Amazon FSx for Windows File Server

Windows User Guide

Amazon FSx for Windows File Server: Windows User Guide

Table of Contents

What is FSx for Windows File Server?

Amazon FSx for Windows File Server provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system. FSx for Windows File Server has the features, performance, and compatibility to easily lift and shift enterprise applications to the AWS Cloud.

Amazon FSx supports a broad set of enterprise Windows workloads with fully managed file storage built on Microsoft Windows Server. Amazon FSx has native support for Windows file system features and for the industry-standard Server Message Block (SMB) protocol to access file storage over a network. Amazon FSx is optimized for enterprise applications in the AWS Cloud, with native Windows compatibility, enterprise performance and features, and consistent sub-millisecond latencies.

With file storage on Amazon FSx, the code, applications, and tools that Windows developers and administrators use today can continue to work unchanged. Windows applications and workloads ideal for Amazon FSx include business applications, home directories, web serving, content management, data analytics, software build setups, and media processing workloads.

As a fully managed service, FSx for Windows File Server eliminates the administrative overhead of setting up and provisioning file servers and storage volumes. Additionally, Amazon FSx keeps Windows software up to date, detects and addresses hardware failures, and performs backups. It also provides rich integration with other AWS services like AWS IAM, AWS Directory Service for Microsoft Active Directory, Amazon WorkSpaces, AWS Key Management Service, and AWS CloudTrail.

FSx for Windows File Server resources: file systems, backups, and file shares

The primary resources in Amazon FSx are *file systems* and *backups*. A file system is where you store and access your files and folders. A file system is made up of one or more Windows file servers and storage volumes. When you create a file system, you specify an amount of storage capacity (in GiB) and an amount of throughput capacity (in MB/s). You can modify these properties as your needs change after you create the file system. For more information, see Managing storage capacity (p. 123) and Managing throughput capacity (p. 133).

FSx for Windows File Server backups are file-system-consistent, highly durable, and incremental. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in Microsoft Windows. Automatic daily backups are turned on by default when you create a file system, and you can also take additional manual backups at any time. For more information, see Working with backups (p. 78).

A Windows file share is a specific folder (and its subfolders) within your file system that you make accessible to your compute instances with SMB. Your file system already comes with a default Windows file share called `\share`. You can create and manage as many other Windows file shares as you want by using the Shared Folders graphical user interface (GUI) tool on Windows. For more information, see Using Microsoft Windows file shares (p. 53).

File shares are accessed using either the file system's DNS name or DNS aliases that you associate with the file system. For more information, see Managing DNS aliases (p. 90).

Accessing file shares

Amazon FSx is accessible from compute instances with the SMB protocol (supporting versions 2.0 to 3.1.1). You can access your shares from all Windows versions starting from Windows Server 2008

and Windows 7, and also from current versions of Linux. You can map your Amazon FSx file shares on Amazon Elastic Compute Cloud (Amazon EC2) instances, and on WorkSpaces instances, Amazon AppStream 2.0 instances, and VMware Cloud on AWS VMs.

You can access your file shares from on-premises compute instances using AWS Direct Connect or AWS VPN. In addition to accessing file shares that are in the same VPC, AWS account, and AWS Region as the file system, you can also access your shares on compute instances that are in a different Amazon VPC, account, or Region. You do so using VPC peering or transit gateways. For more information, see Supported access methods (p. 15).

Security and data protection

Amazon FSx provides multiple levels of security and compliance to help ensure that your data is protected. It automatically encrypts data at rest (for both file systems and backups) using keys that you manage in AWS Key Management Service (AWS KMS). Data in transit is also automatically encrypted using SMB Kerberos session keys. It has been assessed to comply with ISO, PCI-DSS, and SOC certifications, and is HIPAA eligible.

Amazon FSx provides access control at the file and folder level with Windows access control lists (ACLs). It provides access control at the file system level using Amazon Virtual Private Cloud (Amazon VPC) security groups. In addition, it provides access control at the API level using AWS Identity and Access Management (IAM) access policies. Users accessing file systems are authenticated with Microsoft Active Directory. Amazon FSx integrates with AWS CloudTrail to monitor and log your API calls letting you see actions taken by users on your Amazon FSx resources.

Additionally, it protects your data by taking highly durable backups of your file system automatically on a daily basis and allows you to take additional backups at any point. For more information, see Security in Amazon FSx (p. 174).

Availability and durability

FSx for Windows File Server offers file systems with two levels of availability and durability. Single-AZ files ensure high availability within a single Availability Zone (AZ) by automatically detecting and addressing component failures. In addition, Multi-AZ file systems provide high availability and failover support across multiple Availability Zones by provisioning and maintaining a standby file server in a separate Availability Zone within an AWS Region. To learn more about Single-AZ and Multi-AZ file system deployments, see Availability and durability: Single-AZ and Multi-AZ file systems (p. 20).

Managing file systems

You can administer your FSx for Windows File Server file systems using custom remote management PowerShell commands, or using the Windows-native GUI in some cases. To learn more about managing Amazon FSx file systems, see Administering file systems (p. 88).

Price and performance flexibility

FSx for Windows File Server gives you the price and performance flexibility by offering both solid state drive (SSD) and hard disk drive (HDD) storage types. HDD storage is designed for a broad spectrum of workloads, including home directories, user and departmental shares, and content management

systems. SSD storage is designed for the highest-performance and most latency-sensitive workloads, including databases, media processing workloads, and data analytics applications.

With FSx for Windows File Server, you can provision file system storage and throughput independently to achieve the right mix of cost and performance. You can modify your file system's storage and throughput capacities to meet changing workload needs, so that you pay only for what you need. For more information, see Optimizing costs with Amazon FSx (p. 23).

Pricing for Amazon FSx

With Amazon FSx, there are no upfront hardware or software costs. You pay for only the resources used, with no minimum commitments, setup costs, or additional fees. For information about the pricing and fees associated with the service, see Amazon FSx for Windows File Server Pricing.

Assumptions

To use Amazon FSx, you need an AWS account with an Amazon EC2 instance, WorkSpaces instance, AppStream 2.0 instance, or VM running in VMware Cloud on AWS environments of the supported type.

In this guide, we make the following assumptions:

- If you're using Amazon EC2, we assume that you're familiar with Amazon EC2. For more information on how to use Amazon EC2, see Amazon Elastic Compute Cloud documentation.
- If you're using WorkSpaces, we assume that you're familiar with WorkSpaces. For more information on how to use WorkSpaces, see Amazon WorkSpaces User Guide.
- If you're using VMware Cloud on AWS, we assume that you're familiar with it. For more information, see VMware Cloud on AWS.
- We assume that you are familiar with Microsoft Active Directory concepts.

Prerequisites

To create an Amazon FSx file system, you need the following:

- An AWS account with the permissions necessary to create an Amazon FSx file system and an Amazon EC2 instance. For more information, see Setting up (p. 5).
- An Amazon EC2 instance running Microsoft Windows Server in the virtual private cloud (VPC) based on the Amazon VPC service that you want to associate with your Amazon FSx file system. For information on how to create one, see Getting Started with Amazon EC2 Windows Instances in the *Amazon EC2 User Guide for Windows Instances.*
- Amazon FSx works with Microsoft Active Directory to perform user authentication and access control. You join your Amazon FSx file system to a Microsoft Active Directory while creating it. For more information, see Working with Microsoft Active Directory in FSx for Windows File Server (p. 25).
- This guide assumes that you haven't changed the rules on the default security group for your VPC based on the Amazon VPC service. If you have, you need to ensure that you add the necessary rules to allow network traffic from your Amazon EC2 instance to your Amazon FSx file system. For more details, see Security in Amazon FSx (p. 174).
- Install and configure the AWS Command Line Interface (AWS CLI). Supported versions are 1.9.12 and newer. For more information, see Installing, updating, and uninstalling the AWS CLI in the *AWS Command Line Interface User Guide.*

Note
You can check the version of the AWS CLI you're using with the `aws --version` command.

Amazon FSx for Windows File Server forums

If you encounter issues while using Amazon FSx, use the forums.

Are you a first-time user of Amazon FSx?

If you are a first-time user of Amazon FSx, we recommend that you read the following sections in order:

1. If you're ready to create your first Amazon FSx file system, try the Getting started with Amazon FSx (p. 7).
2. For information about performance, see FSx for Windows File Server performance (p. 153).
3. For Amazon FSx security details, see Security in Amazon FSx (p. 174).
4. For information about the Amazon FSx API, see Amazon FSx API Reference.

Setting up

Before you use Amazon FSx for the first time, complete the following tasks:

1. Sign up for AWS (p. 5)
2. Create an IAM user (p. 5)

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon FSx.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

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

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

Note your AWS account number, because you need it for the next task.

Create an IAM user

Services in AWS, such as Amazon FSx, require that you provide credentials when you access them, so that the service can determine whether you have permissions to access its resources. AWS recommends that you don't use the root credentials of your AWS account to make requests. Instead, create an AWS Identity and Access Management (IAM) user and grant that user full access. We call these users administrator users.

You can use the administrator user credentials, instead of root credentials of your account, to interact with AWS and perform tasks, such as create users and grant them permissions. For more information, see Root Account Credentials vs. IAM User Credentials in the *AWS General Reference* and IAM Best Practices in the *IAM User Guide*.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM Management Console.

To create an administrator user for yourself and add the user to an administrators group (console)

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

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add users**.

3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note
    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

To sign in as this new IAM user, first sign out of the AWS Management Console. Then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`).

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays ***your_user_name@your_aws_account_id***.

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. To do so, from the IAM dashboard, choose **Create Account Alias** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL.

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **AWS Account Alias** on the dashboard.

Next step

Getting started with Amazon FSx (p. 7)

Getting started with Amazon FSx

Following, you can learn how to get started using Amazon FSx. This getting started exercise includes the following steps.

Topics

Step 1: Create your file system

To create your Amazon FSx file system, you must create your Amazon Elastic Compute Cloud (Amazon EC2) instance and the AWS Directory Service directory. If you don't have that set up already, see Walkthrough 1: Prerequisites for getting started (p. 157).

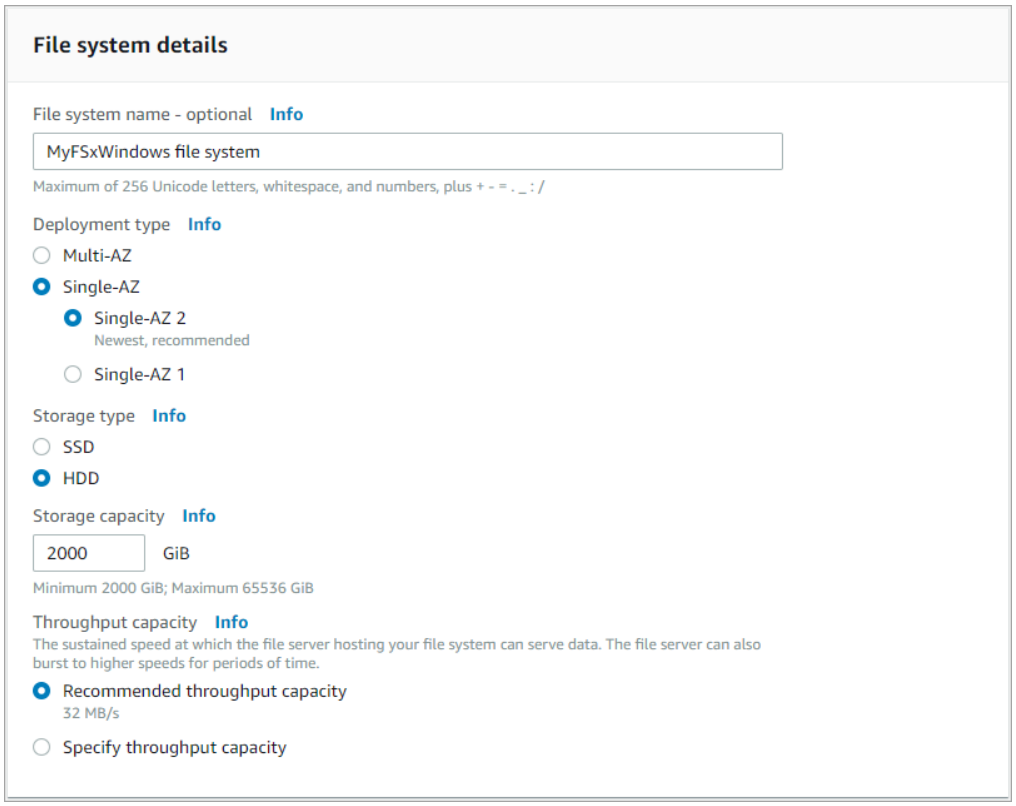
To create your first file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the dashboard, choose **Create file system** to start the file system creation wizard.
3. On the **Select file system type** page, choose **FSx for Windows File Server**, and then choose **Next**. The **Create file system** page appears.
4. In the **File system details** section, provide a name for your file system. It's easier to find and manage your file systems when you name them. You can use a maximum of 256 Unicode letters, white space, and numbers, plus the special characters + - = . _ : /
5. For **Deployment type** choose **Multi-AZ** or **Single-AZ**.

   - Choose **Multi-AZ** to deploy a file system that is tolerant to Availability Zone unavailability. This option supports SSD and HDD storage.
   - Choose **Single-AZ** to deploy a file system that is deployed in a single Availability Zone. *Single-AZ 2* is the latest generation of single Availability Zone file systems, and it supports SSD and HDD storage.

   For more information, see Availability and durability: Single-AZ and Multi-AZ file systems (p. 20).

   The following image shows all of the configuration options available in the **File system details** section.



6. For **Storage type**, you can choose either **SSD** or **HDD**.

   FSx for Windows File Server offers solid state drive (SSD) and hard disk drive (HDD) storage types. **SSD** storage is designed for the highest-performance and most latency-sensitive workloads, including databases, media processing workloads, and data analytics applications. **HDD** storage is designed for a broad spectrum of workloads, including home directories, user and departmental file shares, and content management systems. For more information, see Optimizing costs using storage types (p. 23).

7. For **Storage capacity**, enter the storage capacity of your file system, in GiB. If you're using SSD storage, enter any whole number in the range of 32–65,536. If you're using HDD storage, enter any whole number in the range of 2,000–65,536. You can increase the amount of storage capacity as needed at any time after you create the file system. For more information, see Managing storage capacity (p. 123).

8. Keep **Throughput capacity** at its default setting. **Throughput capacity** is the sustained speed at which the file server that hosts your file system can serve data. The **Recommended throughput capacity** setting is based on the amount of storage capacity you choose. If you need more than the recommended throughput capacity, choose **Specify throughput capacity**, and then choose a value. For more information, see FSx for Windows File Server performance (p. 153).

   > **Note**
   > If you are going to enable file access auditing, you must choose a throughput capacity of 32 MB/s or greater. For more information, see File access auditing (p. 98).

   You can modify the throughput capacity as needed at any time after you create the file system. For more information, see Managing throughput capacity (p. 133).

9. In the **Network & security** section, choose the Amazon VPC that you want to associate with your file system. For this getting started exercise, choose the same Amazon VPC that you chose for your AWS Directory Service directory and your Amazon EC2 instance.

10.

For **VPC Security Groups**, the default security group for your default Amazon VPC is already added to your file system in the console. If you're not using the default security group, make sure that you add the following rules to the security group that you use for this exercise:

a. Add the following inbound and outbound rules to allow the following ports.

Rules	Ports
UDP	53, 88, 123, 389, 464
TCP	53, 88, 135, 389, 445, 464, 636, 3268, 3269, 5985, 9389, 49152-65535

Add from and to IP addresses or security group IDs associated with the client compute instances that you want to access your file system from.

b. Add outbound rules to allow all traffic to the Active Directory that you're joining your file system to. To do this, do one of the following:

- Allow outbound traffic to the security group ID associated with your AWS Managed AD directory.
- Allow outbound traffic to the IP addresses associated with your self-managed Active Directory domain controllers.

Note
In some cases, you might have modified the rules of your AWS Managed Microsoft AD security group from the default settings. If so, make sure that this security group has the required inbound rules to allow traffic from your Amazon FSx file system. For more information about the required inbound rules, see AWS Managed Microsoft AD Prerequisites in the *AWS Directory Service Administration Guide*.

For more information, see File System Access Control with Amazon VPC (p. 177).

11. If you have a Multi-AZ deployment (see step 5), choose a **Preferred subnet** value for the primary file server and a **Standby subnet** value for the standby file server. A Multi-AZ deployment has a primary and a standby file server, each in its own Availability Zone and subnet.
12. For **Windows authentication**, you have the following options:

    If you want to join your file system to a Microsoft Active Directory domain that is managed by AWS, choose **AWS Managed Microsoft Active Directory**, and then choose your AWS Directory Service directory from the list. For more information, see Working with Microsoft Active Directory in FSx for Windows File Server (p. 25).

    If you want to join your file system to a self-managed Microsoft Active Directory domain, choose **Self-managed Microsoft Active Directory**, and provide the following details for your Active Directory.

    - The fully qualified domain name of your Active Directory.

      Important
      For Single-AZ 2 and all Multi-AZ file systems, the Active Directory domain name cannot exceed 47 characters. This limitation applies to both AWS managed and self-managed Active Directory domain names.
      Amazon FSx requires a direct connection or internal traffic to your DNS IP Address. Connection via an internet gateway is not supported. Instead, use a VPN, VPC peering, Direct Connect or a transit gateway association.

- **DNS server IP addresses**—the IPv4 addresses of the DNS servers for your domain

  Note
  Your DNS server must have EDNS (Extension Mechanisms for DNS) enabled. If EDNS is disabled, you may not be able to create an Amazon FSx file system.
- **Service account username**—the user name of the service account in your existing Active Directory. Do not include a domain prefix or suffix.
- **Service account password**—the password for the service account.
- **Confirm password**—the password for the service account.
- (Optional) **Organizational Unit (OU)**—the distinguished path name of the organizational unit in which you want to join your file system.
- (Optional) **Delegated file system administrators group**— the name of the group in your Active Directory that can administer your file system. The default group is 'Domain Admins'.

13. For **Encryption**, keep the default **Encryption key** setting of **aws/fsx (default)**.
14. For **Auditing - optional**, file access auditing is disabled by default. For information about enabling and configuring file access auditing, see To enable file access auditing when creating a file system (console) (p. 101).
15. For **Access - optional**, enter any DNS aliases that you want to associate with the file system. Each alias name must be formatted as a fully qualified domain name (FQDN). For more information, see Managing DNS aliases (p. 90).
16. For **Backup and maintenance - optional**, keep the default settings.
17. For **Tags - optional**, enter a key and value to add tags to your file system. A tag is a case-sensitive key-value pair that helps you manage, filter, and search for your file system.

    Choose **Next**.
18. Review the file system configuration shown on the **Create file system** page. For your reference, note which file system settings you can modify after file system is created. Choose **Create file system**.
19. After Amazon FSx creates the file system, choose the file system ID in the **File Systems** dashboard. Choose **Attach**, and note the fully qualified domain name for your file system. You will need it in a later step.

Step 2: Map your file share to an EC2 instance running Windows Server

You can now mount your Amazon FSx file system to your Microsoft Windows–based Amazon EC2 instance joined to your AWS Directory Service directory. The name of your file share is not the same as the name of your file system.

To map a file share on an Amazon EC2 Windows instance using the GUI

1. Before you can mount a file share on a Windows instance, you must launch the EC2 instance and join it to an AWS Directory Service for Microsoft Active Directory. To perform this action, choose one of the following procedures from the *AWS Directory Service Administration Guide*:

   - Seamlessly Join a Windows EC2 Instance
   - Manually Join a Windows Instance
2. Connect to your instance. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. When you're connected, open File Explorer.
4. From the navigation pane, open the context (right-click) menu for **Network** and choose **Map Network Drive**.

5. Choose a drive letter of your choice for **Drive**.
6. You can map your file system using either its default DNS name assigned by Amazon FSx, or using a DNS alias of your choosing. This procedure describes mapping a file share using the default DNS name. If you want to map a file share using a DNS alias, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

   For **Folder**, enter the file system DNS name and the share name. The default Amazon FSx share is called `\share`. You can find the DNS name in the Amazon FSx console, https://console.aws.amazon.com/fsx/, **Windows File Server > Network & Security** section, or in the response of **CreateFileSystem** or **DescribeFileSystems** API command.

   - For a Single-AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following.

     ```
     fs-0123456789abcdef0.ad-domain.com
     ```

   - For a Single-AZ file system joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following.

     ```
     amznfsxaa11bb22.ad-domain.com
     ```

   For example, enter `\\fs-0123456789abcdef0.ad-domain.com\share`.
7. Choose whether the file share should **Reconnect at sign-in**, and then choose **Finish**.

Step 3: Write data to your file share

Now that you've mapped your file share to your instance, you can use your file share like any other directory in your Windows environment.

To write data to your file share

1. Open the Notepad text editor.
2. Write some content in the text editor. For example: *Hello, World!*
3. Save the file to your file share's drive letter.
4. Using File Explorer, navigate to your file share and find the text file that you just saved.

Step 4: Back up your file system

Now that you've had a chance to use your Amazon FSx file system and its file shares, you can back it up. By default, daily backups are created automatically during your file system's 30-minute backup window. However you can create a user-initiated backup at any time. Backups have additional costs associated with them. For more information on backup pricing, see Pricing.

To create a backup of your file system from the console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose the name of the file system you created for this exercise.
3. From the **Overview** tab for your file system, choose **Create backup**.
4. In the **Create backup** dialog box that opens, provide a name for your backup. This name can contain a maximum of 256 Unicode letters and include white space, numbers, and the following special characters: + - = . _ : /

5. Choose **Create backup**.
6. To view all your backups in a list, so you can restore your file system or delete the backup, choose **Backups**.

When you create a new backup, its status is set to **CREATING** while it is being created. This can take a few minutes. When the backup is available for use, its status changes to **AVAILABLE**.

Step 5: Transfer Files to or from Amazon FSx for Windows File Server Using AWS DataSync

Now that you have a functioning setup for Amazon FSx for Windows File Server, you can use AWS DataSync to transfer files between an existing file system and Amazon FSx for Windows File Server.

AWS DataSync is a data transfer service that simplifies, automates, and accelerates moving and replicating data between on-premises storage systems and AWS storage services over the internet or AWS Direct Connect. DataSync can transfer your file data, and also file system metadata such as ownership, time stamps, and access permissions.

In DataSync, a *location* for Amazon FSx for Windows is an endpoint for an FSx for Windows File Server. You can transfer files between a location for Amazon FSx for Windows and a location for other file systems. For information, see Working with Locations in the *AWS DataSync User Guide*.

DataSync accesses your FSx for Windows File Server using the Server Message Block (SMB) protocol. It authenticates by using the user name and password that you configure in the DataSync console or AWS CLI.

Before You Begin

For this step, we assume that you have the following:

- A source location that you can transfer files from. If this source is an Amazon EFS file system, it needs to be accessible over NFS version 3, version 4, or 4.1. Example file systems include those located in on-premises data centers, self-managed in-cloud file systems, and Amazon FSx for Windows file systems.
- A destination file system to transfer files to. Example file systems include those located in on-premises data centers, self-managed in-cloud file systems, and Amazon FSx for Windows file systems. If you don't have an FSx for Windows File Server file system, create one. For more information, see Getting started with Amazon FSx (p. 7).
- A server and network that meet the DataSync requirements. To learn more, see Requirements for DataSync in the *AWS DataSync User Guide*.

When you have the preceding in place, you can begin transfer as discussed following.

Basic Steps for Transferring Files Using DataSync

To transfer files from a source location to a destination location using DataSync, take the following basic steps:

- Download and deploy an agent in your environment and activate it.
- Create and configure a source and destination location.
- Create and configure a task.
- Run the task to transfer files from the source to the destination.

To learn how to transfer files from an existing on-premises file system to your FSx for Windows File Server, see Getting Started with DataSync in the *AWS DataSync User Guide*.

To learn how to transfer files from an existing in-cloud file system to your FSx for Windows File Server, see Deploying the DataSync Agent as an Amazon EC2 Instance in the *AWS DataSync User Guide*.

Step 6: Clean up resources

After you have finished this exercise, you should follow these steps to clean up your resources and protect your AWS account.

To clean up resources

1. On the Amazon EC2 console, terminate your instance. For more information, see Terminate Your Instance in the *Amazon EC2 User Guide for Windows Instances.*
2. On the Amazon FSx console, delete your file system. All automatic backups are deleted automatically. However, you still need to delete the manually created backups. The following steps outline this process:

   a. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
   b. From the console dashboard, choose the name of the file system you created for this exercise.
   c. For **Actions**, choose **Delete file system**.
   d. In the **Delete file system** dialog box that opens, decide whether you want to create a final backup. If you do, provide a name for the final backup. Any automatically created backups are also deleted.

      Important
      New file systems can be created from backups. We recommend that you create a final backup as a best practice. If you find you don't need it after a certain period of time, you can delete this and other manually created backups.

   e. Enter the ID of the file system that you want to delete in the **File system ID** box.
   f. Choose **Delete file system**.
   g. The file system is now being deleted, and its status in the dashboard changes to **DELETING**. When the file system has been deleted, it no longer appears in the dashboard.
   h. Now you can delete any manually created backups for your file system. From the left-side navigation, choose **Backups**.
   i. From the dashboard, choose any backups that have the same **File system ID** as the file system that you deleted, and choose **Delete backup**.
   j. The **Delete backups** dialog box opens. Leave the check box checked for the ID of the backup you selected, and choose **Delete backups**.

   Your Amazon FSx file system and related automatic backups are now deleted.
3. If you created an AWS Directory Service directory for this exercise in Walkthrough 1: Prerequisites for getting started (p. 157), you can delete it now. For more information, see Delete your directory in the *AWS Directory Service Administration Guide.*

Amazon FSx file system status

You can view the status of an Amazon FSx file system by using the Amazon FSx console, the AWS CLI command describe-file-systems, or the API operation DescribeFileSystems.

File system status	Description
AVAILABLE	The file system is in a healthy state, and is reachable and available for use.
CREATING	Amazon FSx is creating a new file system.
DELETING	Amazon FSx is deleting an existing file system.
UPDATING	The file system is undergoing a customer-initiated update.
MISCONFIGURED	The file system is in an impaired state due to a change in your Active Directory environment. Your file system is either currently unavailable or at risk of losing availability, and backups may not succeed. For information on restoring availability, see File system is in a misconfigured state (p. 213).
MISCONFIGURED_UNAVAILABLE	The file system is currently unavailable due to a change in your Active Directory environment. For information on restoring availability, see File system is in a misconfigured state (p. 213).
FAILED	• The file system has failed and Amazon FSx can't recover it. • When creating a new file system, Amazon FSx was unable to create the new file system.

Supported clients, access methods, and environments for Amazon FSx for Windows File Server

You can access your Amazon FSx file systems using a variety of supported clients and methods from both AWS and on-premises environments.

Topics

Supported clients

Amazon FSx supports connecting to your file system from a wide variety of compute instances and operating systems. It does this by supporting access through the Server Message Block (SMB) protocol, versions 2.0 through 3.1.1.

The following AWS compute instances are supported for use with Amazon FSx:

- Amazon Elastic Compute Cloud (Amazon EC2) instances, including Microsoft Windows, Mac, Amazon Linux and Amazon Linux 2 instances. For more information, see Accessing file shares (p. 53).
- Amazon Elastic Container Service (Amazon ECS) containers. For more information, see FSx for Windows File Server volumes in the *Amazon Elastic Container Service Developer Guide*.
- WorkSpaces instances – To learn more, see the AWS blog post Using FSx for Windows File Server with Amazon WorkSpaces.
- Amazon AppStream 2.0 instances – To learn more, see the AWS blog post Using Amazon FSx with Amazon AppStream 2.0.
- VMs running in VMware Cloud on AWS environments – To learn more, see the AWS blog post Storing and Sharing Files with FSx for Windows File Server in a VMware Cloud on AWS Environment.

