateTransferTeam

Action:`datatransferterminal:CreateTransferTeam`

Resource: None

GetTransferTeam

Action:`datatransferterminal:GetTransferTeam`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

UpdateTransferTeam

Action:`datatransferterminal:UpdateTransferTeam`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

DeleteTransferTeam

Action:`datatransferterminal:DeleteTransferTeam`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

ListTransferTeams

Action:`datatransferterminal:ListTransferTeams`

Resource: None

RegisterPerson

Action:`datatransferterminal:RegisterPerson`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

GetPerson

Action:`datatransferterminal:GetPerson`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/person/$`*`PersonId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

DeregisterPerson

Action:`datatransferterminal:DeregisterPerson`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/person/$`*`PersonId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

ListPersons

Action:`datatransferterminal:ListPersons`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

CreateReservation

Action:`datatransferterminal:CreateReservation`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

Dependent action:`datatransferterminal:GetPerson`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/person/$`*`PersonId`*

Dependent action:`datatransferterminal:GetFacility`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:::facility/$`*`FacilityId`*

GetReservation

Action:`datatransferterminal:GetReservation`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/reservation/$`*`ReservationId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

UpdateReservation

Action:`datatransferterminal:UpdateReservation`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/reservation/$`*`ReservationId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*

Dependent action:`datatransferterminal:GetPerson`

Dependent resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/person/$`*`PersonId`*

DeleteReservation

Action:`datatransferterminal:DeleteReservation`

Resource:`arn:aws::$`*`Partition`*`:datatransferterminal:$`*`Region`*`:$`*`Account`*`:transfer-team/$`*`TransferTeamId`*`/person/$`*`PersonId`*

Dependent action:`datatransferterminal:GetTransferTeam`

Dependent resource:`arn:aws::$Partition:datatransferterminal:$Region:$Account:transfer-team/$TransferTeamId`

ListReservations

Action:`datatransferterminal:ListReservations`

Resource:`arn:aws::$Partition:datatransferterminal:$Region:$Account:transfer-team/$TransferTeamId`

ListFacilities

Action:`datatransferterminal:ListFacilities`

Resource: None

GetFacility

Action:`datatransferterminal:GetFacility`

Resource:`arn:aws::$Partition:datatransferterminal:::facility/$FacilityId`

GetFacilityAvailability

Action:`datatransferterminal:GetFacilityAvailability`

Resource:`arn:aws::$Partition:datatransferterminal:::facility/$FacilityId/availability`

Dependent action:`datatransferterminal:GetFacility`

Dependent resource:`arn:aws::$Partition:datatransferterminal:::facility/$FacilityId/availability`

Compliance validation for AWS Data Transfer Terminal

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. 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 AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config 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. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Data Transfer Terminal

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 zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

AWS Data Transfer Terminal is available at locations around the world. You can connect to any AWS Region that is accessible from the internet.

Logging and monitoring in Data Transfer Terminal

AWS Data Transfer Terminal is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Data Transfer Terminal. CloudTrail captures all API calls for Data Transfer Terminal as events. The calls captured include calls from the Data Transfer Terminal console and code calls to the Data Transfer Terminal API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Data Transfer Terminal. 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 Data Transfer Terminal, 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.

Data Transfer Terminal information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Data Transfer Terminal, 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 Data Transfer Terminal, 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 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 Data Transfer Terminal actions are logged by CloudTrail and are documented in the Data Transfer Terminal API references: Actions and resources section of this guide.

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 the CloudTrail userIdentity element.

Understanding Data Transfer Terminal 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.

Infrastructure Security in AWS Data Transfer Terminal

As a managed service, AWS Data Transfer Terminal is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Data Transfer Terminal 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 DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). 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.

Document history for the Data Transfer Terminal User Guide

The following table describes the important changes in each release of the AWS Data Transfer Terminal User Guide. For notification about updates to this documentation, you can subscribe to the RSS feed.

Change	Description	Date
Initial publication	The original documentation launch date.	December 2024
Update layout	Updates to the document layout and minor verbiage and content edits.	January 2025