The following operating systems are supported for use with Amazon FSx:

- Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019.
- Windows Vista, Windows 7, Windows 8, Windows 8.1, and Windows 10 (including the Windows 7 and Windows 10 desktop experiences of WorkSpaces).
- Linux, using the `cifs-utils` tool.

Supported access methods

You can use the following access methods and approaches with Amazon FSx.

Accessing file systems using their default DNS names

FSx for Windows File Server provides a Domain Name System (DNS) name for every file system. You access your FSx for Windows File Server file system by mapping a drive letter on your compute instance to your Amazon FSx file share using this DNS name. To learn more, see Using Microsoft Windows file shares (p. 53).

> **Important**
> Amazon FSx only registers DNS records for a file system if you are using Microsoft DNS as the default DNS. If you are using a third-party DNS, you must manually set up DNS entries for your Amazon FSx file systems. For information about choosing the correct IP addresses to use for the file system, see Obtaining the correct file system IP addresses to use for DNS (p. 49).

To find the DNS name:

- In the Amazon FSx console, choose **File systems**, and then choose **Details**. View the DNS name in the **Network & Security** section.
- Or, view it in the response of the **CreateFileSystem** or **DescribeFileSystems** API command.

For all Single-AZ file systems joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following: `fs-0123456789abcdef0.`*`ad-dns-domain-name`*

For all Single-AZ file systems joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following: `amznfsxaa11bb22.`*`ad-domain`*`.com`

Using DNS names with Kerberos authentication

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients accessing your file system. To enable Kerberos-based authentication and encryption of data in transit for your SMB sessions, use the file system's DNS name provided by Amazon FSx to access your file system.

If you have an external trust configured between your AWS Managed Microsoft Active Directory and your on-premises Active Directory, to use the Amazon FSx Remote PowerShell with Kerberos authentication, you must configure a local group policy on the client for forest search order. For more information, see Configure Kerberos Forest Search Order (KFSO) in the Microsoft documentation.

Accessing file systems using DNS aliases

FSx for Windows File Server provides a DNS name for every file system that you can use to access your file shares. You can also enable access to Amazon FSx from DNS names other than the default DNS name that Amazon FSx creates by registering aliases for your FSx for Windows File Server file systems.

Using DNS aliases, you can move your Windows file share data to Amazon FSx and continue using your existing DNS names to access data on Amazon FSx. DNS aliases also allow you to use meaningful names that make it easier to administer tools and applications to connect to your Amazon FSx file systems. For more information, see Managing DNS aliases (p. 90).

Using DNS aliases with Kerberos authentication

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients accessing your file system. To enable Kerberos authentication for clients that access Amazon FSx using a DNS alias, you must add service principal names (SPNs) that correspond to the DNS alias on your Amazon FSx file system's Active Directory computer object.

You can optionally enforce clients that access the file system using a DNS alias to use Kerberos authentication and encryption by setting the following Group Policy Objects (GPOs) in your Active Directory:

- **Restrict NTLM: Outgoing NTLM traffic to remote servers** - Use this policy setting to deny or audit outgoing NTLM traffic from a computer to any remote server running the Windows operating system.
- **Restrict NTLM: Add remote server exceptions for NTLM authentication** - Use this policy setting to create an exception list of remote servers to which client devices are allowed to use NTLM authentication if the *Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers* policy setting is configured.

For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

Working with FSx for Windows File Server file systems and DFS namespaces

FSx for Windows File Server supports the use of Microsoft Distributed File System (DFS) Namespaces. You can use DFS Namespaces to organize file shares on multiple file systems into one common folder structure (a namespace) that you use to access the entire file dataset. You can use a name in your DFS Namespace to access your Amazon FSx file system by configuring its link target to be the file system's DNS name. For more information, see Grouping multiple file systems with DFS Namespaces (p. 142).

Supported environments

You can access your file system from resources that are in the same VPC as your file system. For more information and detailed instructions, see Walkthrough 1: Prerequisites for getting started (p. 157).

You can also access file systems created after February 22, 2019, from on-premises resources and from resources that are in a different VPC, AWS account, or AWS Region. The following table illustrates the environments from which Amazon FSx supports access from clients in each of the supported environments, depending on when the file system was created.

Clients located in...	Access to file systems created before February 22, 2019	Access to file systems created before December 17, 2020	Access to file systems created after December 17, 2020
Subnets in which the file system is created	✓	✓	✓
Primary CIDR blocks of the VPC in which the file system was created	✓	✓	✓
Secondary CIDRs of the VPC in which the file system was created		Clients with IP addresses in an RFC 1918 private IP address range: • 10.0.0.0/8	Clients with IP addresses outside the following CIDR block range: 198.19.0.0/16

Clients located in...	Access to file systems created before February 22, 2019	Access to file systems created before December 17, 2020	Access to file systems created after December 17, 2020
Other CIDRs or peered networks		• 172.16.0.0/12 • 192.168.0.0/16	

Note
In some cases, you might want to access a file system that was created before December 17, 2020 from on-premises using a non-private IP address range. To do this, create a new file system from a backup of the file system. For more information, see Working with backups (p. 78).

Following, you can find information about how to access your FSx for Windows File Server file systems from on-premises and from different VPCs, AWS accounts, or AWS Regions.

Accessing FSx for Windows File Server file systems from on-premises

FSx for Windows File Server supports the use of AWS Direct Connect or AWS VPN to access your file systems from your on-premises compute instances. With support for AWS Direct Connect, FSx for Windows File Server enables you to access your file system over a dedicated network connection from your on-premises environment. With support for AWS VPN, FSx for Windows File Server enables you to access your file system from your on-premises devices over a secure and private tunnel.

After you connect your on-premises environment to the VPC associated with your Amazon FSx file system, you can access your file system using its DNS name or a DNS alias. You do so just as you do from compute instances within the VPC. For more information on AWS Direct Connect, see the *AWS Direct Connect User Guide*. For more information on setting up AWS VPN connections, see VPN Connections in the *Amazon VPC User Guide*.

FSx for Windows File Server also supports the use of Amazon FSx File Gateway to provide low latency, seamless access to your in-cloud FSx for Windows File Server file shares from your on-premises compute instances. For more information, see the *Amazon FSx File Gateway User Guide*.

Accessing FSx for Windows File Server file systems from another VPC, account, or AWS Region

You can access your FSx for Windows File Server file system from compute instances in a different VPC, AWS account, or AWS Region from that associated with your file system. To do so, you can use VPC peering or transit gateways. When you use a VPC peering connection or transit gateway to connect VPCs, compute instances that are in one VPC can access Amazon FSx file systems in another VPC. This access is possible even if the VPCs belong to different accounts, and even if the VPCs reside in different AWS Regions.

A *VPC peering connection* is a networking connection between two VPCs that you can use to route traffic between them using private IPv4 or IP version 6 (IPv6) addresses. You can use VPC peering to connect VPCs within the same AWS Region or between AWS Regions. For more information on VPC peering, see What is VPC Peering? in the *Amazon VPC Peering Guide*.

A *transit gateway* is a network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information about using VPC transit gateways, see Getting Started with Transit Gateways in the *Amazon VPC Transit Gateways*.

After you set up a VPC peering or transit gateway connection, you can access your file system using its DNS name. You do so just as you do from compute instances within the associated VPC.

Availability and durability: Single-AZ and Multi-AZ file systems

Amazon FSx for Windows File Server offers two file system deployment types: Single-AZ and Multi-AZ.

Choosing Single-AZ or Multi-AZ file system deployment

With Single-AZ file systems, Amazon FSx automatically replicates your data within an Availability Zone (AZ) to protect it from component failure. It continuously monitors for hardware failures and automatically replaces infrastructure components in the event of a failure. Amazon FSx also uses the Windows Volume Shadow Copy Service to make highly durable backups of your file system daily and store them in Amazon S3. You can make additional backups at any point. *Single-AZ 2* is the latest generation of Single-AZ file systems, and it supports both SSD and HDD storage. *Single-AZ 1* file systems support SSD storage, Microsoft Distributed File System Replication (DFSR), and the use of custom DNS names.

Multi-AZ file systems support all the availability and durability features of Single-AZ file systems. In addition, they are designed to provide continuous availability to data, even when an Availability Zone is unavailable. In a Multi-AZ deployment, Amazon FSx automatically provisions and maintains a standby file server in a different Availability Zone. Any changes written to disk in your file system are synchronously replicated across Availability Zones to the standby. With Amazon FSx Multi-AZ deployments can enhance availability during planned system maintenance, and help protect your data against instance failure and Availability Zone disruption. If there is planned file system maintenance or unplanned service disruption, Amazon FSx automatically fails over to the secondary file server, allowing you to continue accessing your data without manual intervention.

Multi-AZ file systems are ideal for business-critical workloads that require high availability to shared Windows file data. Examples of these include business applications, web serving environments, and Microsoft SQL Server. Single-AZ file systems offer a lower price point for workloads that don't require the high availability of a Multi-AZ solution and that can recover from the most recent file system backup if data is lost. Amazon FSx takes automatic daily backups of all file systems by default.

Feature support by deployment types

The following table summarizes features supported by the FSx for Windows File Server file system deployment types:

Deployment type	SSD storage	HDD storage	DFS namespaces	DFS replication	Custom DNS names	CA shares
Single-AZ 1	✓		✓	✓	✓	
Single-AZ 2	✓	✓	✓		✓	✓*
Multi-AZ	✓	✓	✓		✓	✓*

Note
* While you can create CA shares on Single-AZ 2 file systems, you should use CA shares on Multi-AZ file systems for SQL Server HA deployments.

Failover process for FSx for Windows File Server

Multi-AZ file systems automatically fail over from the preferred file server to the standby file server if any of the following conditions occur:

- An Availability Zone outage occurs.
- The preferred file server becomes unavailable.
- The preferred file server undergoes planned maintenance.

When failing over from one file server to another, the new active file server automatically begins serving all file system read and write requests. When the resources in the preferred subnet are available, Amazon FSx automatically fails back to the preferred file server in the preferred subnet. A failover typically completes in less than 30 seconds from the detection of the failure on the active file server to the promotion of the standby file server to active status. Failback to the original Multi-AZ configuration also completes in less than 30 seconds, and only occurs once the file server in the preferred subnet is fully recovered.

During the brief period in which your file system is failing over and failing back, I/O may be paused and Amazon CloudWatch metrics may be temporarily unavailable.

Failover experience on Windows clients

When failing over from one file server to another, the new active file server automatically begins serving all file system read and write requests. After the resources in the preferred subnet are available, Amazon FSx automatically fails back to the preferred file server in the preferred subnet. Because the file system's DNS name remains the same, failovers are transparent to Windows applications, which resume file system operations without manual intervention. A failover typically completes in less than 30 seconds from the detection of the failure on the active file server to the promotion of the standby file server to active status. Failback to the original Multi-AZ configuration also completes in less than 30 seconds, and only occurs after the file server in the preferred subnet is fully recovered.

Failover experience on Linux clients

Linux clients do not support automatic DNS-based failover. Therefore, they don't automatically connect to the standby file server during a failover. They will automatically resume file system operations after the Multi-AZ file system has failed back to the file server in the preferred subnet.

Testing failover on a file system

You can test failover your Multi-AZ file system by modifying its throughput capacity. When you modify your file system's throughput capacity, Amazon FSx switches out the file system's file server. Multi-AZ file systems automatically fail over to the secondary server while Amazon FSx replaces the preferred server file server first. Then the file system automatically fails back to the new primary server and Amazon FSx replaces the secondary file server.

You can monitor the progress of the throughput capacity update request in the Amazon FSx console, the CLI, and the API. Once the update has completed successfully, your file system has failed over to the secondary server, and failed back to the primary server. For more information about modifying your file

system's throughput capacity and monitoring the progress of the request, see Managing throughput capacity (p. 133).

Working with Single and Multi-AZ file system resources

Subnets

When you create a VPC, it spans all the Availability Zones (AZs) in the Region. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. After creating a VPC, you can add one or more subnets in each Availability Zone. The default VPC has a subnet in each Availability Zone. Each subnet must reside entirely within one Availability Zone and cannot span zones. When you create a Single-AZ Amazon FSx file system, you specify a single subnet for the file system. The subnet you choose defines the Availability Zone in which the file system is created.

When you create a Multi-AZ file system, you specify two subnets, one for the preferred file server, and one for the standby file server. The two subnets you choose must be in different Availability Zones within the same AWS Region.

For in-AWS applications, we recommend that you launch your clients in the same Availability Zone as your preferred file server to minimize latency.

File system elastic network interfaces

When you create an Amazon FSx file system, Amazon FSx provisions one or more elastic network interfaces in the Amazon Virtual Private Cloud (VPC) that you associate with your file system. The network interface allows your client to communicate with the FSx for Windows File Server file system. The network interface is considered to be within the service scope of Amazon FSx, despite being part of your account's VPC. Multi-AZ file systems have two elastic network interfaces, one for each file server. Single-AZ file systems have one elastic network interface.

Warning
You must not modify or delete the elastic network interfaces associated with your file system. Modifying or deleting the network interface can cause a permanent loss of connection between your VPC and your file system.

The following table summarizes the subnet, elastic network interface, and IP address resources for FSx for Windows File Server file system deployment types:

File system deployment type	Number of subnets	Number of elastic network interfaces	Number of IP addresses
Single-AZ 2	1	1	2
Single-AZ 1	1	1	1
Multi-AZ	2	2	4

Once a file system is created, its IP addresses don't change until the file system is deleted.

Important
Amazon FSx doesn't support accessing file systems from, or exposing file system to the public Internet. If an Elastic IP address, which is a public IP address reachable from the Internet, gets attached to a file system's elastic network interface, Amazon FSx automatically detaches it.

Optimizing costs with Amazon FSx

FSx for Windows File Server provides several features to help you optimize your total cost of ownership (TCO) based on your application needs. You can pick the storage type (HDD or SSD) to achieve the right balance of cost and performance needs for your application. You have the flexibility to pick throughput capacity separately from the amount of storage capacity to optimize your costs. And, you can use data deduplication to optimize storage costs by eliminating redundant data on your file system.

Topics

Flexibility to choose storage and throughput independently

With FSx for Windows File Server, you can configure your file system's storage and throughput capacities independently. This gives you flexibility to achieve the right mix of cost and performance. For example, you can choose to have a large amount of storage with a relatively small amount of throughput capacity for cold (generally inactive) workloads to save on unneeded throughput costs. Or, as another example, you could choose to have a large amount of throughput capacity for a relatively small amount of storage capacity. Higher throughput capacity comes with higher amounts of memory for caching on the file server. You can take advantage of fast caching on the file server to optimize performance for actively accessed data. For more information, see FSx for Windows File Server performance (p. 153).

You can increase or decrease the amount of throughput capacity at any time, providing you flexibility to address changing performance needs. For more information, see Managing throughput capacity (p. 133). You can increase the amount of storage cacapacity anytime after you create a file system. For more information, see Managing storage capacity (p. 123).

Optimizing storage costs

You can optimize your storage costs with Amazon FSx in a variety of ways, described as follows.

Optimizing costs using storage types

FSx for Windows File Server provides two types of storage—hard disk drives (HDD) and solid state drives (SSD)—to enable you to optimize cost/performance to meet your workload needs. HDD storage is designed for a broad spectrum of workloads, including home directories, user and departmental shares, and content management systems. SSD storage is designed for the highest-performance and most latency-sensitive workloads, including databases, media processing workloads, and data analytics applications. For more information, see Latency (p. 153) and Amazon FSx for Windows File Server Pricing.

Optimizing storage costs using data deduplication

Large datasets often have redundant data, which increases data storage costs. For example, user file shares can have multiple copies of the same file, stored by multiple users. Software development shares

can contain many binaries that remain unchanged from build to build. You can reduce your data storage costs by turning on *data deduplication* for your file system. When it's turned on, data deduplication automatically reduces or eliminates redundant data by storing duplicated portions of the dataset only once. For more information about data deduplication, and how to easily turn it on for your Amazon FSx file system, see Data deduplication (p. 113).

Working with Microsoft Active Directory in FSx for Windows File Server

Amazon FSx works with Microsoft Active Directory (AD) to integrate with your existing Microsoft Windows environments. Active Directory is the Microsoft directory service used to store information about objects on the network and make this information easy for administrators and users to find and use. These objects typically include shared resources such as file servers and network user and computer accounts.

When you create a file system with Amazon FSx, you join it to your Active Directory domain to provide user authentication and file- and folder-level access control. Your users can then use their existing user identities in Active Directory to authenticate themselves and access the Amazon FSx file system. Users can also use their existing identities to control access to individual files and folders. In addition, you can migrate your existing files and folders and these items' security access control list (ACL) configuration to Amazon FSx without any modifications.

Amazon FSx provides you with two options for using your FSx for Windows File Server file system with Active Directory: Using Amazon FSx with AWS Directory Service for Microsoft Active Directory (p. 26) and Using Amazon FSx with your self-managed Microsoft Active Directory (p. 33).

> **Note**
> Amazon FSx supports Microsoft Azure Active Directory Domain Services, which you can join to a Microsoft Azure Active Directory.

After you create a joined Active Directory configuration for a file system, you can update only the following properties:

- Service user credentials
- DNS server IP addresses

You *cannot* change the following properties for your joined Microsoft AD:

- DomainName
- OrganizationalUnitDistinguishedName
- FileSystemAdministratorsGroup

However, you can create a new file system from a backup and change these properties in the Microsoft Active Directory integration configuration for that file system. For more information, see Walkthrough 2: Create a file system from a backup (p. 161).

> **Note**
> Amazon FSx does not support Active Directory Connector and Simple Active Directory.

Topics

Using Amazon FSx with AWS Directory Service for Microsoft Active Directory

AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) provides fully managed, highly available, actual Active Directory (AD) directories in the cloud. You can use these AD directories in your workload deployment.

If your organization is using AWS Managed Microsoft AD to manage identities and devices, we recommend that you integrate your Amazon FSx file system with AWS Managed Microsoft AD. By doing this, you get a turnkey solution using Amazon FSx with AWS Managed Microsoft AD. AWS handles the deployment, operation, high availability, reliability, security, and seamless integration of the two services, enabling you to focus on operating your own workload effectively.

To use Amazon FSx with your AWS Managed Microsoft AD setup, you can use the Amazon FSx console. When you create a new FSx for Windows File Server file system in the console, choose **AWS Managed AD** under the **Windows Authentication** section. You also choose the specific directory that you want to use. For more information, see Step 1: Create your file system (p. 7).

Your organization might manage identities and devices on a self-managed Active Directory domain (on-premises or in the cloud). If so, you can join your Amazon FSx file system directly to your existing, self-managed AD domain. For more information, see Using Amazon FSx with your self-managed Microsoft Active Directory (p. 33).

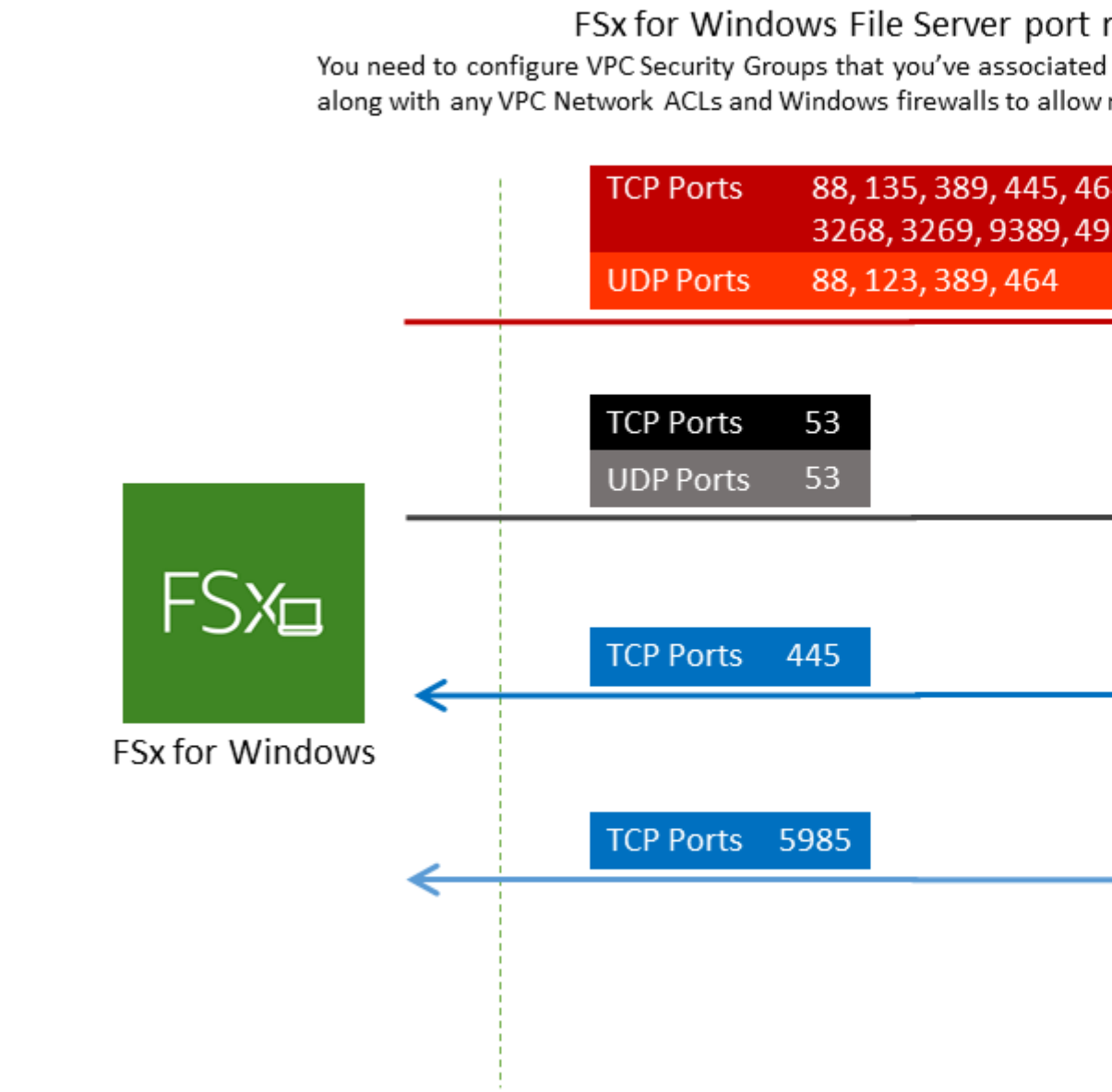
Additionally, you can also set up your system to benefit from a resource forest isolation model. In this model, you isolate your resources, including your Amazon FSx file systems, into a separate AD forest from the one where your users are.

> **Important**
> For Single-AZ 2 and all Multi-AZ file systems, the Active Directory domain name cannot exceed 47 characters.

Networking prerequisites

Before you create an FSx for Windows File Server file system joined to your AWS Microsoft Managed AD domain, make sure that you have created and set up the following network configurations:

- For **VPC security groups**, the default security group for your default Amazon VPC is already added to your file system in the console. Please ensure that the security group and the VPC Network ACLs for the subnet(s) where you're creating your FSx file system allow traffic on the ports and in the directions shown in the following diagram.



The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Kerberos authenticatio
TCP/UDP	464	Change/ Set password
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
UDP	123	Network Time Protocol (NTP)
TCP	135	Distributed Computing Environment End Point Mapper (DCE / EPMAP)
TCP	445	Directory Services SMB file sharing
TCP	636	Lightweight Directory Access Protocol over TLS/ SSL (LDAPS)
TCP	3268	Microsoft Global Catalog

Protocol	Ports	Role
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management
TCP	9389	Microsoft AD DS Web Services, PowerShell
TCP	49152 - 65535	Ephemeral ports for RPC

Important
Allowing outbound traffic on TCP port 9389 is required for Single-AZ 2 and all Multi-AZ file system deployments.

Note
If you're using VPC network ACLs, you must also allow outbound traffic on dynamic ports (49152-65535) from your FSx file system.

- If you are connecting your Amazon FSx file system to an AWS Managed Microsoft AD in a different VPC or account, then ensure connectivity between that VPC and the Amazon VPC where you want to create the file system. For more information, see Using Amazon FSx with AWS Managed Microsoft AD in a different VPC or account (p. 30).

  Important
  While Amazon VPC security groups require ports to be opened only in the direction that network traffic is initiated, VPC network ACLs require ports to be open in both directions.

Use the Amazon FSx Network Validation tool (p. 31) to validate connectivity to your Active Directory domain controllers.

Using a resource forest isolation model

You join your file system to an AWS Managed Microsoft AD setup. You then establish a one-way forest trust relationship between an AWS Managed Microsoft AD domain that you create and your existing self-managed AD domain. For Windows authentication in Amazon FSx, you only need a one-way directional forest trust, where the AWS managed forest trusts the corporate domain forest.

Your corporate domain takes the role of the trusted domain, and the AWS Directory Service managed domain takes the role of the trusting domain. Validated authentication requests travel between the domains in only one direction—allowing accounts in your corporate domain to authenticate against

resources shared in the managed domain. In this case, Amazon FSx interacts only with the managed domain. The managed domain then passes on the authentication requests to your corporate domain.

Test your Active Directory configuration

Before creating your Amazon FSx file system, we recommend that you validate the connectivity to your Active Directory domain controllers using the Amazon FSx Network Validation tool. For more information, see Validating connectivity to your Active Directory domain controllers (p. 31).

The following related resources can help you as you use AWS Directory Service for Microsoft Active Directory with FSx for Windows File Server:

- What Is AWS Directory Service in the *AWS Directory Service Administration Guide*
- Create Your AWS Managed AD Directory in the *AWS Directory Service Administration Guide*
- When to Create a Trust Relationship in the *AWS Directory Service Administration Guide*
- Walkthrough 1: Prerequisites for getting started (p. 157)

Using Amazon FSx with AWS Managed Microsoft AD in a different VPC or account

You can join your FSx for Windows File Server file system to an AWS Managed Microsoft AD directory that's in a different VPC within the same account by using VPC peering. You can also join your file system to an AWS Managed Microsoft AD directory that's in a different AWS account by using directory sharing.

The workflow for joining your file system to an AWS Managed Microsoft AD that's in a different VPC involves the following steps:

1. Set up your networking environment.
2. Share your directory.
3. Join your file system to the shared directory.

For more information, see Share your directory in the *AWS Directory Service Administration Guide*.

To set up your networking environment you can use AWS Transit Gateway or Amazon VPC and create a VPC peering connection. In addition, make sure that network traffic is allowed between the two VPCs.

A *transit gateway* is a network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information about using VPC transit gateways, see Getting Started with Transit Gateways in the *Amazon VPC Transit Gateways Guide*.

A *VPC peering connection* is a networking connection between two VPCs. This connection enables you to route traffic between them using private Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) addresses. You can use VPC peering to connect VPCs within the same AWS Region or between AWS Regions. For more information on VPC peering, see What is VPC Peering? in the *Amazon VPC Peering Guide*.

There is another prerequisite when you join your file system to an AWS Managed Microsoft AD directory in a different account than that of your file system. You also need to share your Microsoft AD directory with the other account. To do this, you can use AWS Managed Microsoft Active Directory's directory sharing feature. To learn more, see Share your directory in the *AWS Directory Service Administration Guide*.

Validating connectivity to your Active Directory domain controllers

Before you create an FSx for Windows File Server file system joined to your Active Directory, use the Amazon FSx Active Directory Validation tool to validate the connectivity to your Active Directory domain. You can use this test whether you are using FSx for Windows File Server with AWS Managed Microsoft Active Directory or with a self-managed Active Directory configuration. The Domain Controller Network Connectivity test (Test-FSxADControllerConnection) does not run the full suite of network connectivity checks against every domain controller in the domain. Instead, use this test to run network connectivity validation against a specific set of domain controllers.

To validate connectivity to your Active Directory domain controllers

1. Launch an Amazon EC2 Windows instance in the same subnet and with the same Amazon VPC security groups that you will use for your FSx for Windows File Server file system. For Multi-AZ deployment types, use the subnet for the preferred active file server.
2. Join your EC2 Windows instance to your Active Directory. For more information, see Manually Join a Windows Instance in the *AWS Directory Service Administration Guide*.
3. Connect to your EC2 instance. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
4. Open a Windows PowerShell window (using **Run as Administrator**) on the EC2 instance.

   To test whether the required Active Directory module for Windows PowerShell is installed, use the following test command.

   ```
   PS C:\> Import-Module ActiveDirectory
   ```

   If above returns an error, install it using the following command.

   ```
   PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
   ```
5. Download the network validation tool using the following command.

   ```
   PS C:\> Invoke-WebRequest "https://docs.aws.amazon.com/fsx/latest/WindowsGuide/samples/
   AmazonFSxADValidation.zip" -OutFile "AmazonFSxADValidation.zip"
   ```
6. Expand the zip file by using the following command.

   ```
   PS C:\> Expand-Archive -Path "AmazonFSxADValidation.zip"
   ```
7. Add the AmazonFSxADValidation module to the current session.

   ```
   PS C:\> Import-Module .\AmazonFSxADValidation
   ```
8. Set the value for the Active Directory domain controller IP address and run the connectivity test using the following commands:

   ```
   $ADControllerIp = '10.0.75.243'
   $Result = Test-FSxADControllerConnection -ADControllerIp $ADControllerIp
   ```

9. The following example demonstrates retrieving the test output, with results of a successful connectivity test.

```
PS C:\AmazonFSxADValidation> $Result

Name                           Value
----                           -----
TcpDetails                     {@{Port=88; Result=Listening; Description=Kerberos
 authentication}, @{Port=135; Resul...
Server                         10.0.75.243
UdpDetails                     {@{Port=88; Result=Timed Out; Description=Kerberos
 authentication}, @{Port=123; Resul...
Success                        True


PS C:\AmazonFSxADValidation> $Result.TcpDetails

Port Result    Description
---- ------    -----------
  88 Listening Kerberos authentication
 135 Listening DCE / EPMAP (End Point Mapper)
 389 Listening Lightweight Directory Access Protocol (LDAP)
 445 Listening Directory Services SMB file sharing
 464 Listening Kerberos Change/Set password
 636 Listening Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
3268 Listening Microsoft Global Catalog
3269 Listening Microsoft Global Catalog over SSL
9389 Listening Microsoft AD DS Web Services, PowerShell
```

The following example shows running the test and getting a failed result.

```
PS C:\AmazonFSxADValidation> $Result = Test-FSxADControllerConnection -ADControllerIp
 $ADControllerIp
WARNING: TCP 9389 failed to connect. Required for Microsoft AD DS Web Services,
 PowerShell.
Verify security group and firewall settings on both client and directory controller.
WARNING: 1 ports failed to connect to 10.0.75.243. Check pre-requisites in
https://docs.aws.amazon.com/fsx/latest/WindowsGuide/self-managed-AD.html#self-manage-
prereqs

PS C:\AmazonFSxADValidation> $Result

Name                           Value
----                           -----
TcpDetails                     {@{Port=88; Result=Listening; Description=Kerberos
 authentication}, @{Port=135; Resul...
Server                         10.0.75.243
UdpDetails                     {@{Port=88; Result=Timed Out; Description=Kerberos
 authentication}, @{Port=123; Resul...
Success                        False
FailedTcpPorts                 {9389}


PS C:\AmazonFSxADValidation> $Result.FailedTcpPorts
9389
```

Windows socket error code mapping

https://msdn.microsoft.com/en-us/library/ms740668.aspx

Using Amazon FSx with your self-managed Microsoft Active Directory

Your organization might manage identities and devices on a self-managed Active Directory (on-premises or in the cloud). If so, you can join your Amazon FSx file system directly to your existing self-managed AD domain. To use Amazon FSx with your AWS Managed Microsoft AD setup, you can use the Amazon FSx console. When you create a new FSx for Windows File Server file system in the console, choose **Self-managed Microsoft Active Directory** under the **Windows Authentication** section. Provide the following details for your self-managed AD:

- A fully qualified domain name of your self-managed directory

  Note
  Domain name must not be in the Single Label Domain (SLD) format. Amazon FSx currently does not support SLD domains.

  Note
  For Single-AZ 2 and all Multi-AZ file systems, the Active Directory domain name cannot exceed 47 characters.

- IP addresses of the DNS servers for your domain

  The DNS server IP addresses, AD domain controller IP addresses, and client network must meet the following requirements:

For file systems created before December 17, 2020	For file systems created after December 17, 2020
IP addresses in an RFC 1918 private IP address range: • 10.0.0.0/8 • 172.16.0.0/12 • 192.168.0.0/16	Any IP address range, except: • IP addresses that conflict with Amazon Web Services-owned IP addresses in that AWS Region. For a list of AWS-owned IP addresses by region, see the AWS IP address ranges. • IP addresses in the following CIDR block range: 198.19.0.0/16

  Note
  Your AD domain controllers must be writable.

- User name and password for a service account on your AD domain, for Amazon FSx to use to join the file system to your AD domain
- (Optional) The Organizational Unit (OU) in your domain in which you want your file system to be joined
- (Optional) The domain group to which you want to delegate authority to perform administrative actions on your file system. For example, this domain group might manage Windows file shares, manage ACLs on the file system's root folder, take ownership of files and folders, and so on. If you don't specify this group, Amazon FSx delegates this authority to the Domain Admins group in your AD domain by default.

  For more information, see Joining an Amazon FSx file system to a self-managed Microsoft Active Directory domain (p. 43).

Important
Amazon FSx only registers DNS records for a file system if you are using Microsoft DNS as the default DNS service. If you are using a third-party DNS, you will need to manually setup DNS entries for your Amazon FSx file systems after you create them.

When you join your file system directly to your self-managed AD, your FSx for Windows File Server resides in the same AD forest (the top-most logical container in an AD configuration that contains domains, users, and computers) and in the same AD domain as your users and existing resources (including existing file servers).

Note
If you'd like to benefit from a resource forest isolation model, where you isolate your resources, including your Amazon FSx file systems, into a separate AD forest from the one where your users reside, you can alternately choose to join your file system to an AWS Managed AD and establish a one-way forest trust relationship between an AWS Managed AD that you create and your existing self-managed AD.

Topics

Prerequisites for using a self-managed Microsoft AD

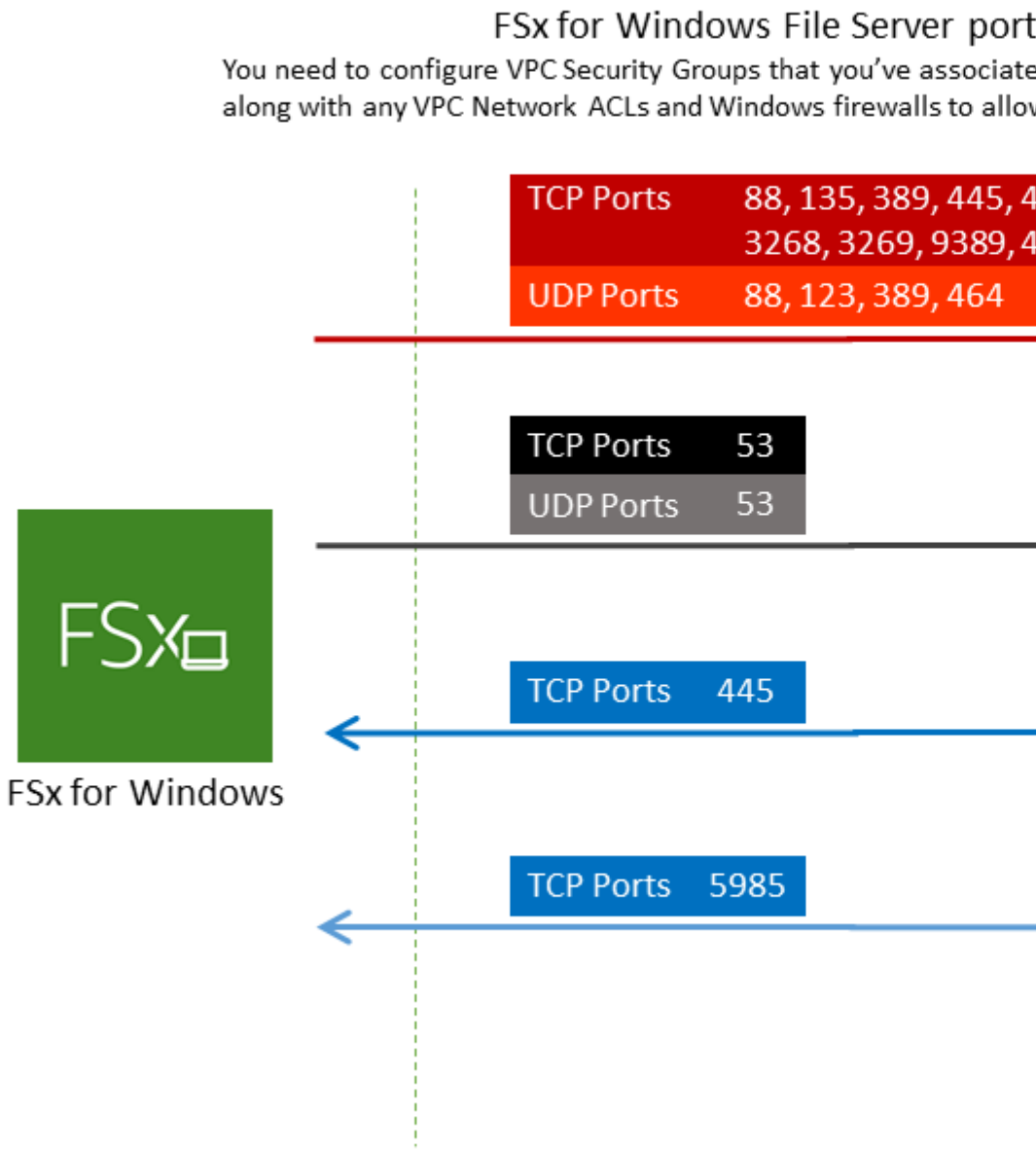
Before you create an Amazon FSx file system joined to your self-managed Microsoft AD domain, make sure that you have created and set up the following requirements:

- An on-premises or other self-managed Microsoft AD that the Amazon FSx file system is to join, with the following configuration:
  - The domain functional level of your AD domain controller is at Windows Server 2008 R2 or higher.
  - DNS server IP addresses and AD domain controller IP addresses as follows, depending on when your file system was created:

For file systems created before December 17, 2020	For file systems created after December 17, 2020
IP addresses in an RFC 1918 private IP address range: • 10.0.0.0/8 • 172.16.0.0/12 • 192.168.0.0/16	Any IP address range, except: • IP addresses that conflict with Amazon Web Services-owned IP addresses in that AWS Region. For a list of AWS-owned IP addresses by region, see the AWS IP address ranges. • IP addresses in the following CIDR block range: 198.19.0.0/16

If you need to access your FSx for Windows File Server file system that was created before December 17, 2020 using a non-private IP address range, you can create a new file system by restoring a backup of the file system. For more information, see Working with backups (p. 78).

  - Domain name that is not in the Single Label Domain (SLD) format. Amazon FSx does not support SLD domains.
  - For Single-AZ 2 and all Multi-AZ file systems, the Active Directory domain name cannot exceed 47 characters.
  - If you have Active Directory sites defined, you must make sure that the subnets in the VPC associated with your Amazon FSx file system are defined in an Active Directory site, and that no conflicts exist between the subnets in your VPC and the subnets in your other sites.
- The following network configurations:
  - Connectivity configured between the Amazon VPC where you want to create the file system and your self-managed Active Directory. You can set up connectivity using AWS Direct Connect, AWS VPN, VPC peering, or AWS Transit Gateway.
  - For **VPC security groups**, the default security group for your default Amazon VPC is already added to your file system in the console. Please ensure that the security group and the VPC Network ACLs for the subnet(s) where you're creating your FSx file system allow traffic on the ports and in the directions shown in the following diagram.



The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Domain Name System (DNS)

Protocol	Ports	Role
TCP/UDP	88	Kerberos authentication
TCP/UDP	464	Change/Set password
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
UDP	123	Network Time Protocol (NTP)
TCP	135	Distributed Computing Environment / End Point Mapper (DCE / EPMAP)
TCP	445	Directory Services SMB file sharing
TCP	636	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
TCP	3268	Microsoft Global Catalog
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management)
TCP	9389	Microsoft AD DS Web Services, PowerShell
TCP	49152 - 65535	Ephemeral ports for RPC

Important
Allowing outbound traffic on TCP port 9389 is required for Single-AZ 2 and all Multi-AZ file system deployments.

Note
If you're using VPC network ACLs, you must also allow outbound traffic on dynamic ports (49152-65535) from your FSx file system.

- Ensure that these traffic rules are also mirrored on the firewalls that apply to each of the AD domain controllers, DNS servers, FSx clients and FSx administrators.

  Important
  While Amazon VPC security groups require ports to be opened only in the direction that network traffic is initiated, most Windows firewalls and VPC network ACLs require ports to be open in both directions.

Use the Amazon FSx Active Directory Validation tool (p. 40) to test these network settings before attempting to join your file system to your self-managed AD.

- A service account in your self-managed Microsoft AD with delegated permissions to join computers to the domain. A *service account* is a user account in your self-managed Microsoft AD that has been delegated certain tasks.

  The service account also needs to, at a minimum, be delegated the following permissions in the OU that you're joining the file system to:

  - Ability to reset passwords
  - Ability to restrict accounts from reading and writing data
  - Validated ability to write to the DNS host name
  - Validated ability to write to the service principal name
  - Be delegated control to create and delete computer objects
  - Validated ability to read and write Account Restrictions

These represent the minimum set of permissions that are required to join computer objects to your Active Directory. For more information, see the Microsoft Windows Server documentation topic Error: Access is denied when non-administrator users who have been delegated control try to join computers to a domain controller.

To learn more about creating a service account with the correct permissions, see Delegating privileges to your Amazon FSx service account (p. 38).

Note
Amazon FSx requires a valid service account throughout the lifetime of your Amazon FSx file system. Amazon FSx must be able to fully manage the file system and perform tasks that require unjoining and rejoining your AD domain using, such as replacing a failed file server or patching Windows Server software. Please keep your Active Directory configuration, including the service account credentials, updated with Amazon FSx. To learn how, see Keeping your Active Directory configuration updated with Amazon FSx (p. 39).

Note
Amazon FSx requires connectivity to all domain controllers in your AD environment. If you have multiple domain controllers, ensure that all of them meet the requirements above, and ensure that any changes to your service account are propagated to all domain controllers. You can validate your AD configuration (including testing connectivity of multiple domain controllers) using the Amazon FSx Active Directory Validation tool (p. 40).

If this is your first time using AWS and FSx for Windows File Server, make sure to set up before starting. For more information, see Setting up (p. 5).

Important
Do not move computer objects that Amazon FSx creates in the OU after your file system is created. Doing so will cause your file system to become misconfigured.

Best practices for joining FSx for Windows File Server file systems to a self-managed Microsoft Active Directory domain

Here are some suggestions and guidelines you should consider when joining Amazon FSx for Windows File Server file systems to your self-managed Microsoft Active Directory. Note that these are recommended as best practices, but not required.

Delegating privileges to your Amazon FSx service account

Make sure to configure the service account that you provide to Amazon FSx with the minimum privileges required. In addition, segregate the Organizational Unit (OU) from other domain controller concerns.

To join Amazon FSx file systems to your domain, make sure that the service account has delegated privileges. Members of the **Domain Admins** group have sufficient privileges to perform this task. However, as a best practice, use a service account that only has the minimum privileges necessary to do this. The following procedure demonstrates how to delegate just the privileges necessary to join Amazon FSx file systems to your domain.

Perform this procedure on a machine that is joined to your directory and has the Active Directory User and Computers MMC snap-in installed.

To create a service account for your Active Directory domain

1. Make sure that you are logged in as a domain administrator for your Active Directory domain.

2. Open the **Active Directory User and Computers** MMC snap-in.
3. In the task pane, expand the domain node.
4. Locate and open the context (right-click) menu for the OU that you want to modify, and then choose **Delegate Control**.
5. On the **Delegation of Control Wizard** page, choose **Next**.
6. Choose **Add** to add a specific user or a specific group for **Selected users and groups**, and then choose **Next**.
7. On the **Tasks to Delegate** page, choose **Create a custom task to delegate**, and then choose **Next**.
8. Choose **Only the following objects in the folder**, and then choose **Computer objects**.
9. Choose **Create selected objects in this folder** and **Delete selected objects in this folder**. Then choose **Next**.
10. For **Permissions**, choose the following:
    - **Reset Password**
    - **Read and write Account Restriction**s
    - **Validated write to DNS host name**
    - **Validated write to service principal name**
11. Choose **Next**, and then choose **Finish**.
12. Close the Active Directory User and Computers MMC snap-in.

Important
Do not move computer objects that Amazon FSx creates in the OU after your file system is created. Doing so will cause your file system to become misconfigured.

Keeping your Active Directory configuration updated with Amazon FSx

To help ensure continued, uninterrupted availability of your Amazon FSx file system, update the file system's self-managed Active Directory (AD) configuration any time that you make changes to your self-managed AD setup.

For example, suppose that your AD uses a time-based password reset policy. In this case, as soon as the password is reset, make sure to update the service account password with Amazon FSx. To do this, use the Amazon FSx console, Amazon FSx API, or AWS CLI. Similarly, if the DNS server IP addresses change for your Active Directory domain, as soon as the change occurs update the DNS server IP addresses with Amazon FSx. Again, do this using the Amazon FSx console, API, or CLI.

When you update the self-managed AD configuration for your Amazon FSx file system, your file system's state switches from **Available** to **Updating** while the update is applied. Verify that the state switches back to **Available** after the update has been applied – note that the update can take up to several minutes to complete. For more information, see Updating the self-managed Active Directory configuration (p. 49).

If there's an issue with the updated self-managed AD configuration, the file system state switches to **Misconfigured**. This state shows an error message and recommended corrective action beside the file system description in the console, API, and CLI. After taking the recommended corrective action, verify that your file system's state eventually changes to **Available** – note that this change can take several minutes to complete.

To learn more about troubleshooting possible self-managed AD misconfigurations, see File system is in a misconfigured state (p. 213).

Using security groups to limit traffic within your VPC

To limit network traffic in your virtual private cloud (VPC), you can implement the principle of least privilege in your VPC. In other words, you can limit privileges to the minimum ones necessary. To do this, use security group rules. To learn more, see Amazon VPC Security Groups (p. 177).

Creating outbound security group rules for your file system's network interface

For greater security, consider configuring a security group with outbound traffic rules. These rules should allow outbound traffic only to your self-managed Microsoft AD domains controllers or within the subnet or security group. Apply this security group to the VPC associated with your Amazon FSx file system's elastic network interface. To learn more, see File System Access Control with Amazon VPC (p. 177).

Validating your Active Directory configuration

Before you create an FSx for Windows File Server file system joined to your Active Directory, we recommend that you validate your Active Directory configuration using the Amazon FSx Active Directory Validation tool.

To validate your Active Directory configuration

1. Launch an Amazon EC2 Windows instance in the same subnet and with the same Amazon VPC security groups that you will use for your FSx for Windows File Server file system. Please ensure that your EC2 instance has the required `AmazonEC2ReadOnlyAccess` IAM permissions. You can validate EC2 instance role permissions using the IAM policy simulator. For more information, see Testing IAM Policies with the IAM Policy Simulator in the *IAM User Guide*.
2. Join your EC2 Windows instance to your Active Directory. For more information, see Manually Join a Windows Instance in the *AWS Directory Service Administration Guide*.
3. Connect to your EC2 instance. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
4. Open a Windows PowerShell window (using **Run as Administrator**) on the EC2 instance.

   To test whether the required Active Directory module for Windows PowerShell is installed, use the following test command.

   ```
   PS C:\> Import-Module ActiveDirectory
   ```

   If above returns an error, install it using the following command.

   ```
   PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
   ```
5. Download the network validation tool using the following command.

   ```
   PS C:\> Invoke-WebRequest "https://docs.aws.amazon.com/fsx/latest/WindowsGuide/samples/
   AmazonFSxADValidation.zip" -OutFile "AmazonFSxADValidation.zip"
   ```
6. Expand the zip file by using the following command.

   ```
   PS C:\> Expand-Archive -Path "AmazonFSxADValidation.zip"
   ```
7. Add the `AmazonFSxADValidation` module to the current session.

```
PS C:\> Import-Module .\AmazonFSxADValidation
```

8. Set required parameters by substituting into the following command your:

   - Active Directory domain name (*DOMAINNAME.COM*)
   - Prepare the `$Credential` object for the service account password using one of the following options.
     - To generate the credential object interactively, use the following command.

       ```
       $Credential = Get-Credential
       ```

     - To generate the credential object using an AWS Secrets Manager resource, use the following command.

       ```
       $Secret = ConvertFrom-Json -InputObject (Get-SECSecretValue -SecretId
        $AdminSecret).SecretString
       $Credential = (New-Object PSCredential($Secret.UserName,(ConvertTo-SecureString
        $Secret.Password -AsPlainText -Force)))
       ```

   - DNS server IP addresses (*IP_ADDRESS_1*, *IP_ADDRESS_2*)
   - Subnet ID(s) for subnets where you plan to create your Amazon FSx file system (*SUBNET_1*, *SUBNET_2*, for example, `subnet-04431191671ac0d19`).

```
PS C:\>
$FSxADValidationArgs = @{
    # DNS root of ActiveDirectory domain
    DomainDNSRoot = 'DOMAINNAME.COM'

    # IP v4 addresses of DNS servers
    DnsIpAddresses = @('IP_ADDRESS_1', 'IP_ADDRESS_2')

    # Subnet IDs for Amazon FSx file server(s)
    SubnetIds = @('SUBNET_1', 'SUBNET_2')

    Credential = $Credential
}
```

9. (Optional) Set Organizational Unit, Delegated Administrators group, DomainControllersMaxCount, and enable service account permission validation by following instructions in the included `README.md` file prior to running the validation tool.

   Note
   The Builtin `Domain Admins` group has a different name if the operating system is not in English. For example, the group is named `Administrateurs du domaine` in the French OS version. If you don't specify a value, the default `Domain Admins` group name is used and the file system creation fails.

10. Run the validation tool by using this command.

```
PS C:\> $Result = Test-FSxADConfiguration @FSxADValidationArgs
```

11. The following is an example of a successful test result.

```
Test 1 - Validate EC2 Subnets ...
...
Test 17 - Validate 'Delete Computer Objects' permission ...

Test computer object amznfsxtestd53f deleted!
```

```
...
SUCCESS - All tests passed! Please proceed to creating an Amazon FSx file system.
 For your convenience, SelfManagedActiveDirectoryConfiguration of result can be used
 directly in CreateFileSystemWindowsConfiguration for New-FSXFileSystem
PS C:\AmazonFSxADValidation> $Result.Failures.Count
0
PS C:\AmazonFSxADValidation> $Result.Warnings.Count
0
```

The following is an example of a test result with errors.

```
Test 1 - Validate EC2 Subnets ...
...
Test 7 - Validate that provided EC2 Subnets belong to a single AD Site ...

Name          DistinguishedName
 Site
----          -----------------
 ----
10.0.0.0/19   CN=10.0.0.0/19,CN=Subnets,CN=Sites,CN=Configuration,DC=test-ad,DC=local
 CN=SiteB,CN=Sites,CN=Configu...
10.0.128.0/19 CN=10.0.128.0/19,CN=Subnets,CN=Sites,CN=Configuration,DC=test-ad,DC=local
 CN=Default-First-Site-Name,C...
10.0.64.0/19  CN=10.0.64.0/19,CN=Subnets,CN=Sites,CN=Configuration,DC=test-ad,DC=local
 CN=SiteB,CN=Sites,CN=Configu...



Best match for EC2 subnet subnet-092f4caca69e360e7 is AD site CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=te
st-ad,DC=local
Best match for EC2 subnet subnet-04431191671ac0d19 is AD site
 CN=SiteB,CN=Sites,CN=Configuration,DC=test-ad,DC=local
WARNING: EC2 subnets subnet-092f4caca69e360e7 subnet-04431191671ac0d19 matched to
 different AD sites! Make sure they
are in a single AD site.
...
9 of 16 tests skipped.
FAILURE - Tests failed. Please see error details below:

Name                          Value
----                          -----
SubnetsInSeparateAdSites      {subnet-04431191671ac0d19, subnet-092f4caca69e360e7}



Please address all errors and warnings above prior to re-running validation to confirm
 fix.
PS C:\AmazonFSxADValidation> $Result.Failures.Count
1
PS C:\AmazonFSxADValidation> $Result.Failures

Name                          Value
----                          -----
SubnetsInSeparateAdSites      {subnet-04431191671ac0d19, subnet-092f4caca69e360e7}


PS C:\AmazonFSxADValidation> $Result.Warnings.Count
0
```

If you receive warnings or errors when you run the validation tool, refer to the Troubleshooting guide included in the validation tool package (`TROUBLESHOOTING.md`) and Troubleshooting Amazon FSx (p. 204).

Joining an Amazon FSx file system to a self-managed Microsoft Active Directory domain

When you create a new FSx for Windows File Server file system, you can configure Microsoft Active Directory integration so that it joins to your self-managed Microsoft Active Directory domain. To do this, provide the following information for your Microsoft AD:

- The fully qualified domain name of your on-premises Microsoft AD directory.

  Note
  Amazon FSx currently does not support Single Label Domain (SLD) domains.
- The IP addresses of the DNS servers for your domain.
- Credentials for a service account in your on-premises Microsoft AD domain. Amazon FSx uses these credentials to join to your self-managed AD.

Optionally, you can also specify the following:

- A specific Organizational Unit (OU) within the domain that you want your Amazon FSx file system to join to.
- The name of the domain group whose members are granted administrative privileges for the Amazon FSx file system.

After you specify this information, Amazon FSx joins your new file system to your self-managed AD domain using the service account that you provided.

Important
Amazon FSx only registers DNS records for a file system if the AD domain that you are joining it to is using Microsoft DNS as the default DNS. If you are using a third-party DNS, you will need to manually setup DNS entries for your Amazon FSx file systems after you create your file system. For more information on choosing the correct IP addresses to use for the file system, see Obtaining the correct file system IP addresses to use for DNS (p. 49).

Before you begin

Make sure that you have completed the Prerequisites for using a self-managed Microsoft AD (p. 34) detailed in Using Amazon FSx with your self-managed Microsoft Active Directory (p. 33).

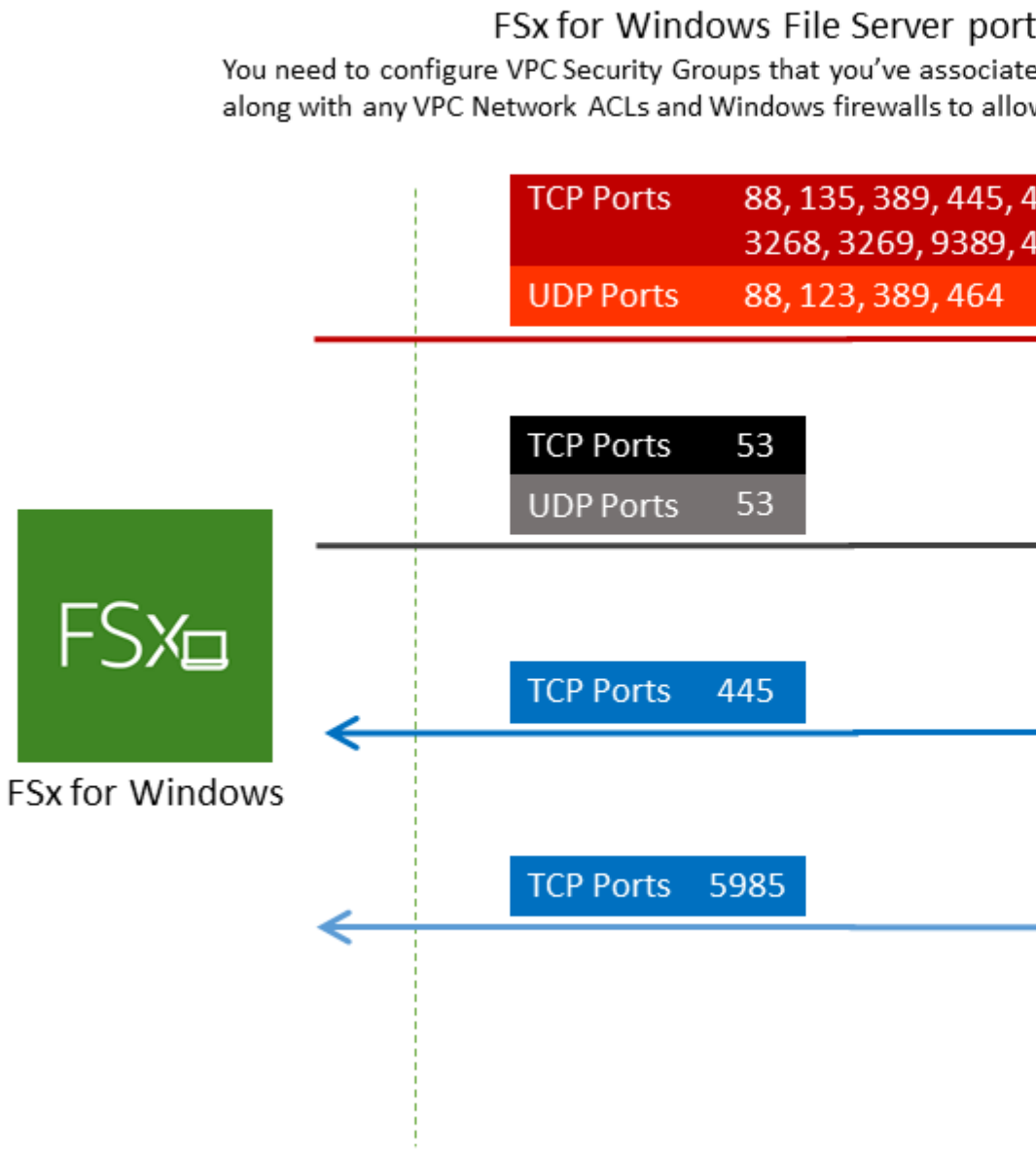
To create an FSx for Windows File Server file system joined to a self-managed AD (Console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the dashboard, choose **Create file system** to start the file system creation wizard.
3. Choose **FSx for Windows File Server** and then choose **Next**. The **Create file system page appears.**
4. Provide a name for your file system. You can use a maximum of 256 Unicode letters, white space, and numbers, plus the special characters + - = . _ : /
5. For **Storage capacity**, enter the storage capacity of your file system, in GiB. If you're using SSD storage, enter any whole number in the range of 32–65,536. If you're using HDD storage, enter any whole number in the range of 2,000–65,536. You can increase the amount of storage capacity as needed at any time after you create the file system. For more information, see Managing storage capacity (p. 123).
6. Keep **Throughput capacity** at its default setting. **Throughput capacity** is the sustained speed at which the file server that hosts your file system can serve data. The **Recommended throughput capacity** setting is based on the amount of storage capacity you choose. If you need more than the

recommended throughput capacity, choose **Specify throughput capacity**, and then choose a value. For more information, see FSx for Windows File Server performance (p. 153).

You can modify the throughput capacity as needed at any time after you create the file system. For more information, see Managing throughput capacity (p. 133).

7. Choose the VPC that you want to associate with your file system. For the purposes of this getting started exercise, choose the same VPC as for your AWS Directory Service directory and Amazon EC2 instance.
8. Choose any value for **Availability Zones** and **Subnet**.
9. For **VPC security groups**, the default security group for your default Amazon VPC is already added to your file system in the console. Please ensure that the security group and the VPC Network ACLs for the subnet(s) where you're creating your FSx file system allow traffic on the ports and and in the directions shown in the following diagram.



The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Kerberos authenticatio
TCP/UDP	464	Change/ Set password
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
UDP	123	Network Time Protocol (NTP)
TCP	135	Distributed Computing Environment End Point Mapper (DCE / EPMAP)
TCP	445	Directory Services SMB file sharing
TCP	636	Lightweight Directory Access Protocol over TLS/ SSL (LDAPS)
TCP	3268	Microsoft Global Catalog

Protocol	Ports	Role
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management
TCP	9389	Microsoft AD DS Web Services, PowerShell
TCP	49152 - 65535	Ephemeral ports for RPC

Important
Allowing outbound traffic on TCP port 9389 is required for Single-AZ 2 and all Multi-AZ file system deployments.

Note
If you're using VPC network ACLs, you must also allow outbound traffic on dynamic ports (49152-65535) from your FSx file system.

- Outbound rules to allow all traffic to the IP addresses associated with the DNS servers and domain controllers for your self-managed Microsoft AD domain. For more information, see Microsoft's documentation on configuring your firewall for Active Directory communication.
- Ensure that these traffic rules are also mirrored on the firewalls that apply to each of the AD domain controllers, DNS servers, FSx clients and FSx administrators.

Note
If you have Active Directory sites defined, you must ensure that the subnet(s) in the VPC associated with your Amazon FSx file system are defined in an Active Directory site, and that no conflicts exist between the subnet(s) in your VPC and the subnets in your other sites. You can view and change these settings using the Active Directory Sites and Services MMC snap-in.

Important
While Amazon VPC security groups require ports to be opened only in the direction that network traffic is initiated, most Windows firewalls and VPC network ACLs require ports to be open in both directions.

10. For **Windows authentication**, choose **Self-managed Microsoft Active Directory**.
11. Enter a value for **Fully qualified domain name** for the self-managed Microsoft AD directory.

Note
Domain name must not be in the Single Label Domain (SLD) format. Amazon FSx currently does not support SLD domains.

Important
For Single-AZ 2 and all Multi-AZ file systems, the Active Directory domain name cannot exceed 47 characters.

12. Enter a value for **Organizational Unit** for the self-managed Microsoft AD directory.

    Note
    Ensure that the service account you provided has permissions delegated to the OU that you specify here or to the default OU if you don't specify one.

13. Enter at least one, and no more than two, values for **DNS Server IP Addresses** for the self-managed Microsoft AD directory.
14. Enter a string value for **Service account username** for the account on your self-managed AD domain, such as `ServiceAcct`. Amazon FSx uses this user name to join to your Microsoft AD domain.

    Important
    DO NOT include a domain prefix (`corp.com\ServiceAcct`) or domain suffix (`ServiceAcct@corp.com`) when entering the **Service account username**.
    DO NOT use the Distinguished Name (DN) when entering the **Service account username** (`CN=ServiceAcct,OU=example,DC=corp,DC=com`).

15. Enter a value for **Service account password** for the account on your self-managed AD domain. Amazon FSx uses this password to join to your Microsoft AD domain.
16. Re-enter the password to confirm it in **Confirm password**.
17. For **Delegated file system administrators group**, specify the `Domain Admins` group or a custom delegated file system administrators group (if you've created one). The group you specify should have the delegated authority to perform administrative tasks on your file system. If you don't provide a value, Amazon FSx uses the Builtin `Domain Admins` group.

    Important
    If you do not provide a **Delegated file system administrators group**, by default Amazon FSx attempts to use the Builtin Domain Admins group in your AD domain. If the name of this Builtin group has been changed or if you're using a different group for domain administration, you must provide that name for the group here.

    Important
    DO NOT include a domain prefix (corp.com\FSxAdmins) or domain suffix (FSxAdmins@corp.com) when providing the group name parameter.
    DO NOT use the Distinguished Name (DN) for the group. An example of a distinguished name is CN=FSxAdmins,OU=example,DC=corp,DC=com.

To create an FSx for Windows File Server file system joined to a self-managed AD (AWS CLI)

The following example creates an FSx for Windows File Server file system with a `SelfManagedActiveDirectoryConfiguration` in the `us-east-2` Availability Zone.

```
aws fsx --region us-east-2 \
create-file-system \
--file-system-type WINDOWS \
--storage-capacity 300 \
--security-group-ids security-group-id \
--subnet-ids subnet-id\
--windows-configuration
 SelfManagedActiveDirectoryConfiguration='{DomainName="corp.example.com", \
```

```
OrganizationalUnitDistinguishedName="OU=FileSystems,DC=corp,DC=example,DC=com",FileSystemAdministrators
 \
UserName="FSxService",Password="password", \
   DnsIps=["10.0.1.18"]}',ThroughputCapacity=8
```

Important
Do not move computer objects that Amazon FSx creates in the OU after your file system is created. Doing so will cause your file system to become misconfigured.

Obtaining the correct file system IP addresses to use for DNS

Amazon FSx only registers DNS records for a file system if you are using Microsoft DNS as the default DNS service. If you are using a third-party DNS, you will need to manually setup DNS entries for your Amazon FSx file systems. This section describes how to obtain the correct file system IP addresses to use if you have to manually add the file system to your DNS.

How to obtain file system IP addresses to use for DNS A entries

1. In the https://console.aws.amazon.com/fsx/, choose the file system that you want to obtain the IP address of to display the file system details page.
2. In the **Network & security** tab do one of the following:

   - For Single-AZ 1 file systems:
     - In the **Subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 console.
     - The IP address for the Single-AZ 1 file system to use is shown in the **Primary private IPv4 IP** column.
   - For Single-AZ 2 or Multi-AZ file systems:
     - In the **Preferred subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 console.
     - The IP address for the preferred subnet to use is shown in the **Secondary private IPv4 IP** column.
     - In the Amazon FSx **Standby subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 console.
     - The IP address for the standby subnet to use is shown in the **Secondary private IPv4 IP** column.

Updating the self-managed Active Directory configuration

You can use the AWS Management Console, Amazon FSx API, or AWS CLI to update the username and password for the service account and the IP addresses for the Active Directory DNS servers of the self-managed Active Directory configuration. You can track the progress of a self-managed Active Directory configuration update at any time using the AWS Management Console, CLI, and API. For more information, see Monitoring self-managed Active Directory updates (p. 50).

To update the self-managed Active Directory configuration (Console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system for which you want to update self-managed AD configuration.

3. In the **Network & security** tab, then choose **Update** for the **DNS server IP addresses**, or for the service account username, depending on which Active Directory properties you are updating.
4. Enter the new DNS server IP addresses, or the new service account credentials in the dialog that appears.
5. Choose **Update** to initiate the Active Directory configuration update.

   You can monitor the update progress (p. 50) using the AWS Management Console or the AWS CLI.

To update the self-managed Active Directory configuration (CLI)

- To update the self-managed Active Directory configuration of an FSx for Windows File Server file system, use the AWS CLI command update-file-system. Set the following parameters:
  - `--file-system-id` to the ID of the file system you are updating.
  - `UserName` the new username for the self-managed AD service account.
  - `Password` the new password for the self-managed AD service account.
  - `DnsIps` the IP addresses for the self-managed AD DNS servers.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --windows-configuration
 SelfManagedActiveDirectoryConfiguration={UserName=username,Password=password,DnsIps=[192.0.2.0,192
```

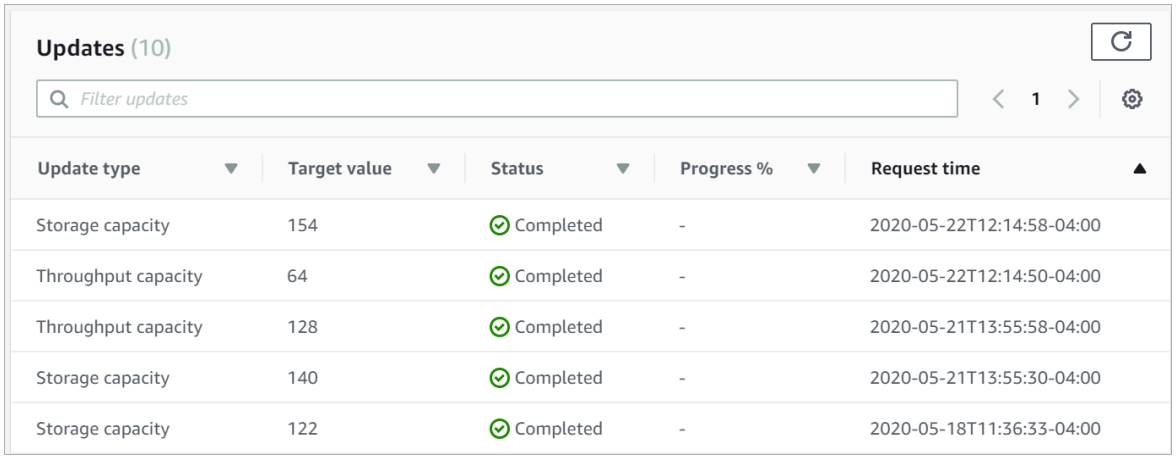
If the update action is successful, the service sends back an HTTP 200 response. The AdminstrativeActions data in the response describes the request and its status. For more information, see Monitoring self-managed Active Directory updates (p. 50).

Monitoring self-managed Active Directory updates

You can monitor the progress of a self-managed Active Directory configuration update using the AWS Management Console, the API, or the AWS CLI.

Monitoring updates in the console

In the **Updates** tab in the **File system details** window, you can view the 10 most recent updates for each update type.



For self-managed Active Directory updates, you can view the following information.

Update type

Supported types are as follows:

- DNS server IP address
- Service account credentials

Target value

The desired value to update the file system property to. For **Service account credentials** updates, only the user name is shown, service account passwords are never included in this field.

Status

The current status of the update. For self-managed Active Directory updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Completed** – The file system update completed successfully.
- **Failed** – The file system update failed. Choose the question mark (**?**) to see details about the failure.

Progress %

Displays the progress of the file system update as percent complete.

Request time

The time that Amazon FSx received the update action request.

Monitoring updates using the AWS CLI and API

You can view and monitor file system update requests that are in progress using the describe-file-systems AWS CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type.

The following example shows an excerpt of the response of a **describe-file-systems** CLI command show two self-managed AD file system updates.

```
{
    "OwnerId": "111122223333",
    .
    .
    .
    "StorageCapacity": 1000,
    "AdministrativeActions": [
        {
            "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
            "RequestTime": 1581694766.757,
            "Status": "PENDING",
            "TargetFileSystemValues": {
                "WindowsConfiguration": {
                    "SelfManagedActiveDirectoryConfiguration": {
                        "UserName": "serviceUser",
                    }
                }
            }
        },
        {
            "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
            "RequestTime": 1619032957.759,
            "Status": "FAILED",
```

```
                    "TargetFileSystemValues": {
                        "WindowsConfiguration": {
                            "SelfManagedActiveDirectoryConfiguration": {
                            "DnsIps": [
                                    "10.0.138.161"
                                ]
                            }
                        }
                    },
                    "FailureDetails": {
                        "Message": "Failure details message."
                    }
                }
            ],
    .
    .
    .
```

Using Microsoft Windows file shares

A Microsoft Windows *file share* is a specific folder in your file system. It includes that folder's subfolders, which you make accessible to your compute instances with the Server Message Block (SMB) protocol. Your file system comes with a default Windows file share, named `share`. You can create and manage as many other Windows file shares as you want by using the Windows graphical user interface (GUI) tool named *Shared Folders*.

Accessing file shares

To access your file shares, you use the Windows Map Network Drive functionality to map a drive letter on your compute instance to your Amazon FSx file share. The process of mapping a file share to a drive on your compute instance is known as *mounting* a file share in Linux. This process differs depending on the type of compute instance and the operating system. After your file share is mapped, your applications and users can access files and folders on your file share as if they are local files and folders.

Following are procedures for mapping a file share on the different supported compute instances.

Topics

Mapping a file share on an Amazon EC2 Windows instance

You can map a file share on an EC2 Windows instance by using the Windows File Explorer or the command prompt.

To map a file share on an Amazon EC2 Windows instance (console)

1. Launch the EC2 Windows instance and connect it to the Microsoft Active Directory that you joined your Amazon FSx file system to. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:
   - Seamlessly join a Windows EC2 instance
   - Manually join a Windows instance
2. Connect to your EC2 Windows instance. For more information, see Connecting to your Windows instance in the *Amazon EC2 User Guide for Windows Instances*.
3. After you're connected, open File Explorer.
4. In the navigation pane, open the context (right-click) menu for **Network**, and choose **Map Network Drive**.

5. For **Drive**, choose a drive letter.
6. For **Folder**, enter either the file system's DNS name or a DNS alias associated with the file system, and the share name.

   You can find the file system's DNS name and any associated DNS aliases on the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of the CreateFileSystem or DescribeFileSystems API operation. For more information about using DNS aliases, see Managing DNS aliases (p. 90).

   - For a Single-AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following.

     ```
     fs-0123456789abcdef0.ad-domain.com
     ```

   - For a Single-AZ file system joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following.

     ```
     amznfsxaa11bb22.ad-domain.com
     ```

   For example, to use a Single-AZ file system's DNS name, enter the following for **Folder**.

   ```
   \\fs-0123456789abcdef0.ad-domain.com\share
   ```

   To use a Multi-AZ file system's DNS name, enter the following for **Folder**.

   ```
   \\famznfsxaa11bb22.ad-domain.com\share
   ```

   To use a DNS alias associated with the file system, enter the following for **Folder**.

   ```
   \\fqdn-dns-alias\share
   ```

7. Choose an option for **Reconnect at sign-in**, which indicates whether the file share should reconnect at sign-in, and then choose **Finish**.

To map a file share on an Amazon EC2 Windows instance (command prompt)

1. Launch the EC2 Windows instance and connect it to the Microsoft Active Directory that you joined your Amazon FSx file system to. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:

   - Seamlessly join a Windows EC2 instance
   - Manually join a Windows instance

2. Connect to your EC2 Windows instance as a user in your AWS Managed Microsoft AD directory. For more information, see Connecting to your Windows instance in the *Amazon EC2 User Guide for Windows Instances*.
3. After you're connected, open a command prompt window.
4. Mount the file share using a drive letter of your choice, the file system's DNS name, and the share name. You can find the DNS name using the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of the `CreateFileSystem` or `DescribeFileSystems` API operation.

   - For a Single-AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following.

```
fs-0123456789abcdef0.ad-domain.com
```

- For a Single-AZ file system joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following.

```
amznfsxaa11bb22.ad-domain.com
```

The following is an example command to mount the file share.

```
$ net use H: \\amzfsxaa11bb22.ad-domain.com\share /persistent:yes
```

Instead of the `net use` command, you can also use any supported PowerShell command to mount a file share.

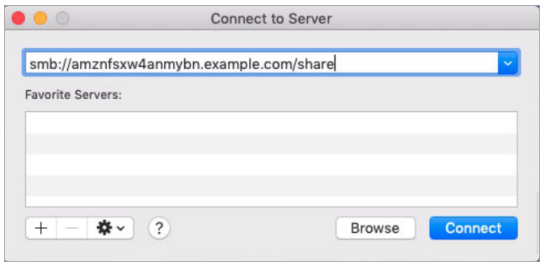
Mounting a file share on an Amazon EC2 Mac instance

You can mount a file share on an Amazon EC2 Mac instance that is either joined to your Active Directory or not joined. If the instance is not joined to your Active Directory, be sure to update the DHCP options set for the Amazon Virtual Private Cloud (Amazon VPC) in which the instance resides to include the DNS name servers for your Active Directory domain. Then relaunch the instance.

To mount a file share on an Amazon EC2 Mac instance (GUI)

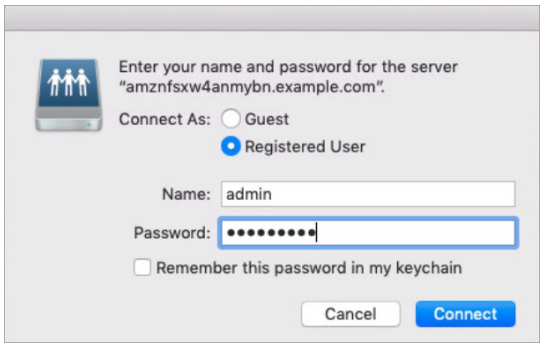
1. Launch the EC2 Mac instance. To do this, choose one of the following procedures from the *Amazon EC2 User Guide for Linux Instances*:
   - Launch a Mac instance using the console
   - Launch a Mac instance using the AWS CLI
2. Connect to your EC2 Mac instance using Virtual Network Computing (VNC). For more information, see Connect to your instance using VNC in the *Amazon EC2 User Guide for Linux Instances*.
3. On your EC2 Mac instance, connect to your Amazon FSx file share, as follows:
   a. Open Finder, choose **Go**, and then choose **Connect to Server**.
   b. In the **Connect to Server** dialog box, enter either the file system's DNS name or a DNS alias associated with the file system, and the share name. Then choose **Connect**.

      You can find the file system's DNS name and any associated DNS aliases on the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of the CreateFileSystem or DescribeFileSystems API operation. For more information about using DNS aliases, see Managing DNS aliases (p. 90).



c. On the next screen, choose **Connect** to continue.

d. Enter your Microsoft Active Directory (AD) credentials for the Amazon FSx service account, as shown in the following example. Then choose **Connect**.



e. If the connection is successful, you can see the Amazon FSx share, under **Locations** in your Finder window.

To mount a file share on an Amazon EC2 Mac instance (command line)

1. Launch the EC2 Mac instance. To do this, choose one of the following procedures from the *Amazon EC2 User Guide for Linux Instances*:

   - Launch a Mac instance using the console
   - Launch a Mac instance using the AWS CLI

2. Connect to your EC2 Mac instance using Virtual Network Computing (VNC). For more information, see Connect to your instance using VNC in the *Amazon EC2 User Guide for Linux Instances*.

3. Mount the file share with the following command.

```
mount_smbfs //file_system_dns_name/file_share mount_point
```

You can find the DNS name on the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of the `CreateFileSystem` or `DescribeFileSystems` API operation.

- For a Single-AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following.

```
fs-0123456789abcdef0.ad-domain.com
```

- For a Single-AZ file system joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following.

```
amznfsxaa11bb22.ad-domain.com
```

The mount command used in this procedure does the following at the given points:

- *//file_system_dns_name/file_share* – Specifies the DNS name and share of the file system to mount.
- *mount_point* – The directory on the EC2 instance that you are mounting the file system to.

Mounting a file share on an Amazon EC2 Linux instance

You can mount an FSx for Windows File Server file share on an Amazon EC2 Linux instance that is either joined to your Active Directory or not joined.

> **Note**
> The following commands specify parameters such as SMB protocol, caching, and read and write buffer size as examples only. Parameter choices for the Linux `cifs` command, as well as the Linux kernel version used, can impact throughput and latency for network operations between the client and the Amazon FSx file system. For more information, see `cifs` documentation for the Linux environment you are using.

To mount a file share on an Amazon EC2 Linux instance joined to your Active Directory

1. If you don't already have a running EC2 Linux instance joined to your Microsoft Active Directory, see Manually join a Linux instance in the *AWS Directory Service Administration Guide* for the instructions to do so.
2. Connect to your EC2 Linux instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*.
3. Run the following command to install the `cifs-utils` package. This package is used to mount network file systems like Amazon FSx on Linux.

```
$ sudo yum install cifs-utils
```

4. Create the mount point directory **`/mnt/fsx`**. This is where you will mount the Amazon FSx file system.

```
$ sudo mkdir -p /mnt/fsx
```

5. Authenticate with kerberos using the following command.

```
$ kinit
```

6. Mount the file share with the following command.

```
$ sudo mount -t cifs //file_system_dns_name/file_share mount_point --verbose -o
 vers=SMB_version,sec=krb5,cruid=ad_user,rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize,cache=none,ip=pr
file-server-Ip
```

   You can find the DNS name on the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of `CreateFileSystem` or `DescribeFileSystems` API operation.

   - For a Single-AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks like the following.

     ```
     fs-0123456789abcdef0.ad-domain.com
     ```

   - For a Single-AZ file system joined to a self-managed Active Directory, and any Multi-AZ file system, the DNS name looks like the following.

     ```
     amznfsxaa11bb22.ad-domain.com
     ```

Replace *CIFSMaxBufSize* with the largest value allowed by your kernel. Run the following command to get this value.

```
$ modinfo cifs | grep CIFSMaxBufSize
parm:           CIFSMaxBufSize:Network buffer size (not including header). Default:
 16384 Range: 8192 to 130048 (uint)
```

The output shows that the maximum buffer size is 130048.

7. Verify that the file system is mounted by running the following command, which returns only file systems of the Common Internet File System (CIFS) type.

```
$ mount -l -t cifs
//fs-0123456789abcdef0/share on /mnt/fsx type cifs
 (rw,relatime,vers=SMB_version,sec=krb5,cache=cache_mode,username=user1@CORP.NETWORK.COM,uid=0,nofo
```

The mount command used in this procedure does the following at the given points:

- `//`*file_system_dns_name*`/`*file_share* – Specifies the DNS name and share of the file system to mount.
- *mount_point* – The directory on the EC2 instance that you are mounting the file system to.
- `-t cifs vers=`*SMB_version* – Specifies the type of file system as CIFS and the SMB protocol version. Amazon FSx for Windows File Server supports SMB versions 2.0 through 3.1.1.
- `sec=krb5` – Specifies to use Kerberos version 5 for authentication.
- `cache=`*cache_mode* – Sets the cache mode. This option for CIFS cache can impact performance, and you should test which settings work best (and review Linux documentation) for your kernel and workload. Options `strict` and `none` are recommended, because `loose` can cause data inconsistency due to the looser protocol semantics.
- `cruid=`*ad_user* – Sets the uid of the owner of the credentials cache to the AD directory administrator.
- */mnt/fsx* – Specifies the mount point for the Amazon FSx file share on your EC2 instance.
- `rsize=`*CIFSMaxBufSize*`,wsize=`*CIFSMaxBufSize* – Specifies the read and write buffer size as the maximum allowed by the CIFS protocol. Replace *CIFSMaxBufSize* with the largest value allowed by your kernel. Determine the `CIFSMaxBufSize` by running the following command.

```
$ modinfo cifs | grep CIFSMaxBufSize
parm:           CIFSMaxBufSize:Network buffer size (not including header). Default: 16384
 Range: 8192 to 130048 (uint)
```

The output shows that the maximum buffer size is 130048.

- `ip=`*preferred-file-server-Ip* – Sets the destination IP address to that of the file system's preferred file server.

  You can retrieve the file system's preferred file server IP address as follows:

  - Using the Amazon FSx console, on the **Network & security** tab of the **File system details** page.
  - In the response of the `describe-file-systems` CLI command or the equivalent DescribeFileSystems API command.

To Mount a File Share on an Amazon EC2 Linux Instance Not Joined to Your Active Directory

The following procedure mounts an Amazon FSx file share to an Amazon EC2 Linux instance that is not joined to your Active Directory (AD). For an EC2 Linux instance that is *not* joined to your Active Directory, you can only mount an FSx for Windows File Server file share by using its private IP address. You can get the file system's private IP address using the Amazon FSx console, on the **Network & security** tab, in **Preferred File Server IP Address**.

This example uses NTLM authentication. To do this, you mount the file system as a user that is a member of the Microsoft Active Directory domain that the FSx for Windows File Server file system is joined to. The credentials for the user account are provided in a text file that you create on your EC2 instance, `creds.txt`. This file contains the user name, password, and domain for the user.

```
$ cat creds.txt
username=user1
password=Password123
domain=EXAMPLE.COM
```

To launch and configure the Amazon Linux EC2 instance

1. Launch an Amazon Linux EC2 instance using the Amazon EC2 console. For more information, see Launch an instance in the *Amazon EC2 User Guide for Linux Instances*.
2. Connect to your Amazon Linux EC2 instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*.
3. Run the following command to install the `cifs-utils` package. This package is used to mount network file systems like Amazon FSx on Linux.

   ```
   $ sudo yum install cifs-utils
   ```
4. Create the mount point **`/mnt/fsxx`** where you plan to mount the Amazon FSx file system.

   ```
   $ sudo mkdir -p /mnt/fsx
   ```
5. Create the `creds.txt` credentials file in the `/home/ec2-user` directory, using the format shown previously.
6. Set the `creds.txt` file permissions so that only you (the owner) can read and write to the file by running the following command.

   ```
   $ chmod 700 creds.txt
   ```

To mount the file system

1. You mount a file share not joined to your Active Directory by using its private IP address. You can get the file system's private IP address using the Amazon FSx console, on the **Network & security** tab, in the **Preferred File Server IP Address**.
2. Mount the file system using the following command:

   ```
   $ sudo mount -t cifs //file-system-IP-address/file_share /mnt/fsx
    --verbose -o vers=SMB_version,sec=ntlmsspi,cred=/home/ec2-user/
   creds.txt,rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize,cache=none
   ```

   Replace *`CIFSMaxBufSize`* with the largest value allowed by your kernel. Run the following command to get this value.

```
$ modinfo cifs | grep CIFSMaxBufSize
parm:           CIFSMaxBufSize:Network buffer size (not including header). Default:
 16384 Range: 8192 to 130048 (uint)
```

The output shows that the maximum buffer size is 130048.

3. Verify that the file system is mounted by running the following command, which returns only CIFS file systems.

```
$ mount -l -t cifs
//file-system-IP-address/file_share on /mnt/fsx type cifs
 (rw,relatime,vers=SMB_version,sec=ntlmsspi,cache=cache_mode,username=user1,domain=CORP.EXAMPLE.COM
```

The mount command used in this procedure does the following at the given points:

- `//file-system-IP-address/file_share` – Specifies the IP address and share of the file system to mount.
- `-t cifs vers=SMB_version` – Specifies the type of file system as CIFS and the SMB protocol version. Amazon FSx for Windows File Server supports SMB versions 2.0 through 3.1.1.
- `sec=ntlmsspi` – Specifies to use NT LAN Manager Security Support Provider Interface (NTLMSSPI) for authentication.
- `cache=cache_mode` – Sets the cache mode. This option for CIFS cache can impact performance, and you should test which settings work best (and review Linux documentation) for your kernel and workload. Options `strict` and `none` are recommended, because `loose` can cause data inconsistency due to the looser protocol semantics.
- `cred=/home/ec2-user/creds.txt` – Specifies where to get the user credentials.
- `/mnt/fsx` – Specifies the mount point for the Amazon FSx file share on your EC2 instance.
- `rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize` – Specifies the read and write buffer size as the maximum allowed by the CIFS protocol. Replace `CIFSMaxBufSize` with the largest value allowed by your kernel. Determine the `CIFSMaxBufSize` by running the following command.

```
$ modinfo cifs | grep CIFSMaxBufSize
parm:           CIFSMaxBufSize:Network buffer size (not including header). Default: 16384
 Range: 8192 to 130048 (uint)
```

The output shows that the maximum buffer size is 130048.

Automatically mounting file shares on an Amazon Linux EC2 instance not joined to your Active Directory

You can automatically mount your FSx for Windows File Server file share whenever the Amazon EC2 Linux instance to which it's mounted reboots. To do so, add an entry to the `/etc/fstab` file on the EC2 instance. The `/etc/fstab` file contains information about file systems. The command **mount -a**, which runs during instance startup, mounts the file systems listed in the `/etc/fstab` file.

For an Amazon Linux EC2 instance that is not joined to your Active Directory, you can only mount an FSx for Windows File Server file share by using its private IP address. You can get the file system's private IP address using the Amazon FSx console, on the **Network & security** tab, in **Preferred File Server IP Address**.

The following procedure uses Microsoft NTLM authentication. You mount the file system as a user that is a member of the Microsoft Active Directory domain to which the FSx for Windows File Server file system is joined. The credentials for the user account are provided in the text file `creds.txt`. This file contains the user name, password, and domain for the user.

```
$ cat creds.txt
username=user1
password=Password123
domain=EXAMPLE.COM
```

To automatically mount a file share on an Amazon Linux EC2 instance not joined to your Active Directory

To launch and configure the Amazon Linux EC2 instance

1. Launch an Amazon Linux EC2 instance using the Amazon EC2 console. For more information, see Launch an instance in the *Amazon EC2 User Guide for Linux Instances*.
2. Connect to your instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*.
3. Run the following command to install the `cifs-utils` package. This package is used to mount network file systems like Amazon FSx on Linux.

   ```
   $ sudo yum install cifs-utils
   ```

4. Create the `/mnt/fsx` directory. This is where you will mount the Amazon FSx file system.

   ```
   $ sudo mkdir /mnt/fsx
   ```

5. Create the `creds.txt` credentials file in the `/home/ec2-user` directory.
6. Set the file permissions so that only you (the owner) can read the file by running the following command.

   ```
   $ sudo chmod 700 creds.txt
   ```

To automatically mount the file system

1. You automatically mount a file share not joined to your Active Directory by using its private IP address. You can get the file system's private IP address from the Amazon FSx console, in the **Network & security** tab, the **Preferred File Server IP Address**.
2. To automatically mount the file share using its private IP address, add the following line to the `/etc/fstab` file.

   ```
   //file-system-IP-address/file_share /mnt/fsx cifs vers=SMB_version,sec=ntlmsspi,cred=/
   home/ec2-user/creds.txt,rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize,cache=none
   ```

   Replace *`CIFSMaxBufSize`* with the largest value allowed by your kernel. Run the following command to get this value.

   ```
   $ modinfo cifs | grep CIFSMaxBufSize
   parm:           CIFSMaxBufSize:Network buffer size (not including header). Default:
    16384 Range: 8192 to 130048 (uint)
   ```

   The output shows that the maximum buffer size is 130048.

3. Test the `fstab` entry by using the `mount` command with the 'fake' option in conjunction with the 'all' and 'verbose' options.

```
$ sudo mount -fav
home/ec2-user/fsx      : successfully mounted
```

4. To mount the file share, reboot the Amazon EC2 instance.
5. When the instance is available again, verify that the file system is mounted by running the following command.

```
$ sudo mount -l -t cifs
//file-system-IP-address/file_share on /mnt/fsx type cifs
 (rw,relatime,vers=SMB_version,sec=ntlmsspi,cache=cache_code,username=user1,domain=CORP.EXAMPLE.COM
```

The line added to the `/etc/fstab` file in this procedure does the following at the given points:

- `//file-system-IP-address/file_share` – Specifies the IP address and share of the Amazon FSx file system you're mounting.
- `/mnt/fsx` – Specifies the mount point for the Amazon FSx file system on your EC2 instance.
- `cifs vers=SMB_version` – Specifies the type of file system as CIFS and the SMB protocol version. Amazon FSx for Windows File Server supports SMB versions 2.0 through 3.1.1.
- `sec=ntlmsspi` – Specifies using NT LAN Manager Security Support Provider Interface to facilitate NTLM challenge-response authentication.
- `cache=cache_mode` – Sets the cache mode. This option for CIFS cache can impact performance, and you should test which settings work best (and review Linux documentation) for your kernel and workload. Options `strict` and `none` are recommended, because `loose` can cause data inconsistency due to the looser protocol semantics.
- `cred=/home/ec2-user/creds.txt` – Specifies where to get the user credentials.
- `_netdev` – Tells the operating system that the file system resides on a device that requires network access. Using this option prevents the instance from mounting the file system until the network service is enabled on the client.
- `0` – Indicates that the file system should be backed up by `dump`, if it's a nonzero value. For Amazon FSx, this value should be `0`.
- `0` – Specifies the order in which `fsck` checks file systems at boot. For Amazon FSx file systems, this value should be `0` to indicate that `fsck` shouldn't run at startup.

Migrating existing file storage to Amazon FSx

FSx for Windows File Server has the features, performance, and compatibility to help you easily lift and shift enterprise applications to the Amazon Web Services Cloud. The process of migrating to FSx for Windows File Server involves the following steps:

1. Migrate your files to FSx for Windows File Server. For more information, see Migrating existing file storage to FSx for Windows File Server (p. 63).
2. Migrate your file share configuration to FSx for Windows File Server. For more information, see Migrating file share configurations to Amazon FSx (p. 67).
3. Associate your existing DNS name as a DNS alias for your Amazon FSx file system. For more information, see Associating a DNS alias with Amazon FSx (p. 68).
4. Cut over to FSx for Windows File Server. For more information, see Cutting over to Amazon FSx (p. 71).

You can find the details for each step in the process in the following sections.

Topics

Migrating existing file storage to FSx for Windows File Server

To migrate your existing files to FSx for Windows File Server file systems, we recommend using AWS DataSync, an online data transfer service designed to simplify, automate, and accelerate copying large amounts of data to and from AWS storage services. DataSync copies data over the internet or AWS Direct Connect. As a fully managed service, DataSync removes much of the need to modify applications, develop scripts, or manage infrastructure. For more information, see Migrating existing files to FSx for Windows File Server using AWS DataSync (p. 64).

As an alternative solution, you can use Robust File Copy, or Robocopy, which is a command line directory and file replication command set for Microsoft Windows. For detailed procedures on how to use Robocopy to migrate file storage to FSx for Windows File Server, see Migrating existing files to FSx for Windows File Server using Robocopy (p. 65).

Best practices for migrating existing file storage to FSx for Windows File Server

To migrate large amounts of data to FSx for Windows File Server as quickly as possible, use Amazon FSx file systems configured with solid state drive (SSD) storage. After the migration is complete, you can move the data to Amazon FSx file systems using hard disk drive (HDD) storage if that is the best solution

for your application. To move data from an Amazon FSx file system using SDD storage to HDD storage you do the following:

1. Take a backup of your SSD file system. For more information, see Creating user-initiated backups (p. 79).
2. Restore the backup to a file system using HDD storage. For more information, see Restoring backups (p. 82).

Migrating existing files to FSx for Windows File Server using AWS DataSync

We recommend using AWS DataSync to transfer data between FSx for Windows File Server file systems. DataSync is a data transfer service that simplifies, automates, and accelerates moving and replicating data between on-premises storage systems and other AWS storage services over the internet or AWS Direct Connect. DataSync can transfer your file system data and metadata, such as ownership, timestamps, and access permissions.

DataSync supports copying NTFS access control lists (ACLs), and also supports copying file audit control information, also known as NTFS system access control lists (SACLs), which are used by administrators to control audit logging of user attempts to access files.

You can also use DataSync to transfer files between two FSx for Windows File Server file systems, including file systems in different AWS Regions and file systems owned by different AWS accounts. You can also use DataSync with FSx for Windows File Server file systems for other tasks. For example, you can perform one-time data migrations, periodically ingest data for distributed workloads, and schedule replication for data protection and recovery.

In AWS DataSync, a *location* for FSx for Windows File Server is an endpoint for an FSx for Windows File Server. You can transfer files between a location for FSx for Windows File Server and a location for other file systems. For information, see Working with Locations in the *AWS DataSync User Guide*.

DataSync accesses your FSx for Windows File Server using the Server Message Block (SMB) protocol. It authenticates with the user name and password that you configure in the AWS DataSync console or AWS CLI.

Prerequisites

To migrate data into your Amazon FSx for Windows File Server setup, you need a server and network that meet the DataSync requirements. To learn more, see Requirements for DataSync in the *AWS DataSync User Guide*.

If are using HDD storage and will be performing large data transfers with DataSync, we recommend that you switch to using SSD storage. For more information, see Best practices for migrating existing file storage to FSx for Windows File Server (p. 63).

When you have the DataSync requirements in place, you can begin transfer as discussed following.

Basic steps for migrating files using DataSync

To transfer files from a source location to a destination location using DataSync, take the following basic steps:

- Download and deploy an agent in your environment and activate it.
- Create and configure a source and destination location.
- Create and configure a task.

- Run the task to transfer files from the source to the destination.

To learn how to transfer files from an existing on-premises file system to your FSx for Windows File Server, see Data transfer between self-managed storage and AWS, Creating a location for SMB, and Creating a location for Amazon FSx for Windows File Server in the *AWS DataSync User Guide*.

To learn how to transfer files from an existing in-cloud file system to your FSx for Windows File Server, see Deploy your agent as an Amazon EC2 instance in the *AWS DataSync User Guide*.

Migrating existing files to FSx for Windows File Server using Robocopy

Built on Microsoft Windows Server, Amazon FSx for Windows File Server enables you to migrate your existing datasets fully into your Amazon FSx file systems. You can migrate the data for each file. You can also migrate all the relevant file metadata including attributes, timestamps, access control lists (ACLs), owner information, and auditing information. With this total migration support, Amazon FSx enables moving your Windows-based workloads and applications relying on these file datasets to the Amazon Web Services Cloud.

Use the following topics as a guide through the process for copying existing file data. As you perform this copy, you preserve all file metadata from your on-premises data centers or from your self-managed file servers on Amazon EC2.

Prerequisites

Before you begin, make sure that you do the following:

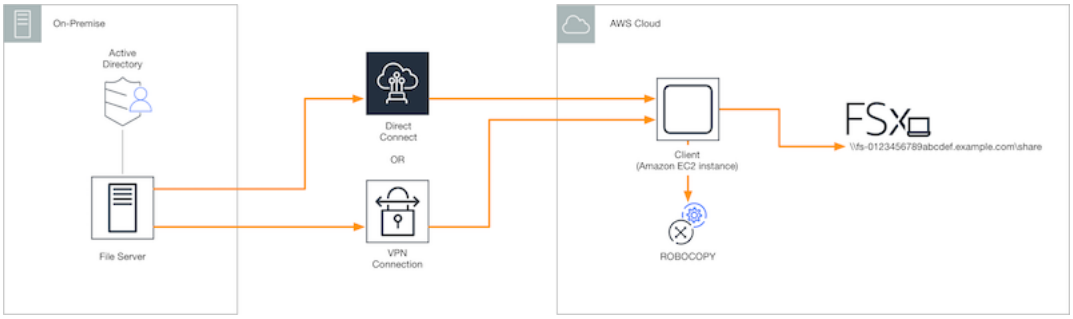
- Establish network connectivity (by using AWS Direct Connect or VPN) between your on-premises Active Directory and the VPC where you want to create the Amazon FSx file system.
- Create a service account on your Active Directory with delegated permissions to join computers to the domain. For more information, see Delegate Privileges to Your Service Account in the *AWS Directory Service Administration Guide*.
- Create an Amazon FSx file system, joined to your self-managed (on-premises) Microsoft AD directory.
- Note the location (for example, `\\Source\Share`) of the file share (either on-premises or in AWS) that contains the existing files you want to transfer over to Amazon FSx.
- Note the location (for example, `\\Target\Share`) of the file share on your Amazon FSx file system to which you want to transfer over your existing files.

The following table summarizes the source and destination file system accessibility requirements for three migration user access models.

Migration user access model	Source file system accessibility requirements	Destination FSx file server accessibility requirements
Direct read/write permissions model	The user needs to have at least read permissions (NTFS ACLs) on the files and folders being migrated.	The user needs to have at least write permissions (NTFS ACLs) on the files and folders being migrated.
Backup/restore privilege model to override access permissions	The user needs to be a member of the on-premises Active Directory's Backup Operators group, and use the /b flag with RoboCopy.	The user needs to be a member of the Amazon FSx file system's *administrators group**, and use the /b flag with RoboCopy.

Migration user access model	Source file system accessibility requirements	Destination FSx file server accessibility requirements
Domain administrator (full) privilege model to override access permissions	The user needs to be a member of the on-premises Active Directory's Domain Admins group.	The user needs to be a member of the Amazon FSx file system's *administrators group**, and use the /b flag with RoboCopy

Note
* For file systems joined to an AWS Managed Microsoft AD, the Amazon FSx file system administrators group is **AWS Delegated FSx Administrators**. In your self-managed Microsoft AD, the Amazon FSx file system administrators group is **Domain Admins** or the custom group that you specified for administration when you created your file system.



How to migrate existing files to Amazon FSx using Robocopy

You can migrate existing files to Amazon FSx by using the following procedure.

To migrate existing files to Amazon FSx

1. Launch a Windows Server 2016 Amazon EC2 instance in the same Amazon VPC as that of your Amazon FSx file system.
2. Connect to your Amazon EC2 instance. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Open **Command Prompt** and map the source file share on your existing file server (on-premises or in AWS) to a drive letter (for example, *Y*:) as follows. As part of this, you provide credentials for a member of your on-premises Active Directory's **Domain Administrators** group.

```
C:\>net use Y: \\fileserver1.mydata.com\localdata /user:mydata.com\Administrator
Enter the password for 'fileserver1.mydata.com': _

Drive Y: is now connected to \\fileserver1.mydata.com\localdata.

The command completed successfully.
```

4. Map the target file share on your Amazon FSx file system to a different drive letter (for example, *Z*:) on your Amazon EC2 instance as follows. As part of this, you provide credentials for a user account that is a member of your on-premises Active Directory's domain administrators group and your Amazon FSx file system's administrators group. For file systems joined to an AWS Managed Microsoft AD, that group is **`AWS Delegated FSx Administrators`**. In your self-managed Microsoft AD, that group is **`Domain Admins`** or the custom group that you specified for administration when you created your file system.

   For more information, see the table of source and destination file system accessibility requirements (p. 65) in the Prerequisites (p. 65).

```
C:\>net use Z: \\amznfsxabcdef1.mydata.com\share /user:mydata.com\Administrator
Enter the password for 'amznfsxabcdef1.mydata.com': _

Drive Z: is now connected to \\amznfsxabcdef1.mydata.com\share.

The command completed successfully.
```

5. Choose **Run as Administrator** from the context menu. Open **Command Prompt** or **Windows PowerShell** as an administrator, and run the following Robocopy command to copy the files from the source share to the target share.

   The `ROBOCOPY` command is a flexible file-transfer utility with multiple options to control the data transfer process. Because of this `ROBOCOPY` command execution, all the files and directories from the source share are copied to the Amazon FSx target share. The copy preserves file and folder NTFS ACLs, attributes, timestamps, owner information, and auditing information.

   ```
   robocopy Y:\ Z:\ /copy:DATSOU /secfix /e /b /MT:8
   ```

   The example command preceding uses the following elements and options:

   - Y – Refers to the source share located in the on-premises Active Directory forest mydata.com.
   - Z – Refers to the target share \\amznfsxabcdef1.mydata.com\share on Amazon FSx.
   - /copy – Specifies the following file properties to be copied:
     - D – data
     - A – attributes
     - T – timestamps
     - S – NTFS ACLs
     - O – owner information
     - U – auditing information.
   - /secfix – Fixes file security on all files, even skipped ones.
   - /e – Copies subdirectories, including empty ones.
   - /b – Uses the backup and restore privilege in Windows to copy files even if their NTFS ACLs deny permissions to the current user.
   - /MT:8 – Specifies how many threads to use for performing multithreaded copies.

   Note
   If you are copying large files over a slow or unreliable connection, you can enable restartable mode by using the **/zb** option with the **robocopy** in place of the **/b** option. With restartable mode, if the transfer of a large file is interrupted, a subsequent Robocopy operation can pick up in the middle of the transfer instead of having to re-copy the entire file from the beginning. Enabling restartable mode can reduce the data transfer speed.

Migrating file share configurations to Amazon FSx

You can migrate an existing file share configuration to Amazon FSx by using the following procedure. In this procedure, the source file server is the file server whose file share configuration you want to migrate to Amazon FSx.

Note
First migrate your files to Amazon FSx before migrating your file share configuration. For more information, see Migrating existing file storage to FSx for Windows File Server (p. 63).

To migrate existing file shares to FSx for Windows File Server

1. On the source file server, choose **Run as Administrator** from the context menu. Open **Windows PowerShell** as an administrator.
2. Export the source file server's file shares to a file named `SmbShares.xml` by running the following commands in the PowerShell. Replace F: in this example with the drive letter on your file server from which you are exporting file shares.

```
$shareFolder = Get-SmbShare -Special $false | ? { $_.Path -like “F:\*” }
$shareFolder | Export-Clixml -Path F:\SmbShares.xml
```

3. Edit the `SmbShares.xml` file, replacing all references to F: (your drive letter) to D: as Amazon FSx file systems reside on D:.
4. Import the existing file share configuration to FSx for Windows File Server. On a client that has access to your destination Amazon FSx file system and the source file server, copy the saved file share configuration. Then import it into a variable by using the following command.

```
$shares = Import-Clixml -Path F:\SmbShares.xml
```

5. Prepare the credential object required to create the file shares on your FSx for Windows File Server file server using one of the following options.

   To generate the credential object interactively, use the following command.

```
$credential = Get-Credential
```

   To generate the credential object using an AWS Secrets Manager resource, use the following command.

```
$credential = ConvertFrom-Json -InputObject (Get-SECSecretValue -SecretId
 $AdminSecret).SecretString
$FSxAdminUserCredential = (New-Object PSCredential($credential.UserName,(ConvertTo-
SecureString $credential.Password -AsPlainText -Force)))
```

6. Migrate the file share configuration to your Amazon FSx file server using the following script.

```
$FSxAcceptedParameters = ("ContinuouslyAvailable", "Description",
 "ConcurrentUserLimit", "CATimeout", "FolderEnumerationMode", "CachingMode",
 "FullAccess", "ChangeAccess", "ReadAccess", "NoAccess", "SecurityDescriptor", "Path",
 "Name", "EncryptData")
ForEach ($item in $shares) {
    $param = @{};
    Foreach ($property in $item.psObject.properties) {
        if ($property.Name -In $FSxAcceptedParameters) {
            $param[$property.Name] = $property.Value
        }
    }
    Invoke-Command -ConfigurationName FSxRemoteAdmin -ComputerName
 amznfsxxxxxxxxx.corp.com -ErrorVariable errmsg -ScriptBlock { New-FSxSmbShare -
Credential $Using:credential @Using:param }
}
```

Migrating DNS configuration to use Amazon FSx

FSx for Windows File Server provides a default Domain Name System (DNS) name for every file system that you can use to access the data on your file system. You can also access your file systems using any

DNS name of your choosing by configuring the alternate DNS name as a DNS alias for your Amazon FSx file system.

With DNS aliases, you can continue to use your existing DNS names to access data stored on Amazon FSx when migrating file system storage from on-premises to Amazon FSx. This helps eliminate the need to update any tools or applications that use your DNS names when migrating to Amazon FSx. You can associate DNS aliases with existing FSx for Windows File Server file systems, when you create new file systems, and when you create a new file system from a backup. You can associate up to 50 DNS aliases with a file system at any one time. For more information, see Managing DNS aliases (p. 90).

A DNS alias name has to meet the following requirements:

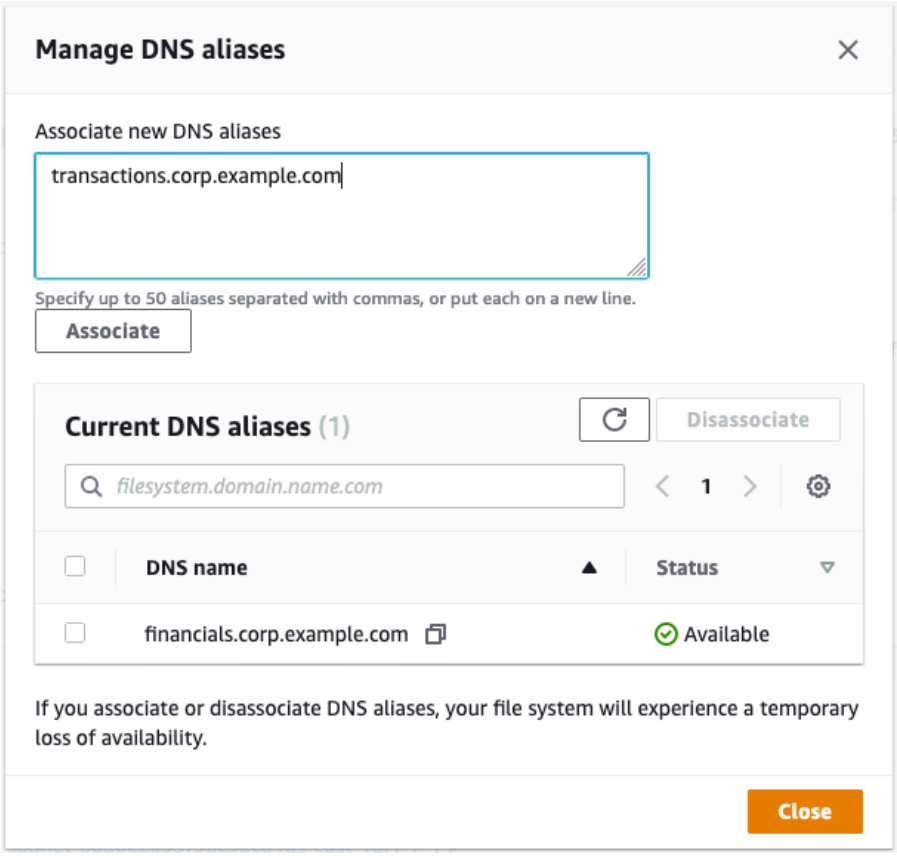
- Must be formatted as a fully qualified domain name (FQDN), for example, `accounting.example.com`.
- Can contain alphanumeric characters and the hyphen (-).
- Cannot start or end with a hyphen.
- Can start with a numeric.

For DNS alias names, Amazon FSx stores alphabetic characters as lowercase letters (a-z), regardless of how you specify them: as uppercase letters, lowercase letters, or the corresponding letters in escape codes.

The following procedures describe how to associate DNS aliases with your existing FSx for Windows File Server file systems using the Amazon FSx console, CLI, and API. For more information about associating DNS aliases when creating new file systems, including new file systems from a backup, see Associating DNS aliases when creating a new file system (p. 92).

To associate DNS aliases with an existing file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to associate your DNS aliases with.
3. On the **Network & security** tab, choose **Manage** for **DNS aliases** to open the **Manage DNS aliases** dialog box.



4. In the **Associate new aliases** box, enter the DNS aliases that you want to associate.
5. Choose **Associate** to add the aliases to the file system.

   You can monitor the status of the aliases that you just associated in the **Current aliases** list. When the status reads **Available**, the alias is associated with the file system (a process that can take up to 2.5 minutes).

To associate DNS aliases with an existing file system (CLI)

- Use the `associate-file-system-aliases` CLI command or the AssociateFileSystemAliases API operation to associate DNS aliases with an existing file system.

  The following CLI request associates two aliases with the specified file system.

```
aws fsx associate-file-system-aliases \
    --file-system-id fs-0123456789abcdef0 \
    --aliases financials.corp.example.com transfers.corp.example.com
```

  The response shows the status of the aliases that Amazon FSx is associating with the file system.

```
{
      "Aliases": [
          {
              "Name": "financials.corp.example.com",
              "Lifecycle": CREATING
          },
```

```
        {
            "Name": "transfers.corp.example.com",
            "Lifecycle": CREATING
        }
    ]
}
```

To monitor the status of the aliases that you are associating, use the `describe-file-system-aliases` CLI command (DescribeFileSystemAliases is the equivalent API operation). When `Lifecycle` for an alias has a value of AVAILABLE, you can use it to access the file system (a process that can take up to 2.5 minutes).

Cutting over to Amazon FSx

To cut over to your FSx for Windows File Server file system, you perform the following steps:

- Prepare for the cut over.
  - Temporarily disconnect SMB clients from the original file system.
  - Perform a final file and file share configuration sync.
- Configure service principal names (SPNs) for your Amazon FSx file system.
- Update DNS CNAME records to point to your Amazon FSx file system.

The procedures to perform each of these steps are provided in the following sections.

Topics

Preparing for the cutover to Amazon FSx

To prepare for the cutover to your Amazon FSx file system, you must do the following:

- Disconnect all clients that write to the original file system.
- Perform a final file sync using AWS DataSync or Robocopy. For more information, see Migrating existing file storage to FSx for Windows File Server (p. 63).
- Perform a final file share configuration sync. For more information, see Migrating file share configurations to Amazon FSx (p. 67).

Configure SPNs for Kerberos authentication

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients that access your file system. To enable Kerberos authentication for clients accessing Amazon FSx using a DNS alias, you must add service principal names (SPNs) that correspond to the DNS alias on your Amazon FSx file system's Active Directory computer object.

There are two required SPNs for Kerberos authentication.

```
HOST/alias
HOST/alias.domain
```

As an example, if the alias is `finance.domain.com`, the two required SPNs are as follows.

```
HOST/finance
HOST/finance.domain.com
```

An SPN can only be associated with a single Active Directory computer object at a time. If there are existing SPNs for the DNS name configured for your original file system's Active Directory computer object, you must delete them before creating SPNs for your Amazon FSx file system.

The following procedures describe how to find any existing SPNs, delete them, and create new SPNs for your Amazon FSx file system's Active Directory computer object.

To install the required PowerShell Active Directory module

1. Log on to a Windows instance joined to the Active Directory that your Amazon FSx file system is joined to.
2. Open PowerShell as administrator.
3. Install the PowerShell Active Directory module using the following command.

```
Install-WindowsFeature RSAT-AD-PowerShell
```

To find and delete existing DNS alias SPNs on the original file system's Active Directory computer object

1. Find any existing SPNs by using the following commands. Replace *alias_fqdn* with the DNS alias that you associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).

```
## Find SPNs for original file system's AD computer object
$ALIAS = "alias_fqdn"
SetSPN /Q ("HOST/" + $ALIAS)
SetSPN /Q ("HOST/" + $ALIAS.Split(".")[0])
```

2. Delete the existing HOST SPNs returned in the previous step by using the following example script.

   - Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).
   - Replace *file_system_DNS_name* with the original file system's DNS name .

```
## Delete SPNs for original file system's AD computer object
$Alias = "alias_fqdn"
$FileSystemDnsName = "file_system_dns_name"
$FileSystemHost = (Resolve-DnsName ${FileSystemDnsName} | Where Type -eq 'A')
[0].Name.Split(".")[0]
$FSxAdComputer = (Get-AdComputer -Identity ${FileSystemHost})

SetSPN /D ("HOST/" + ${Alias}) ${FSxAdComputer}.Name
SetSPN /D ("HOST/" + ${Alias}.Split(".")[0]) ${FSxAdComputer}.Name
```

3. Repeat these steps for each DNS alias that you associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).

To set SPNs on your Amazon FSx file system's Active Directory computer object

1. Set new SPNs for your Amazon FSx file system by running the following commands.

- Replace *file_system_DNS_name* with the DNS name that Amazon FSx assigned to the file system.

  To find your file system's DNS name on the Amazon FSx console, choose **File systems**, and choose your file system. Choose the **Network & security** pane of the file system details page. You can also get the DNS name in the response of the DescribeFileSystems API operation.
- Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).

```
## Set SPNs for FSx file system AD computer object
$FSxDnsName = "file_system_DNS_name"
$Alias = "alias_fqdn"
$FileSystemHost = (Resolve-DnsName $FSxDnsName | Where Type -eq 'A')[0].Name.Split(".")
[0]
$FSxAdComputer = (Get-AdComputer -Identity $FileSystemHost)

Set-AdComputer -Identity $FSxAdComputer -Add @{"msDS-AdditionalDnsHostname"="$Alias"}
SetSpn /S ("HOST/" + $Alias.Split('.')[0]) $FSxAdComputer.Name
SetSpn /S ("HOST/" + $Alias) $FSxAdComputer.Name
```

Note
Setting an SPN for your Amazon FSx file system will fail if an SPN for the DNS alias exists in the AD for the original file system's computer object. For information about finding and deleting existing SPNs, see To find and delete existing DNS alias SPNs on the original file system's Active Directory computer object (p. 72).

2. Verify that the new SPNs are configured for the DNS alias using the following example script. Ensure that the response includes two HOST SPNs, `HOST/`*alias* and `HOST/`*alias_fqdn*.

   Replace *file_system_DNS_name* with the DNS name that Amazon FSx assigned to your file system. To find your file system's DNS name on the Amazon FSx console, choose **File systems**, choose your file system, and then choose the **Network & security** pane on the file system details page.

   You can also get the DNS name in the response of the DescribeFileSystems API operation.

```
## Verify SPNs on FSx file system AD computer object
$FileSystemDnsName = "file_system_dns_name"
$FileSystemHost = (Resolve-DnsName ${FileSystemDnsName} | Where Type -eq 'A')
[0].Name.Split(".")[0]
$FSxAdComputer = (Get-AdComputer -Identity ${FileSystemHost})
SetSpn /L ${FSxAdComputer}.Name
```

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).

Note
You can enforce Kerberos authentication and encryption in transit with clients connecting to your file system using DNS aliases by setting the following Group Policy Objects (GPOs) in your Active Directory:

- Restrict NTLM: Outgoing NTLM traffic to remote servers
- Restrict NTLM: Add remote server exceptions for NTLM authentication

For more information, see Enforcing Kerberos authentication using GPOs (p. 171) in *Walkthrough 5: Using DNS aliases to access your file system*.

Update the DNS CNAME records for the Amazon FSx file system

After you properly configure SPNs for your file system, you can cut over to Amazon FSx by replacing each DNS record that resolved to the original file system with a DNS record that resolves to the default DNS name of the Amazon FSx file system.

To install the required PowerShell cmdlets

1. Log on to a Windows instance joined to the Active Directory that your Amazon FSx file system is joined to as a user that is a member of a group that has DNS administration permissions (**AWS Delegated Domain Name System Administrators** inAWS Managed Microsoft Active Directory, and **Domain Admins** or another group to which you've delegated DNS administration permissions in your self-managed Active Directory)

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
2. Open PowerShell as administrator.
3. The PowerShell DNS server module is required to perform the instructions in this procedure. Install it using the following command.

```
Install-WindowsFeature RSAT-DNS-Server
```

To update an existing a DNS CNAME record

1. The following script updates any existing DNS CNAME records for the *alias_fqdn* to your Amazon FSx file system's computer object. If none is found, it creates a new DNS CNAME record for the DNS alias *alias_fqdn* that resolves to the default DNS name for your Amazon FSx file system.

   To run the script:

   - Replace *alias_fqdn* with the DNS alias that you associated with the file system.
   - Replace *file_system_DNS_name* with the default DNS name Amazon FSx has assigned to the file system.

```
$Alias="alias_fqdn"
$FSxDnsName="file_system_dns_name"
$AliasHost=$Alias.Split('.')[0]
$ZoneName=((Get-WmiObject Win32_ComputerSystem).Domain)
$DnsServerComputerName = (Resolve-DnsName $ZoneName -Type NS | Where Type -eq 'A' |
 Select -ExpandProperty Name)[0]

Add-DnsServerResourceRecordCName -Name $AliasHost -ComputerName $DnsServerComputerName
 -HostNameAlias $FSxDnsName -ZoneName $ZoneName
```

2. Repeat the previous step for each DNS alias that you associated with the file system in Migrating DNS configuration to use Amazon FSx (p. 68).

Using FSx for Windows File Server with Microsoft SQL Server

High availability (HA) Microsoft SQL Server is typically deployed across multiple database nodes in a Windows Server Failover Cluster (WSFC), with each node having access to shared file storage. You can use FSx for Windows File Server as shared storage for High Availability (HA) Microsoft SQL Server deployments in two ways: as storage for active data files and as an SMB file share witness.

SSD storage is recommended for SQL Server. SSD storage is designed for the highest-performance and most latency-sensitive workloads, including databases.

For information about using Amazon FSx to reduce complexity and costs for your SQL Server high availability deployments, see the following posts on the *AWS Storage Blog*:

- Simplify your Microsoft SQL Server high availability deployments using Amazon FSx for Windows File Server
- Optimizing cost for your high availability SQL Server deployments on AWS
- Simplify SQL Server Always On deployments with AWS Launch Wizard and Amazon FSx

Using Amazon FSx for Active SQL Server Data Files

Microsoft SQL Server can be deployed with an SMB file share as the storage option for active data files. Amazon FSx is optimized to provide shared storage for SQL Server databases by supporting continuously available (CA) file shares. These file shares are designed for applications like SQL Server that require uninterrupted access to shared file data. While you can create CA shares on Single-AZ 2 file systems, it is required that you use CA shares on Multi-AZ file systems for all SQL Server deployments, whether HA or not.

Create a Continuously Available Share

You can create CA shares using the Amazon FSx CLI for Remote Management on PowerShell. To specify that the share is a continuously available share, use the `New-FSxSmbShare` with the `-ContinuouslyAvailable` option set to `$True`. To learn more about creating a new CA share, see Creating a continuously available share (p. 97).

Configure SMB timeout settings

As described in Failover process for FSx for Windows File Server (p. 21), failover and failback for Multi-AZ can result in I/O pauses that typically complete in less than 30 seconds. Your SQL Server application may have different sensitivity to timeout settings depending on how it is configured.

You can tune the SMB client configuration session timeout to make sure your application is resilient to Multi-AZ file system failovers. You can test the behavior of your application during failovers by updating your file system's throughput capacity, which initiates an automatic failover and failback.

Using Amazon FSx as an SMB File Share Witness

Windows Server Failover cluster deployments commonly deploy an SMB file share witness to maintain quorum of the cluster's resources. Witness file shares require only a small amount of storage for quorum

information. Amazon FSx file systems can be used as an SMB file share witness for Windows Server Failover Cluster deployments.

Using FSx for Windows File Server with Amazon Kendra

Amazon Kendra is a highly accurate and intelligent search service. FSx for Windows File Server file systems can be used as data sources for Amazon Kendra, allowing you to index and intelligently search for information contained in documents stored on your file system.

- For more information about Amazon Kendra, see What is Amazon Kendra in the *Amazon Kendra Developer's Guide*.
- For more information about how to add your file system as an Amazon Kendra data source, see Getting started with an Amazon FSx data source (console) in the *Amazon Kendra Developer's Guide*.
- For overview information about Amazon Kendra, see the Amazon Kendra website.
- For a walkthrough of how to search your file system using Amazon Kendra, see Securely search unstructured data on Windows file systems with the Amazon Kendra connector for Amazon FSx for Windows File Server on the *AWS Machine Learning Blog*.

File system performance

When you add an FSx for Windows File Server file system as a data source, Amazon Kendra crawls the files and folders on the file system on a regular sync frequency to create and maintain its search index. (You can select the sync frequency when you establish the integration.) This file access activity from Amazon Kendra will consume file system resources, similar to activity from your own workloads accessing the file system.

Ensure your file system is configured with sufficient resources such that your workload performance is not impacted. Specifically, if you are planning to index a large number of files, we recommend using a file system with SSD storage type, which provides higher maximum throughput and IOPS levels for requests that need to access the storage volumes.

For more information about the Amazon FSx performance model, see FSx for Windows File Server performance (p. 153).

Protecting your data with backups, shadow copies, and scheduled replication

Beyond automatically replicating your file system's data to ensure high durability, Amazon FSx provides you with the following options to further protect the data stored on your file systems:

- Native Amazon FSx backups support your backup retention and compliance needs within Amazon FSx.
- AWS Backup backups of your Amazon FSx file systems are part of a centralized and automated backup solution across AWS services in the cloud and on premises.
- Windows shadow copies enable your users to easily undo file changes and compare file versions by restoring files to previous versions.
- AWS DataSync scheduled replication of your Amazon FSx file system to a second file system provides data protection and recovery.

Topics

Working with backups

With Amazon FSx, backups are file-system-consistent, highly durable, and incremental. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in Microsoft Windows. To ensure high durability, Amazon FSx stores backups in Amazon Simple Storage Service (Amazon S3).

Amazon FSx backups are incremental, whether they are generated using the automatic daily backup or the user-initiated backup feature. This means that only the data on the file system that has changed after your most recent backup is saved. This minimizes the time required to create the backup and saves on storage costs by not duplicating data. When you delete a backup, only the data unique to that backup is removed. Each FSx for Windows File Server backup contains all of the information that is needed to create a new file system from the backup, effectively restoring a point-in-time snapshot of the file system.

Creating regular backups for your file system is a best practice that complements the replication that Amazon FSx for Windows File Server performs for your file system. Amazon FSx backups help support your backup retention and compliance needs. Working with Amazon FSx backups is easy, whether it's creating backups, copying a backup, restoring a file system from a backup, or deleting a backup.

Topics

Working with automatic daily backups

By default, Amazon FSx takes an automatic daily backup of your file system. These automatic daily backups occur during the daily backup window that was established when you created the file system. At some point during the daily backup window, storage I/O might be suspended briefly while the backup process initializes (typically for less than a few seconds). When you choose your daily backup window, we recommend that you choose a convenient time of the day. This time ideally is outside of the normal operating hours for the applications that use the file system.

Automatic daily backups are kept for a certain period of time, known as a retention period. The default retention period for automatic daily backups is 7 days. You can set the retention period to be between 0–90 days. Setting the retention period to 0 (zero) days turns off automatic daily backups. Automatic daily backups are deleted when the file system is deleted.

> **Note**
> Setting the retention period to 0 days means that your file system is never automatically backed up. We highly recommend that you use automatic daily backups for file systems that have any level of critical functionality associated with them.

You can use the AWS CLI or one of the AWS SDKs to change the backup window and backup retention period for your file systems. Use the `UpdateFileSystem` API operation or the `update-file-system` CLI command. For more information, see Walkthrough 3: Update an existing file system (p. 162).

Working with user-initiated backups

With Amazon FSx, you can manually take backups of your file systems at any time. You can do so using the Amazon FSx console, API, or the AWS Command Line Interface (AWS CLI). Your user-initiated backups of Amazon FSx file systems never expire, and they are available for as long as you want to keep them. User-initiated backups are retained even after you delete the file system that was backed up. You can delete user-initiated backups only by using the Amazon FSx console, API, or CLI. They are never automatically deleted by Amazon FSx. For more information, see Deleting backups (p. 83).

If a backup is initiated while the file system is being modified (such as during an update to throughput capacity, or during file system maintenance), the backup request is queued and will resume when the activity is complete.

Creating user-initiated backups

The following procedure guides you through how to create a user-initiated backup in the Amazon FSx console for an existing file system.

To create a user-initiated file system backup

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose the name of the file system that you want to back up.
3. From **Actions**, choose **Create backup**.
4. In the **Create backup** dialog box that opens, provide a name for your backup. Backup names can be a maximum of 256 Unicode characters, including letters, white space, numbers, and the special characters . + - = _ : /
5. Choose **Create backup**.

You have now created your file system backup. You can find a table of all your backups in the Amazon FSx console by choosing **Backups** in the right side navigation. You can search for the name you gave your backup, and the table filters to only show matching results.

When you create a user-initiated backup as this procedure described, it has the type `USER_INITIATED`, and it has the `CREATING` status until it is fully available.

Using AWS Backup with Amazon FSx

AWS Backup is a simple and cost-effective way to protect your data by backing up your Amazon FSx file systems. AWS Backup is a unified backup service designed to simplify the creation, copying, restoration, and deletion of backups, while providing improved reporting and auditing. AWS Backup makes it easier to develop a centralized backup strategy for legal, regulatory, and professional compliance. AWS Backup also makes protecting your AWS storage volumes, databases, and file systems simpler by providing a central place where you can do the following:

- Configure and audit the AWS resources that you want to back up.
- Automate backup scheduling.
- Set retention policies.
- Copy backups across AWS Regions and across AWS accounts.
- Monitor all recent backup, copy, and restore activity.

AWS Backup uses the built-in backup functionality of Amazon FSx. Backups taken from the AWS Backup console have the same level of file system consistency and performance, and the same restore options as backups taken through the Amazon FSx console. If you use AWS Backup to manage these backups, you gain additional functionality, such as unlimited retention options and the ability to create scheduled backups as frequently as every hour. In addition, AWS Backup retains your immutable backups even after the source file system is deleted. This protects against accidental or malicious deletion.

Backups taken by AWS Backup are considered user-initiated backups, and they count toward the user-initiated backup quota for Amazon FSx. You can see and restore backups taken by AWS Backup in the Amazon FSx console, CLI, and API. However, you can't delete backups taken by AWS Backup in the Amazon FSx console, CLI, or API. For more information about how to use AWS Backup to back up your Amazon FSx file systems, see Working with Amazon FSx File Systems in the *AWS Backup Developer Guide*.

Copying backups

You can use Amazon FSx to manually copy backups within the same AWS account to another AWS Region (cross-Region copies) or within the same AWS Region (in-Region copies). You can make cross-Region copies only within the same AWS partition. You can create user-initiated backup copies using the Amazon FSx console, AWS CLI, or API. When you create a user-initiated backup copy, it has the type `USER_INITIATED`.

You can also use AWS Backup to copy backups across AWS Regions and across AWS accounts. AWS Backup is a fully managed backup management service that provides a central interface for policy-based backup plans. With its cross-account management, you can automatically use backup policies to apply backup plans across the accounts within your organization.

Cross-Region backup copies are particularly valuable for cross-Region disaster recovery. You take backups and copy them to another AWS Region so that in the event of a disaster in the primary AWS Region, you can restore from backup and recover availability quickly in the other AWS Region. You can also use backup copies to clone your file dataset to another AWS Region or within the same AWS Region. You make backup copies within the same AWS account (cross-Region or in-Region) by using the Amazon FSx console, AWS CLI, or Amazon FSx API. You can also use AWS Backup to perform backup copies, either on-demand or policy-based.

Cross-account backup copies are valuable for meeting your regulatory compliance requirements to copy backups to an isolated account. They also provide an additional layer of data protection to help prevent accidental or malicious deletion of backups, loss of credentials, or compromise of AWS KMS keys. Cross-

account backups support *fan-in* (copy backups from multiple primary accounts to one isolated backup copy account) and *fan-out* (copy backups from one primary account to multiple isolated backup copy accounts).

You can make cross-account backup copies by using AWS Backup with AWS Organizations support. Account boundaries for cross-account copies are defined by AWS Organizations policies. For more information about using AWS Backup to make cross-account backup copies, see Creating backup copies across AWS accounts in the *AWS Backup Developer Guide*.

Backup copy limitations

The following are some limitations when you copy backups:

- Cross-Region backup copies are supported only between any two commercial AWS Regions, between the China (Beijing) and China (Ningxia) Regions, and between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions, but not across those sets of Regions.
- Cross-Region backup copies are not supported in opt-in Regions.
- You can make in-Region backup copies within any AWS Region.
- The source backup must have a status of `AVAILABLE` before you can copy it.
- You cannot delete a source backup if it is being copied. There might be a short delay between when the destination backup becomes available and when you are allowed to delete the source backup. You should keep this delay in mind if you retry deleting a source backup.
- You can have up to five backup copy requests in progress to a single destination AWS Region per account.

Permissions for cross-Region backup copies

You use an IAM policy statement to grant permissions to perform a backup copy operation. To communicate with the source AWS Region to request a cross-Region backup copy, the requester (IAM role or IAM user) must have access to the source backup and the source AWS Region.

You use the policy to grant permissions to the `CopyBackup` action for the backup copy operation. You specify the action in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field, as in the following example.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fsx:CopyBackup",
            "Resource": "arn:aws:fsx:*:111111111111:backup/*"
        }
    ]
}
```

For more information on IAM policies, see Policies and permissions in IAM in the *IAM User Guide*.

Full and incremental copies

When you copy a backup to a different AWS Region from the source backup, the first copy is a full backup copy. After the first backup copy, all subsequent backup copies to the same destination Region within the same AWS account are incremental, provided that you haven't deleted all previously-copied backups in that Region and have been using the same AWS KMS key. If both conditions aren't met, the copy operation results in a full (not incremental) backup copy.

To copy a backup within the same account (cross-Region or in-Region) using the console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the navigation pane, choose **Backups**.
3. In the **Backups** table, choose the backup that you want to copy, and then choose **Copy backup**.
4. In the **Settings** section, do the following:
   - In the **Destination Region** list, choose a destination AWS Region to copy the backup to. The destination can be in another AWS Region (cross-Region copy) or within the same AWS Region (in-Region copy).
   - (Optional) Select **Copy Tags** to copy tags from the source backup to the destination backup. If you select **Copy Tags** and also add tags at step 6, all the tags are merged.
5. For **Encryption**, choose the AWS KMS encryption key to encrypt the copied backup.
6. For **Tags - optional**, enter a key and value to add tags for your copied backup. If you add tags here and also selected **Copy Tags** at step 4, all the tags are merged.
7. Choose **Copy backup**.

Your backup is copied within the same AWS account to the selected AWS Region.

To copy a backup within the same account (cross-Region or in-Region) using the CLI

- Use the `copy-backup` CLI command or the CopyBackup API operation to copy a backup within the same AWS account, either across an AWS Region or within an AWS Region.

  The following command copies a backup with an ID of `backup-0abc123456789cba7` from the `us-east-1` Region.

  ```
  aws fsx copy-backup \
    --source-backup-id backup-0abc123456789cba7 \
    --source-region us-east-1
  ```

  The response shows the description of the copied backup.

  You can view your backups on the Amazon FSx console or programmatically using the `describe-backups` CLI command or the DescribeBackups API operation.

Restoring backups

You can use an available backup to create a new file system, effectively restoring a point-in-time snapshot of another file system. You can restore a backup using the console, AWS CLI, or one of the AWS SDKs. Restoring a backup to a new file system takes the same amount of time as creating a new file system. The data restored from the backup is lazy-loaded onto the file system, during which time you will experience slightly higher latency.

The following procedure guides you through how to restore a backup using the console to create a new file system.

Note
You can only restore your backup to a file system of the same deployment type and storage capacity as the original. You can increase your restored file system's storage capacity after it becomes available. For more information, see Managing storage capacity (p. 123).

To restore a file system from a backup

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose **Backups** from the left side navigation.
3. Choose the backup that you want to restore from the **Backups** table, and then choose **Restore backup**.

   Doing so opens the file system creation wizard. This wizard is identical to the standard file system creation wizard, except the **Deployment type** and **Storage capacity** are already set and can't be changed. However, you can change the throughput capacity, associated VPC, and other settings, and storage type. The storage type is set to **SSD** by default, but you can change it to **HDD** under the following conditions:

   - The file system deployment type is **Multi-AZ** or **Single-AZ 2**.
   - The storage capacity is at least 2,000 GiB.
4. Complete the wizard as you do when you create a new file system.
5. Choose **Review and create**.
6. Review the settings you chose for your Amazon FSx file system, and then choose **Create file system**.

You have restored from a backup, and a new file system is now being created. When its status changes to `AVAILABLE`, you can use the file system as normal.

Deleting backups

Deleting a backup is a permanent, unrecoverable action. Any data in a deleted backup is also deleted. Do not delete a backup unless you're sure you won't need that backup again in the future. You can't delete backups taken by AWS Backup, which have type **AWS Backup**, in the Amazon FSx console, CLI, or API.

To delete a backup

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose **Backups** from the right side navigation.
3. Choose the backup that you want to delete from the **Backups** table, and then choose **Delete backup**.
4. In the **Delete backups** dialog box that opens, confirm that the ID of the backup identifies the backup that you want to delete.
5. Confirm that the check box is checked for the backup that you want to delete.
6. Choose **Delete backups**.

Your backup and all included data are now permanently and unrecoverably deleted.

Working with shadow copies

A Microsoft Windows *shadow copy* is a snapshot of a Windows file system at a point in time. With shadow copies enabled, your users can easily view and restore individual files or folders from an earlier snapshot in Windows File Explorer. Doing this enables users to easily undo changes and compare file versions. Storage administrators using Amazon FSx can easily schedule shadow copies to be taken periodically using Windows PowerShell commands.

Shadow copies are stored alongside your file system's data, and therefore consume the file system's storage capacity. However, shadow copies consume storage capacity only for the changed portions of files. All shadow copies stored in your file system are included in backups of your file system.

Note
Shadow copies are *not* enabled on FSx for Windows File Server by default. To have shadow copies running on your file system, you must enable shadow copies and set up a shadow copy schedule on your file system. For more information, see Setting up shadow copies using default settings (p. 85).

Note
Shadow copies are not a substitute for backups. If you enable shadow copies, make sure that you continue performing regular backups.

Topics

Shadow copies configuration overview

You enable and schedule periodic shadow copies on your file system using Windows PowerShell commands defined by Amazon FSx. Shadow copy configuration contains two settings:

- The maximum amount of storage that shadow copies can consume on your file system
- (Optional) A schedule to take shadow copies at defined times and intervals, such as daily, weekly, and monthly

You can store up to 500 shadow copies per file system at any point in time. When you reach this limit, the next shadow copy that you take replaces the oldest shadow copy. Similarly, when the maximum shadow copy storage amount is reached, one or more of the oldest shadow copies are deleted to make sufficient storage space for the next shadow copy.

For information about how to quickly enable and schedule periodic shadow copies by using default Amazon FSx settings, see Setting up shadow copies using default settings (p. 85). For information about how to customize your shadow copy configuration, see Shadow copies (p. 117).

Considerations for allocating shadow copy storage

A shadow copy is a block-level copy of file changes that were made since the last shadow copy. The entire file is not copied, only the changes. Therefore, previous versions of files typically don't take up as much storage space as the current file. The amount of volume space used for changes can vary according to your workload. When a file is modified, the storage space used by shadow copies depends on your workload. When you determine how much storage space to allocate for shadow copies, you should account for your workload's file system usage patterns.

When you enable shadow copies, you can specify the maximum amount of storage that shadow copies can consume on the file system. The default limit is 10 percent of your file system. We recommend that you increase the limit if your users frequently add or modify files. Setting the limit too small can result in the oldest shadow copies being deleted more often than users might expect.

You can set the shadow copy storage as unbounded (`Set-FsxShadowStorage -Maxsize "UNBOUNDED"`). However, an unbounded configuration can result in a large number of shadow copies consuming your file system storage. This could result in not having enough storage capacity for your workloads. If you set an unbounded storage, be sure to scale your storage capacity as the shadow copy limits are reached. For information about configuring your shadow copy storage to a specific size or as unbounded, see Setting shadow copy storage (p. 118).

After you enable shadow copies, you can monitor the amount of storage space consumed by the shadow copies. For more information, see Viewing your shadow copy storage (p. 119).

File system recommendations for shadow copies

Following are file system recommendations for using shadow copies.

- Make sure you provision sufficient performance capacity for your workload needs on your file system. Amazon FSx delivers the Shadow Copies feature as provided by Microsoft Windows Server. By design, Microsoft Windows uses a copy-on-write method for recording the changes since the most recent shadow copy point, and this copy-on-write activity can result in up to three I/O operations for every file write operation. If Windows is unable to keep up with the incoming rate of I/O operations per second, it can cause all shadow copies to be deleted because it can no longer maintain the shadow copies via copy-on-write. Therefore, it is important that you provision sufficient I/O performance capacity for your workload needs on your file system (both the throughput capacity dimension that determines the file server I/O performance, and the storage type and capacity that determine the storage I/O performance).
- We generally recommend that you use file systems configured with SSD storage rather than HDD storage when you enable shadow copies, given that Windows consumes a higher I/O performance to maintain shadow copies, and given that HDD storage provides lower performance capacity for I/O operations.
- Your file system should have at least 320 MB of free space, in addition to the maximum shadow copy storage amount configured (`MaxSpace`). For example, if you allocated 5 GB `MaxSpace` to shadow copies, your file system should always have at least 320 MB free space in addition to the 5 GB `MaxSpace`.
- When configuring your shadow copy schedule, make sure that you don't schedule shadow copies when migrating data or when data deduplication jobs are scheduled to run. You should schedule shadow copies when you expect your file system to be idle. For information about configuring a custom shadow copy schedule, see Creating a custom shadow copy schedule (p. 120).

Setting up shadow copies using default settings

You can quickly set up shadow copies on your file system by using the default settings available for shadow copy storage and schedule. The default shadow copy storage setting lets shadow copies consume a maximum of 10 percent of your file system. If you increase your file system's storage capacity (either as a percentage or an absolute value), the amount of the currently allocated shadow copy storage is not similarly increased.

The default schedule automatically takes shadow copies every Monday, Tuesday, Wednesday, Thursday, and Friday, at 7:00 AM and 12:00 PM UTC.

To set up the default level of shadow copy storage

1. Connect to a Windows compute instance that has network connectivity with your file system.
2. Log in to the Windows compute instance as a member of the file system administrators group. In AWS Managed Microsoft AD, that group is **AWS Delegated FSx Administrators**. In your self-managed Microsoft AD, that group is **Domain Admins** or the custom group that you specified for administration when you created your file system. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Set the default amount of shadow storage using the following command. Replace *`FSxFileSystem-Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer. You can find the Windows Remote PowerShell endpoint in the Amazon FSx console, in the **Network & Security** section of the file system details screen, or in the response of the `DescribeFileSystem` API operation.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName FSxFileSystem-Remote-
PowerShell-Endpoint -ConfigurationName FSxRemoteAdmin -scriptblock {Set-
FsxShadowStorage -Default}
```

The response looks like the following.

```
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace     MaxSpace
-------------- ---------     --------
             0         0  32530536858
```

To create the default shadow copy schedule

- Set the default shadow copy schedule by entering the following command.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName FSxFileSystem-Remote-
PowerShell-Endpoint -ConfigurationName FSxRemoteAdmin -scriptblock {Set-
FsxShadowCopySchedule -Default}
```

The response displays the default schedule that is now set.

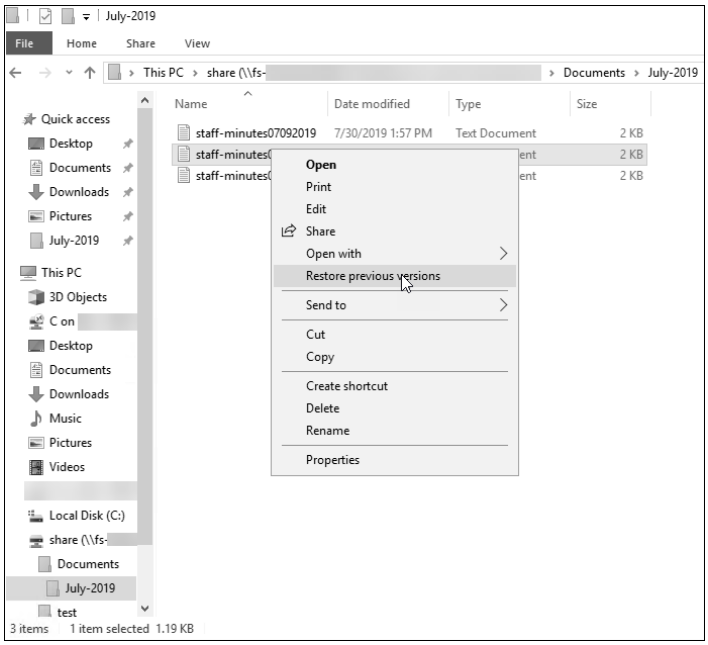
```
FSx Shadow Copy Schedule

Start Time                Days of week                            WeeksInterval
----------                ------------                            -------------
2019-07-16T07:00:00+00:00 Monday,Tuesday,Wednesday,Thursday,Friday              1
2019-07-16T12:00:00+00:00 Monday,Tuesday,Wednesday,Thursday,Friday              1
```

To learn about additional options and creating a custom shadow copy schedule, see Creating a custom shadow copy schedule (p. 120).

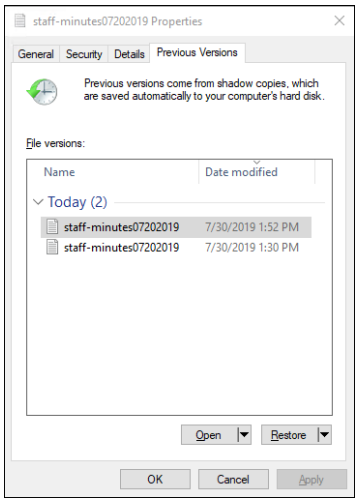
Restoring individual files and folders

After you configure shadow copies on your Amazon FSx file system, your users can quickly restore previous versions of individual files or folders. Doing this enables them to recover deleted or changed files stored on the shared file system. They do this in a self-service manner directly on their desktop without administrator assistance. This self-service approach increases productivity and reduces administrative workload.

Users restore files to previous versions using the familiar Windows File Explorer interface. To restore a file, you choose the file to restore, then choose **Restore previous versions** from the context (right-click) menu.



Users can then view and restore a previous version from the **Previous Versions** list.



To learn about the complete set of custom PowerShell commands available for managing shadow copies on your FSx for Windows File Server shares, see Shadow copies (p. 117).

Scheduled replication using AWS DataSync

You can use AWS DataSync to schedule periodic replication of your FSx for Windows File Server file system to a second file system. This capability is available for both in-Region and cross-Region deployments. To learn more, see Migrating existing files to FSx for Windows File Server using AWS DataSync (p. 64) in this guide and Data transfer between AWS storage services in the *AWS DataSync User Guide*.

Administering file systems

You can administer your FSx for Windows File Server file systems using custom remote-management PowerShell commands, or the Microsoft Windows–native graphical user interface (GUI) in some cases. Following, you can find a description of all custom PowerShell commands in each of the file system management categories available.

Topics

Getting started with the Amazon FSx CLI for remote management on PowerShell

The Amazon FSx CLI for remote management on PowerShell enables file system administration for users in the file system administrators group. To start a remote PowerShell session on your FSx for Windows File Server file system, first meet the following prerequisites:

- Be able to connect to a Windows compute instance that has network connectivity with your file system.
- Be logged into the Windows compute instance as a member of the file system administrators group. In AWS Managed Microsoft AD, that group is AWS Delegated FSx Administrators. In your self-managed Microsoft AD, that group is Domain Admins or the custom group that you specified for administration when you created your file system. For more information, see Self-managed AD best practices (p. 38).
- Make sure that your file system's security group inbound rules allows traffic on port 5985.

Security and the CLI for remote management on PowerShell

The Amazon FSx CLI for remote management on PowerShell uses the following security features:

- User logins are authenticated using Kerberos authentication.
- Management session communications are encrypted using Kerberos.

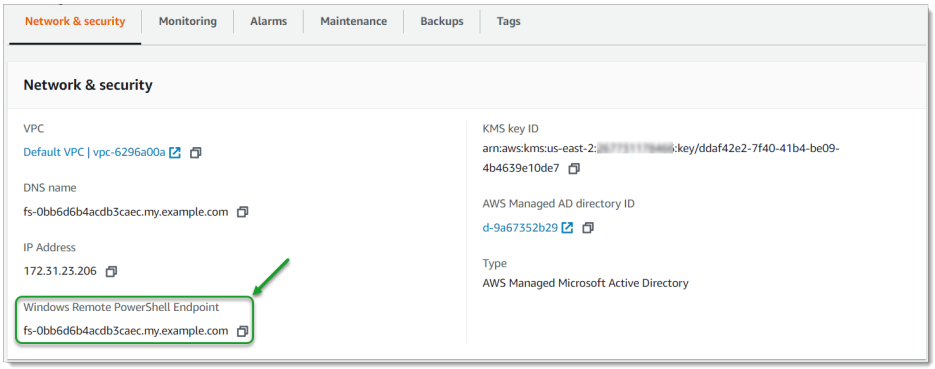
Using the CLI for remote management on PowerShell

You have two options to run remote management commands on your Amazon FSx file system. You can establish a long-running Remote PowerShell session and run the commands inside the session. Or, you can use the `Invoke-Command` to run a single command or a single block of commands without establishing a long-running Remote PowerShell session. If you want to set and pass variables as parameters to the remote management command, you need to use `Invoke-Command`.

> **Note**
> For Multi-AZ file systems, you can only use the Amazon FSx CLI for Remote Management while the file system is on its preferred file server. For more information, see Availability and durability: Single-AZ and Multi-AZ file systems (p. 20).

To run these commands, you must know the *Windows Remote PowerShell Endpoint* for your file system. To find this endpoint, follow these steps:

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose your file system. On the **Network & security** tab, locate the **Windows Remote PowerShell endpoint**, as shown following.



To start a remote PowerShell session on your file system

1. Connect to a compute instance that has network connectivity with your file system as a user that is a member of the file system administrators group.
2. Open a Windows PowerShell window on the compute instance.
3. Use the following command to open the remote session on your Amazon FSx file system. Replace *`FSxFileSystem-Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer.

```
PS C:\Users\delegateadmin> enter-pssession -ComputerName FSxFileSystem-Remote-
PowerShell-Endpoint -ConfigurationName FsxRemoteAdmin
[fs-0123456789abcdef0]: PS>
```

You are prompted to enter user credentials in a pop-up.

You can also run Amazon FSx CLI for remote management CLI on PowerShell commands on your file system using the `Invoke-Command` cmdlet, described following.

The following example illustrates the syntax required when using the `Invoke-Command` cmdlet to run PowerShell commands on an FSx for Windows File Server file system.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -scriptblock { fsx-command}
```

Managing DNS aliases

FSx for Windows File Server provides a default Domain Name System (DNS) name for every file system that you can use to access the data on your file system. You can also access your file systems using a DNS alias of your choosing. With DNS aliases, you can continue using existing DNS names to access data stored on Amazon FSx when migrating file system storage from on-premises to Amazon FSx, without needing to update any tools or applications. For more information, see Migrating existing file storage to Amazon FSx (p. 63).

Note
Support for DNS aliases is available on FSx for Windows File Server file systems created after 12:00 pm ET on November 9, 2020. To use DNS aliases on a file system created before 12:00 pm ET on November 9, 2020, do the following:

1. Take a backup of the existing file system. For more information, see Working with user-initiated backups (p. 79).
2. Restore the backup to a new file system. For more information, see Restoring backups (p. 82).

Once the new file system is available, you will be able to use DNS aliases to access it, using the information provided in this section.

Note
The information presented here assumes that you're working entirely within Active Directory and that you're not using external DNS providers.

You can associate DNS aliases with existing FSx for Windows File Server file systems, when you create new file systems, and when you create a new file system from a backup. You can associate up to 50 DNS aliases with a file system at any one time.

In addition to associating DNS aliases with your file system, for clients to connect to the file system using the DNS aliases, you also must do the following:

- Configure service principal names (SPNs) for Kerberos authentication and encryption.
- Configure a DNS CNAME record for the DNS alias that resolves to the default DNS name for your Amazon FSx file system.

For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

A DNS alias name must meet the following requirements:

- Must be formatted as a fully qualified domain name (FQDN).
- Can contain alphanumeric characters and hyphens (-).
- Cannot start or end with a hyphen.
- Can start with a numeric.

For DNS alias names, Amazon FSx stores alphabetic characters as lowercase letters (a-z), regardless of how you specify them: as uppercase letters, lowercase letters, or the corresponding letters in escape codes.

If you try to associate an alias that is already associated with the file system, it has no effect. If you try to disassociate an alias from a file system that is not associated with the file system, Amazon FSx responds with a bad request error.

> **Note**
> When Amazon FSx adds or removes aliases on a file system, connected clients are temporarily disconnected and will automatically reconnect to the file system. Any files that were open by clients mapping a non-Continuously-Available (non-CA) share at the time of disconnection must be reopened by the client.

Topics

Using DNS aliases with Kerberos authentication

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients that access your file system. To enable Kerberos authentication for clients that access your Amazon FSx file system using a DNS alias, you must configure service principal names (SPNs) that correspond to the DNS alias on your Amazon FSx file system's Active Directory computer object.

If you have SPNs configured for the DNS alias that you've assigned to another file system on a computer object in your Active Directory, you must first remove those SPNs before adding SPNs to your file system's computer object. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

Viewing DNS aliases associated with file systems and backups

You can see the DNS aliases that are currently associated with file systems and backups using the Amazon FSx console, the AWS CLI, and the Amazon FSx API and SDKs.

To view DNS aliases associated with file systems:

- Using the console — Choose a file system to view the **File systems** detail page. Choose the **Network & security** tab to view the **DNS aliases**.
- Using the CLI or API — Use the `describe-file-system-aliases` CLI command or the DescribeFileSystemAliases API operation.

To view DNS aliases associated with backups:

- Using the console — In the navigation pane, choose **Backups**, and then choose the backup that you want to view. In the **Summary** pane, view the **DNS aliases** field.
- Using the CLI or API — Use the `describe-backups` CLI command or the DescribeBackups API operation.

DNS alias status

DNS aliases can have one of the following values:

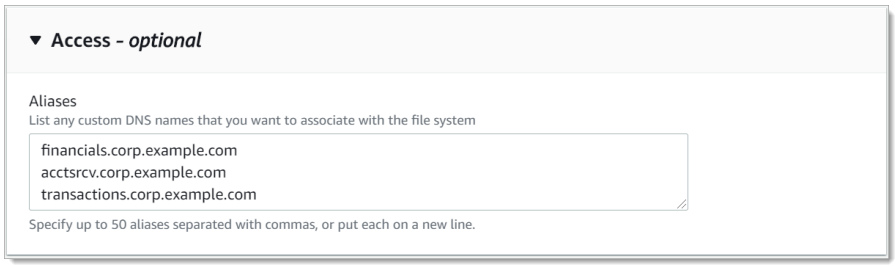
- Available – The DNS alias is associated with an Amazon FSx file system.
- Creating – Amazon FSx is creating the DNS alias and associating it with the file system.
- Deleting – Amazon FSx is disassociating the DNS alias from the file system and deleting it.
- Failed to create – Amazon FSx was unable to associate the DNS alias with the file system.
- Failed to delete – Amazon FSx was unable to disassociate the DNS alias from the file system.

Associating DNS aliases when creating a new file system

You can associate DNS aliases when creating a new file system from scratch, or when creating a file system from a backup.

To associate DNS aliases when creating a new Amazon FSx file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system described in Step 1: Create your file system (p. 7) in the Getting Started section.
3. In the **Access - optional** section of the **Create file system** wizard, enter the DNS aliases that you want to associate with your file system.



4. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

To associate DNS aliases when creating a new Amazon FSx file system (CLI)

1. When creating a new file system, use the Alias property with the CreateFileSystem API operation to associate DNS aliases with the new file system.

```
aws fsx create-file-system \
  --file-system-type WINDOWS \
  --storage-capacity 2000 \
  --storage-type SSD \
  --subnet-ids subnet-123456 \
  --windows-configuration Aliases=[financials.corp.example.com,accts-
rcv.corp.example.com]
```

2. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

To associate or disassociate DNS aliases when creating a new Amazon FSx file system from a backup (CLI)

1. When creating a new file system from a backup of an existing file system, you can use the Aliases property with the CreateFileSystemFromBackup API operation as follows:

   - Any aliases associated with the backup are associated with the new file system by default.
   - To create a file system without preserving any aliases from the backup, use the `Aliases` property with an empty set.

     To associate additional DNS aliases, use the `Aliases` property and include both the original aliases associated with the backup and the new aliases you want to associate.

   The following CLI command associates two aliases with the file system Amazon FSx is creating from a backup.

   ```
   aws fsx create-file-system-from-backup \
     --backup-id backup-0123456789abcdef0
     --storage-capacity 2000 \
     --storage-type HDD \
     --subnet-ids subnet-123456 \
     --windows-configuration Aliases=[transactions.corp.example.com,accts-
   rcv.corp.example.com]
   ```

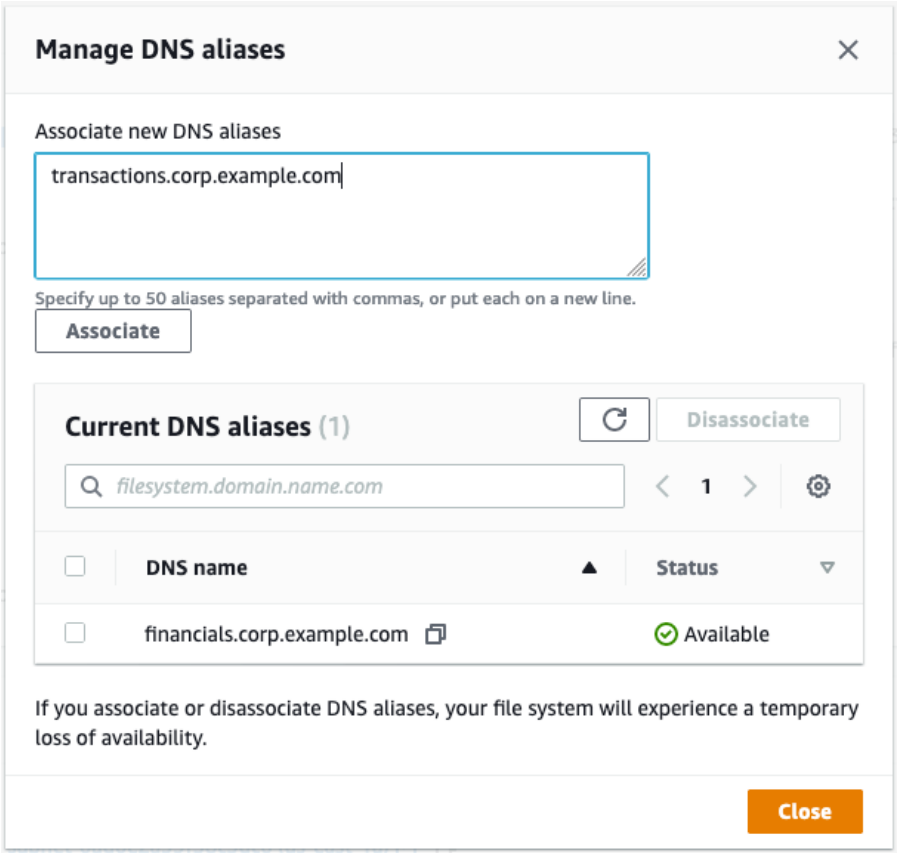
2. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

Managing DNS aliases on existing file systems

You can add and remove aliases on existing file systems.

To manage DNS aliases on an existing file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to manage DNS aliases for.
3. On the **Network & security** tab, choose **Manage** for **DNS aliases** to display the **Manage DNS aliases** dialog box.



- To associate DNS aliases – In the **Associate new aliases** box, enter the DNS aliases that you want to associate. Choose **Associate**.
- To disassociate DNS aliases – In the **Current aliases** list, choose the aliases to disassociate from. Choose **Disassociate**.

You can monitor the status of the aliases you have managed in the **Current aliases** list. Refresh the list to update the status. It takes up to 2.5 minutes for an alias to be associated or disassociated with a file system.

4. When the alias is **Available**, you can access your file system using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

To associate DNS aliases with an existing file system (CLI)

1. Use the `associate-file-system-aliases` CLI command or the AssociateFileSystemAliases API operation to associate DNS aliases with an existing file system.

   The following CLI request associates two aliases with the specified file system.

```
aws fsx associate-file-system-aliases \
  --file-system-id fs-0123456789abcdef0 \
  --aliases financials.corp.example.com transfers.corp.example.com
```

   The response shows the status of the aliases that Amazon FSx is associating with the file system.

```
{
    "Aliases": [
        {
            "Name": "financials.corp.example.com",
            "Lifecycle": CREATING
        },
        {
            "Name": "transfers.corp.example.com",
            "Lifecycle": CREATING
        }
    ]
}
```

2. Use the `describe-file-system-aliases` CLI command (DescribeFileSystemAliases is the equivalent API operation) to monitor the status of the aliases that you are associating.
3. When the `Lifecycle` has a value of AVAILABLE (a process that takes up to 2.5 minutes), you can access your file system using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Walkthrough 5: Using DNS aliases to access your file system (p. 166).

To disassociate DNS aliases from a file system (CLI)

- Use the `disassociate-file-system-aliases` CLI command or the DisassociateFileSystemAliases API operation to disassociate DNS aliases from an existing file system.

  The following command disassociates one alias from a file system.

```
aws fsx disassociate-file-system-aliases \
  --file-system-id fs-0123456789abcdef0 \
  --aliases financials.corp.example.com
```

  The response shows the status of the aliases that Amazon FSx is disassociating from the file system.

```
{
    "Aliases": [
        {
            "Name": "financials.corp.example.com",
            "Lifecycle": DELETING
        }
    ]
}
```

  Use the `describe-file-system-aliases` CLI command (DescribeFileSystemAliases is the equivalent API operation) to monitor the status of the aliases. It takes up to 2.5 minutes for the alias to be deleted.

File shares

You can manage file shares and perform the following tasks.

- Create a new file share
- Modify a file share
- Remove a file share

You can use the Windows-native Shared Folders GUI and the Amazon FSx CLI for remote management on PowerShell to manage file shares on your FSx for Windows File Server file system.

Warning
Amazon FSx requires that the SYSTEM user has **Full control** NTFS ACL permissions on every folder on which you create an SMB file share. Do not change the NTFS ACL permissions for this user on your folders, as doing so can make your file shares inaccessible.

Using the GUI to manage file shares

To manage file shares on your Amazon FSx file system, you can use the Shared Folders GUI. The Shared Folders GUI provides a central location for managing all shared folders on a Windows server. The following procedures detail how to manage your file shares.

To connect shared folders to your FSx file system

1. Launch your Amazon EC2 instance and connect it to the Microsoft Active Directory that your Amazon FSx file system is joined to. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:
   - Seamlessly join a Windows EC2 instance
   - Manually join a Windows instance
2. Connect to your instance as a user that is a member of the file system administrators group. In AWS Managed Microsoft Active Directory, this group is called AWS Delegated FSx Administrators. In your self-managed Microsoft Active Directory, this group is called Domain Admins or the custom name for the administrators group that you provided during creation. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Open the **Start** menu and run **fsmgmt.msc** using **Run As Administrator**. Doing this opens the Shared Folders GUI tool.
4. For **Action**, choose **Connect to another computer**.
5. For **Another computer**, enter the Domain Name System (DNS) name for your Amazon FSx file system, for example **`amznfsxabcd0123.corp.example.com`**.

   To find your file system's DNS name on the Amazon FSx console, choose **File systems**, choose your file system, and then check the **Network & Security** section of the file system details page. You can also get the DNS name in the response of the DescribeFileSystems API operation.
6. Choose **OK**. An entry for your Amazon FSx file system then appears in the list for the Shared Folders tool.

Now that Shared Folders is connected to your Amazon FSx file system, you can manage the Windows file shares on the file system. The default share is called `\share`. You can do so with the following actions:

- **Create a new file share** – In the Shared Folders tool, choose **Shares** in the left pane to see the active shares for your Amazon FSx file system. Choose **New Share** and complete the Create a Shared Folder wizard.

  You have to create the local folder prior to creating the new file share. You can do so as follows:
  - Using the Shared Folders tool: click on "Browse" when specifying local folder path and click on "Make new folder" to create the local folder.
  - Using command line:

```
New-Item -Type Directory -Path \\amznfsxabcd0123.corp.example.com\D$\share\MyNewShare
```

- **Modify a file share** – In the Shared Folders tool, open the context (right-click) menu for the file share that you want to modify in the right pane, and choose **Properties**. Modify the properties and choose **OK**.

- **Remove a file share** – In the Shared Folders tool, open the context (right-click) menu for the file share that you want to remove in the right pane, and then choose **Stop Sharing**.

  Note
  For Single-AZ 2 and Multi-AZ file systems, removing file shares or modifying file shares (including updating permissions, user limits, and other properties) using the Shared Folders GUI tool is possible only if you connect to **fsmgmt.msc** using the DNS Name of the Amazon FSx file system. The Shared Folders GUI tool does not support these actions if you connect using the IP address or DNS alias name of the file system.

Using PowerShell to manage file shares

You can manage file shares using custom remote-management commands for PowerShell. These commands can help you more easily automate these tasks:

- Migration of file shares on existing file servers to Amazon FSx
- Synchronization of file shares across AWS Regions for disaster recovery
- Programmatic management of file shares for ongoing workflows, such as team file-share provisioning

To learn how to use the Amazon FSx CLI for remote management on PowerShell, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Creating a continuously available share

You can create continuously available (CA) shares using the Amazon FSx CLI for Remote Management on PowerShell. CA shares created on an FSx for Windows File Server Multi-AZ file system are highly durable and highly available. An Amazon FSx Single-AZ file system is built on a single node cluster. As a result, CA shares created on a Single-AZ file system are highly durable, but are not highly available. Use the `New-FSxSmbShare` with the `-ContinuouslyAvailable` option set to `$True` to specify that the share is a continuously available share. The following is an example command to create a CA share.

```
New-FSxSmbShare -Name "New CA Share" -Path "D:\share\new-share" -Description "CA share" -
ContinuouslyAvailable $True
```

Following are custom remote-management PowerShell commands that you can use.

Share Management Command	Description
New-FSxSmbShare	Creates a new file share.
Remove-FSxSmbShare	Removes a file share.
Get-FSxSmbShare	Retrieves existing file shares.
Set-FSxSmbShare	Sets properties for a share.
Get-FSxSmbShareAccess	Retrieves the access control list (ACL) of a share.
Grant-FSxSmbShareAccess	Adds an allow access control entry (ACE) for a trustee to the security descriptor of a share.
Revoke-FSxSmbShareAccess	Removes all of the allow ACEs for a trustee from the security descriptor of a share.
Block-FSxSmbShareAccess	Adds a deny ACE for a trustee to the security descriptor of a share.

Share Management Command	Description
Unblock-FSxSmbShareAccess	Removes all of the deny ACEs for a trustee from the security descriptor of a share.

The online help for each command provides a reference of all command options. To access this help, run the command with a `-?`, for example `New-FSxSmbShare -?`.

Passing credentials to New-FSxSmbShare

You can pass credentials to New-FSxSmbShare so that you can run it in a loop to create hundreds or thousands of shares without having to re-enter credentials each time.

Prepare the credential object required to create the file shares on your FSx for Windows File Server file server using one of the following options.

- To generate the credential object interactively, use the following command.

```
$credential = Get-Credential
```

- To generate the credential object using an AWS Secrets Manager resource, use the following command.

```
$credential = ConvertFrom-Json -InputObject (Get-SECSecretValue -SecretId
 $AdminSecret).SecretString
$FSxAdminUserCredential = (New-Object PSCredential($credential.UserName,(ConvertTo-
SecureString $credential.Password -AsPlainText -Force)))
```

File access auditing

Amazon FSx for Windows File Server supports auditing of end-user accesses on files, folders, and file shares. You can choose to send the audit event logs to a rich set of other AWS services enabling querying, processing, storing and archiving logs, issuing notifications, and triggering actions to further advance your security and compliance goals.

Topics

File access auditing overview

File access auditing enables you to record end-user accesses of individual files, folders, and file shares based on your defined audit controls. Audit controls are also known as NTFS system access control lists (SACLs). If you already have audit controls set up on your existing file data, you can take advantage of file access auditing by creating a new Amazon FSx for Windows File Server file system and migrating your data.

Amazon FSx supports the following audit events provided by Windows for file, folder, and file share accesses:

- For file accesses, it supports: All, Traverse folder / Execute file, List folder / Read data, Read attributes, Create files / Write data, Create folders / Append data, Write attributes, Delete subfolders and files, Delete, Read permissions, Change permissions, and Take ownership.
- For file share accesses, it supports: Connect to a file share.

Across file, folder, and file share accesses, Amazon FSx supports logging of successful attempts (such as a user with sufficient permissions successfully accessing a file or file share), failed attempts, or both.

You can configure whether you want access auditing only on files and folders, only on file shares, or both. You can also configure which types of accesses should be logged (successful attempts only, failed attempts only, or both). You can also turn off file access auditing at any time.

> **Note**
> File access auditing records end-user access data only from the time it was enabled. That is, file access auditing doesn't generate audit event logs of end-user file, folder, and file share access activity that occurred before file access auditing was enabled.

The maximum rate of access audit events supported is 5,000 events per second. Access audit events are not generated for each file read and write operation, but generated once per file metadata operation, such as when a user creates, opens, or deletes a file.

Audit event log destinations

When enabled, the file access auditing feature must have a configured AWS service to which Amazon FSx sends the audit event logs. This audit event log destination must be either an Amazon CloudWatch Logs log stream in a CloudWatch Logs log group or an Amazon Kinesis Data Firehose delivery stream. You can choose the audit event logs destination when you create your Amazon FSx for Windows File Server file system or afterwards by updating it. For more information, see Managing file access auditing (p. 101).

Following are some recommendations that may help you decide which audit event logs destination to choose:

- Choose CloudWatch Logs if you want to store, view, and search audit event logs in the Amazon CloudWatch console, run queries on the logs using CloudWatch Logs Insights, and trigger CloudWatch alarms or Lambda functions.
- Choose Kinesis Data Firehose if you want to continuously stream events to storage in Amazon S3, to a database in Amazon Redshift, to Amazon OpenSearch Service, or to AWS Partner solutions (such as Splunk or Datadog) for further analysis.

By default, Amazon FSx will create and use a default CloudWatch Logs log group in your account as the audit event log destination. If you want to use a custom CloudWatch Logs log group or use Kinesis Data Firehose as the audit event log destination, here are the requirements for the names and locations of the audit event log destination:

- The name of the CloudWatch Logs log group must begin with the `/aws/fsx/` prefix. If you don't have an existing CloudWatch Logs log group when you create or update a file system on the console, Amazon FSx can create and use a default log stream in the CloudWatch Logs `/aws/fsx/windows` log group. If you don't want to use the default log group, the configuration UI lets you create a CloudWatch Logs log group when you create or update your file system on the console.
- The name of the Kinesis Data Firehose delivery stream must begin with the `aws-fsx-` prefix. If you don't have an existing Kinesis Data Firehose delivery stream, you can create one when you create or update your file system at the console.

- The Kinesis Data Firehose delivery stream must be configured to use `Direct PUT` as its source. You cannot use an existing Kinesis data stream as a data source for your delivery stream.
- The destination (either CloudWatch Logs log group or Kinesis Data Firehose delivery stream) must be in the same AWS partition, AWS Region, and AWS account as your Amazon FSx file system.

You can change the audit event log destination at any time (for example, from CloudWatch Logs to Kinesis Data Firehose). When you do so, new audit event logs are sent only to the new destination.

Best effort audit event log delivery

Typically, audit event log records are delivered in minutes, but can sometimes take longer. On very rare occasions, audit event log records might be missed. If your use case requires particular semantics (for example, ensuring that no audit events are missed), we recommend that you account for missed events when designing your workflows. You can audit for missed events by scanning the file and folder structure on your file system.

Auditing access to files and folders

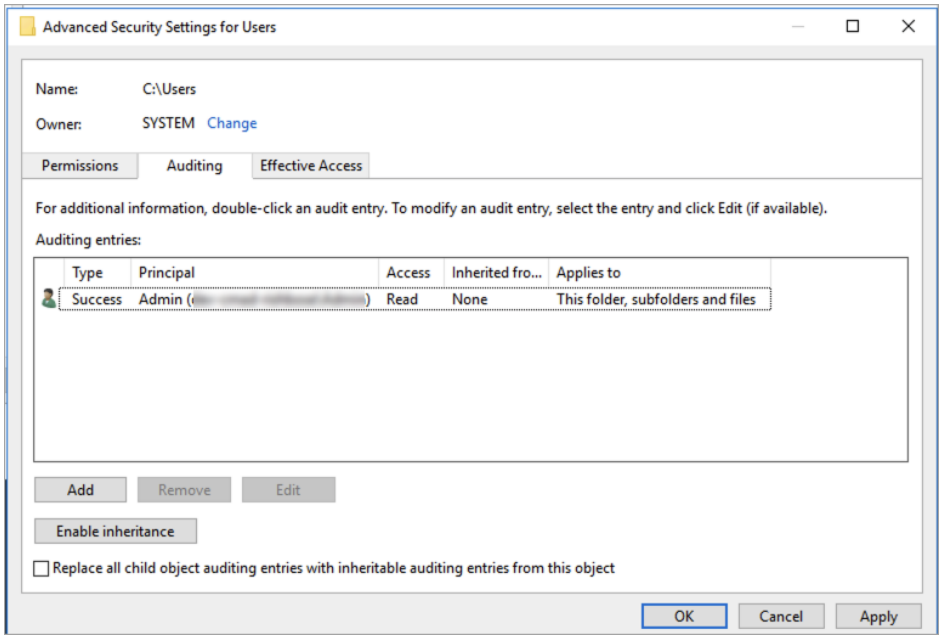
You need to set audit controls on the files and folders that you want audited for user access attempts. Audit controls are also known as NTFS system access control lists (SACLs).

You configure audit controls using the Windows-native GUI interface or programmatically using Windows PowerShell commands. If inheritance is enabled, you typically need to set audit controls only on the top-level folders you want to log accesses for.

Using the Windows GUI to set auditing access

To use a GUI for setting audit controls on your files and folders, use Windows File Explorer. On a given file or folder, open Windows File Explorer and select the **Properties > Security > Advanced > Auditing** tab.

The following audit control example audits successful events for a folder. A Windows event log entry will be emitted whenever that handle is opened for read successfully by the Admin user.



The **Type** field indicates what actions you want to audit. Set this field to **Success** to audit successful attempts, **Fail** to audit failed attempts, or **All** to audit both successful and failed attempts.

For more information on the auditing entry fields, see Apply a basic audit policy on a file or folder in the Microsoft documentation.

Using PowerShell commands to set auditing access

You can use the Microsoft Windows `Set-Acl` command to set the auditing SACL on any file or folder. For information about this command, see the Microsoft Set-Acl documentation.

Following is an example of using a series of PowerShell commands and variables to set auditing access for successful attempts. You can adapt these example commands to fit the needs on your file system.

```
$path = "C:\Users\TestUser\Desktop\DemoTest\"

$ACL = Get-Acl $path

$ACL | Format-List

$AuditUser = "TESTDOMAIN\TestUser"

$AuditRules = "FullControl"

$InheritType = "ContainerInherit,ObjectInherit"

$AuditType = "Success"

$AccessRule = New-Object System.Security.AccessControl.FileSystemAuditRule($AuditUser,
$AuditRules,$InheritType,"None",$AuditType)

$ACL.SetAuditRule($AccessRule)

$ACL | Set-Acl $path

Get-Acl $path -Audit | Format-List
```

Managing file access auditing

You can enable file access auditing when creating a new Amazon FSx for Windows File Server file system. File access auditing is turned off by default when you create a file system from the Amazon FSx console.

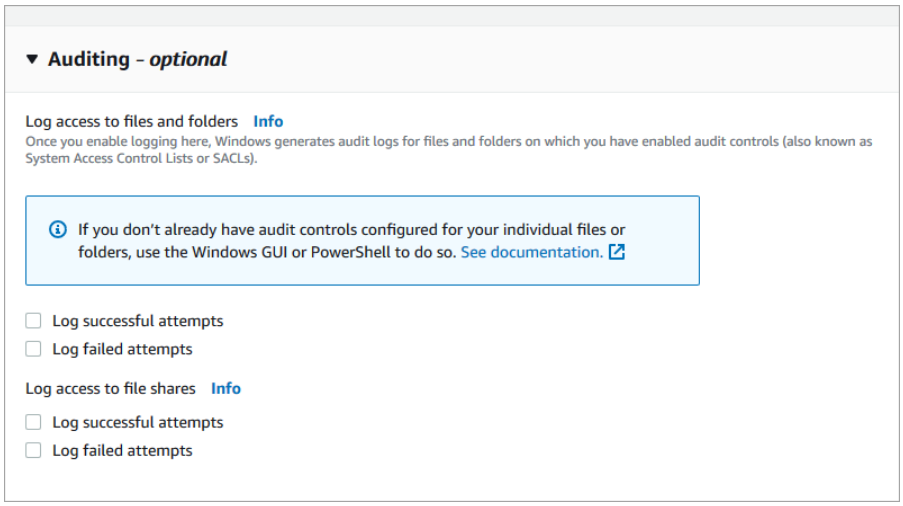
On existing file systems that have file access auditing enabled, you can change the file access auditing settings, including changing the access attempt types for file and file share accesses, and the audit event log destination. You can perform these tasks using the Amazon FSx console, AWS CLI, or API.

> **Note**
> File access auditing is supported only on Amazon FSx for Windows File Server file systems with a throughput capacity of 32 MB/s or greater. You cannot create or update a file system with a throughput capacity of less than 32 MB/s if file access auditing is enabled. You can modify the throughput capacity at any time after you create the file system. For more information, see Managing throughput capacity (p. 133).

To enable file access auditing when creating a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system described in Step 1: Create your file system (p. 7) in the Getting Started section.

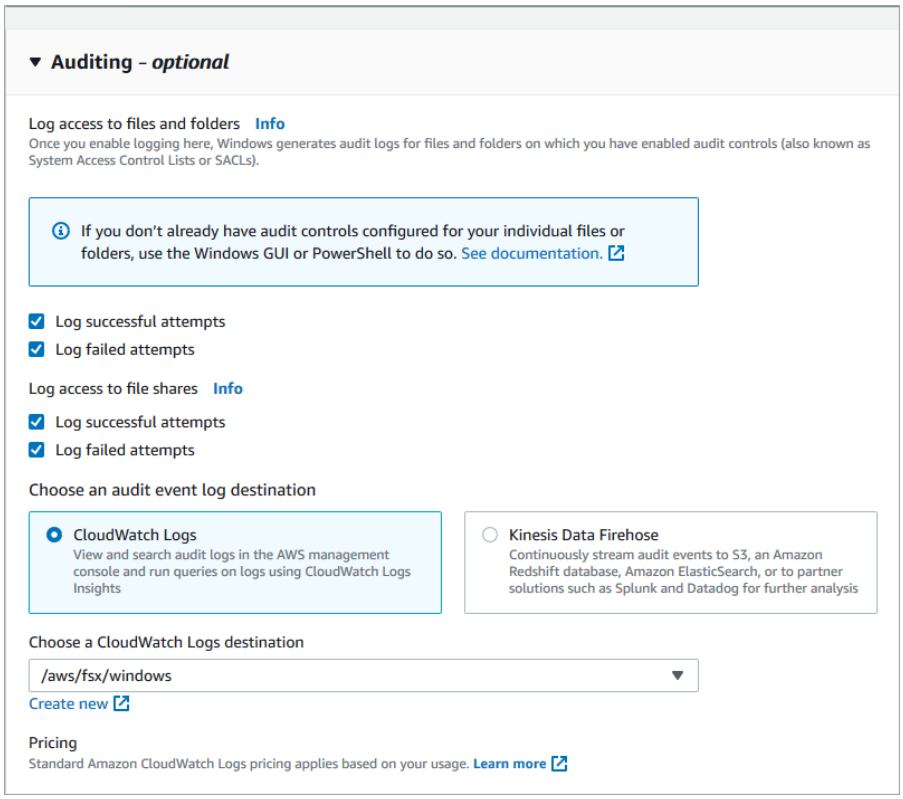
3. Open the **Auditing - optional** section. File access auditing is disabled by default.



4. To enable and configure file access auditing, do the following.

   - For **Log access to files and folders**, select the logging of successful and/or failed attempts. Logging is disabled for files and folders if you don't make a selection.
   - For **Log access to file shares**, select the logging of successful and/or failed attempts. Logging is disabled for file shares if you don't make a selection.
   - For **Choose an audit event log destination**, choose **CloudWatch Logs** or **Kinesis Data Firehose**. Then choose an existing log or delivery stream or create a new one. For CloudWatch Logs, Amazon FSx can create and use a default log stream in the CloudWatch Logs `/aws/fsx/windows` log group.

   Following is an example of a file access auditing configuration that will audit successful and failed access attempts of end users for files, folders, and file shares. The audit event logs will be sent to the default CloudWatch Logs `/aws/fsx/windows` log group destination.



5. Continue with the next section of the file system creation wizard.

When the file system is **Available**, the file access auditing feature is enabled.

To enable file access auditing when creating a file system (CLI)

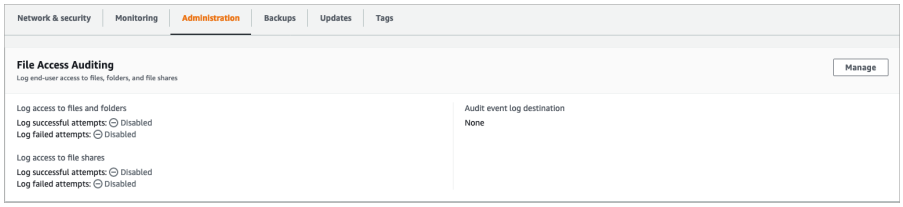
1. When creating a new file system, use the `AuditLogConfiguration` property with the CreateFileSystem API operation to enable file access auditing for the new file system.

```
aws fsx create-file-system \
  --file-system-type WINDOWS \
  --storage-capacity 300 \
  --subnet-ids subnet-123456 \
  --windows-configuration
 AuditLogConfiguration='{FileAccessAuditLogLevel="SUCCESS_AND_FAILURE", \
    FileShareAccessAuditLogLevel="SUCCESS_AND_FAILURE", \
    AuditLogDestination="arn:aws:logs:us-east-1:123456789012:log-group:/aws/fsx/my-
customer-log-group"}'
```

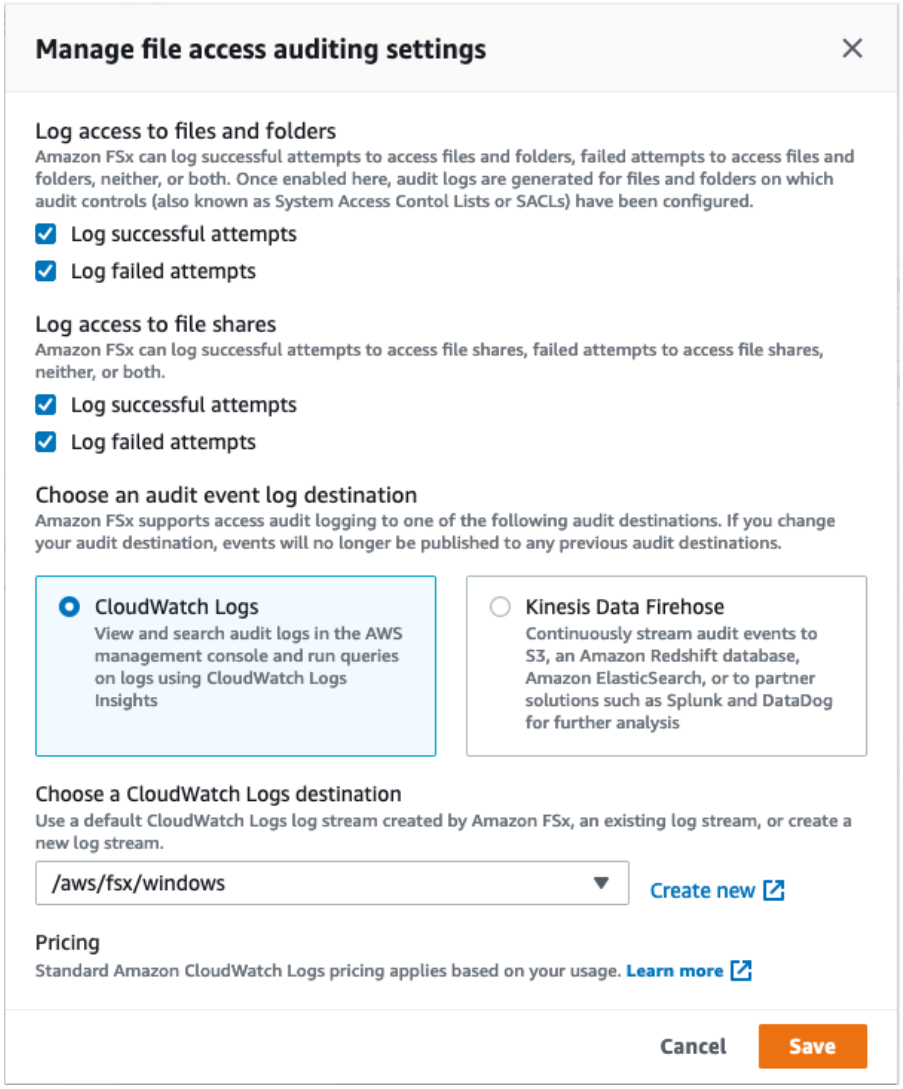
2. When the file system is **Available**, the file access auditing feature is enabled.

To change the file access auditing configuration (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to manage file access auditing for.
3. Choose the **Administration** tab.
4. On the **File Access Auditing** panel, choose **Manage**.



5. On the **Manage file access auditing settings** dialog, change the desired settings.



   - For **Log access to files and folders**, select the logging of successful and/or failed attempts. Logging is disabled for files and folders if you don't make a selection.
   - For **Log access to file shares**, select the logging of successful and/or failed attempts. Logging is disabled for file shares if you don't make a selection.
   - For **Choose an audit event log destination**, choose **CloudWatch Logs** or **Kinesis Data Firehose**. Then choose an existing log or delivery stream or create a new one.
6. Choose **Save**.

To change the file access auditing configuration (CLI)

- Use the `update-file-system` CLI command or the equivalent `UpdateFileSystem` API operation.

```
aws fsx update-file-system \
  --file-system-id fs-0123456789abcdef0 \
  --windows-configuration
 AuditLogConfiguration='{FileAccessAuditLogLevel="SUCCESS_ONLY", \
    FileShareAccessAuditLogLevel="FAILURE_ONLY", \
    AuditLogDestination="arn:aws:logs:us-east-1:123456789012:log-group:/aws/fsx/my-
customer-log-group"}'
```

Migrating your audit controls

If you have audit controls (SACLs) already set up on your existing file data, you can create an Amazon FSx file system and migrate your data to your new file system. We recommend using AWS DataSync to transfer data and the associated SACLs to your Amazon FSx file system. As an alternative solution, you can use Robocopy (Robust File Copy). For more information, see Migrating existing file storage to Amazon FSx (p. 63).

Viewing event logs

You can view the audit event logs after Amazon FSx has started emitting them. Where and how you view the logs depends on the audit event log destination:

- You can view CloudWatch Logs logs by going to the CloudWatch console and choosing the log group and log stream to which your audit event logs are sent. For more information, see View log data sent to CloudWatch Logs in the *Amazon CloudWatch Logs User Guide*.

  You can use CloudWatch Logs Insights to interactively search and analyze your log data. For more information, see Analyzing Log Data with CloudWatch Logs Insights, in the *Amazon CloudWatch Logs User Guide*.

  You can also export the audit event logs to Amazon S3. For more information, see Exporting Log Data to Amazon S3, also in the *Amazon CloudWatch Logs User Guide*.
- You can't view the audit event logs on Kinesis Data Firehose. However, you can configure Kinesis Data Firehose to forward the logs to a destination that you can read from. The destinations include Amazon S3, Amazon Redshift, Amazon OpenSearch Service, and partner solutions such as Splunk and Datadog, For more information, see Choose destination in the *Amazon Kinesis Data Firehose Developer Guide*.

Audit event fields

This section provides descriptions of the information in audit event logs and examples of audit events.

Following are descriptions of the salient fields in a Windows audit event.

- **EventID** refers to the Microsoft-defined Windows event log event ID. See Microsoft documentation for information on file system events and file share events.
- **SubjectUserName** refers to the user performing the access.
- **ObjectName** refers to the target file, folder, or file share that was accessed.
- **ShareName** is available for events that are generated for file share access. For example, `EventID 5140` is generated when a network share object was accessed.
- **IpAddress** refers to the client that initiated the event for file share events.

- **Keywords**, when available, refer to whether the file access was successful or a failure. For successful accesses, the value is `0x8020000000000000`. For failed accesses, the value is `0x8010000000000000`.
- **TimeCreated SystemTime** refers to the time the event was generated in the system and shown in <YYYY-MM-DDThh:mm:ss.s>Z format.
- **Computer** refers to the DNS name of the file system Windows Remote PowerShell Endpoint and can be used to identify the file system.
- **AccessMask**, when available, refers to the type of file access performed (for example, ReadData, WriteData).
- **AccessList** refers to requested or granted access to an Object. For details, see the table below and Microsoft documentation (such as in Event 4556).

Access Type	Access Mask	Value
Read Data or List Directory	0x1	%%4416
Write Data or Add File	0x2	%%4417
Append Data or Add Subdirectory	0x4	%%4418
Read Extended Attributes	0x8	%%4419
Write Extended Attributes	0x10	%%4420
Execute/Traverse	0x20	%%4421
Delete Child	0x40	%%4422
Read Attributes	0x80	%%4423
Write Attributes	0x100	%%4424
Delete	0x10000	%%1537
Read ACL	0x20000	%%1538
Write ACL	0x40000	%%1539
Write Owner	0x80000	%%1540
Synchronize	0x100000	%%1541
Access Security ACL	0x1000000	%%1542

Following are some key events with examples. Note that the XML is formatted for readability.

Event ID 4660 is logged when an object is deleted.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4660</EventID><Version>0</Version><Level>0</Level>
<Task>12800</Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-05-18T04:51:56.916563800Z'/>
<EventRecordID>315452</EventRecordID><Correlation/>
<Execution ProcessID='4' ThreadID='5636'/><Channel>Security</Channel>
<Computer>amznfsxgyzohmw8.example.com</Computer><Security/></System><EventData>
```

```
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x50932f71</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='HandleId'>0x12e0</Data><Data Name='ProcessId'>0x4</Data><Data
 Name='ProcessName'></Data>
<Data Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data></EventData></
Event>
```

Event ID 4659 is logged on a request to delete a file.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4659</EventID><Version>0</Version><Level>0</Level><Task>12800</Task><Opcode>0</
Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-0603T19:18:09.951551200Z'/>
<EventRecordID>308888</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='5540'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/></
System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\shar
\event.txt</Data>
<Data Name='HandleId'>0x0</Data><Data
 Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data>
<Data Name='AccessList'>%%1537
    %%4423
    </Data><Data Name='AccessMask'>0x10080</Data><Data Name='PrivilegeList'>-</Data>
<Data Name='ProcessId'>0x4</Data></EventData></Event>
```

Event ID 4663 is logged when a specific operation was performed on the object. The following example shows reading data from a file, which can be interpreted from `AccessList %%4416`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4663< /EventID><Version>1</Version><Level>0</Level><Task>12800</Task><Opcode>0</
Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:10:13.887145400Z'/>
<EventRecordID>308831</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='6916'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/></
System>
<EventData>< Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113< /
Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\share
\event.txt</Data>
<Data Name='HandleId'>0x101c</Data><Data Name='AccessList'>%%4416
    </Data>
<Data Name='AccessMask'>0x1</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data><Data Name='ResourceAttributes'>S:AI</Data>
</EventData></Event>
```

The following example shows write/append data from a file, which can be interpreted from `AccessList %%4417`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4663</EventID><Version>1</Version><Level>0</Level><Task>12800</Task><Opcode>0</
Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:12:16.813827100Z'/>
<EventRecordID>308838</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='5828'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/></
System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\share
\event.txt</Data>
<Data Name='HandleId'>0xa38</Data><Data Name='AccessList'>%%4417
    </Data><Data Name='AccessMask'>0x2</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data><Data Name='ResourceAttributes'>S:AI</Data></EventData></
Event>
```

Event ID 4656 indicates that a specific access was requested for an object. In the following example, the Read request was initiated to ObjectName "permtest" and was a failed attempt, as seen in the Keywords value of `0x8010000000000000`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4656</EventID><Version>1</Version><Level>0</Level><Task>12800</Task><Opcode>0</
Opcode>
<Keywords>0x8010000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:22:55.113783500Z'/>
<EventRecordID>308919</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='4924'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/></
System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\share
\permtest</Data>
<Data Name='HandleId'>0x0</Data><Data
 Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data>
<Data Name='AccessList'>%%1541
    %%4416
    %%4423
    </Data><Data Name='AccessReason'>%%1541: %%1805
    %%4416: %%1805
    %%4423: %%1811 D:(A;OICI;0x1301bf;;;AU)
    </Data><Data Name='AccessMask'>0x100081</Data><Data Name='PrivilegeList'>-</Data>
<Data Name='RestrictedSidCount'>0</Data><Data Name='ProcessId'>0x4</Data><Data
 Name='ProcessName'></Data>
<Data Name='ResourceAttributes'>-</Data></EventData></Event>
```

Event ID 4670 is logged when permissions for an object are changed. The following example shows that user "admin" modified the permission on ObjectName "permtest" to add permissions to SID "S-1-5-21-658495921-4185342820-3824891517-1113". Refer to Microsoft documentation for more information on how to interpret the permissions.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
```

```
<EventID>4670</EventID><Version>0</Version><Level>0</Level>
<Task>13570</Task><Opcode>0</Opcode><Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime='2021-06-03T19:39:47.537129500Z'/><EventRecordID>308992</
EventRecordID>
<Correlation/><Execution ProcessID='4' ThreadID='2776'/><Channel>Security</Channel>
<Computer>amznfsxgyzohmw8.example.com</Computer><Security/></System><EventData>
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\share
\permtest</Data>
<Data Name='HandleId'>0xcc8</Data>
<Data Name='OldSd'>D:PAI(A;OICI;FA;;;SY)
(A;OICI;FA;;;S-1-5-21-658495921-4185342820-3824891517-2622)</Data>
<Data Name='NewSd'>D:PARAI(A;OICI;FA;;;S-1-5-21-658495921-4185342820-3824891517-1113)
(A;OICI;FA;;;SY)(A;OICI;FA;;;
S-1-5-21-658495921-4185342820-3824891517-2622)</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data></EventData></Event>
```

Event ID 5140 is logged every time a file share is accessed.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>5140</EventID><Version>1</Version><Level>0</Level><Task>12808</Task><Opcode>0</
Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:32:07.535208200Z'/>
<EventRecordID>308947</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='3120'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/></
System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-2620</Data>
<Data Name='SubjectUserName'>EC2AMAZ-1GP4HMN$</Data><Data
 Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2d4ca529</Data><Data Name='ObjectType'>File</Data><Data
 Name='IpAddress'>172.45.6.789</Data>
<Data Name='IpPort'>49730</Data><Data Name='ShareName'>\\AMZNFSXCYDKLDZZ\share</Data>
<Data Name='ShareLocalPath'>\??\D:\share</Data><Data Name='AccessMask'>0x1</Data><Data
 Name='AccessList'>%%4416
    </Data></EventData></Event>
```

Event ID 5145 is logged when access is denied at the file share level. The following example shows access to ShareName "demoshare01" was denied.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>5145</EventID><Version>0</Version><Level>0</Level>
<Task>12811</Task><Opcode>0</Opcode><Keywords>0x8010000000000000</Keywords>
<TimeCreated SystemTime='2021-05-19T22:30:40.485188700Z'/><EventRecordID>282939</
EventRecordID>
<Correlation/><Execution ProcessID='4' ThreadID='344'/><Channel>Security</Channel>
<Computer>amznfsxtmn9autz.example.com</Computer><Security/></System><EventData>
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-
1113</Data><Data Name='SubjectUserName'>Admin</Data><Data
 Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x95b3fb7</Data><Data Name='ObjectType'>File</Data>
<Data Name='IpAddress'>172.31.7.112</Data><Data Name='IpPort'>59979</Data>
<Data Name='ShareName'>\\AMZNFSXDPNTE0DC\demoshare01</Data><Data Name='ShareLocalPath'>\??
\D:\demoshare01</Data>
<Data Name='RelativeTargetName'>Desktop.ini</Data><Data Name='AccessMask'>0x120089</Data>
```

```
<Data Name='AccessList'>%%1538 %%1541 %%4416 %%4419 %%4423 </Data><Data
 Name='AccessReason'>%%1538:
%%1804 %%1541: %%1805 %%4416: %%1805 %%4419: %%1805 %%4423: %%1805 </Data></EventData></
Event>
```

If you use CloudWatch Logs Insights to search your log data, you can run queries on the event fields, as shown by the following examples:

- To query for a specific event ID:

```
fields @message
   | filter @message like /4660/
```

- To query all events matching a particular file name:

```
fields @message
   | filter @message like /event.txt/
```

For more information on the CloudWatch Logs Insights query language, see Analyzing Log Data with CloudWatch Logs Insights, in the *Amazon CloudWatch Logs User Guide*.

User sessions and open files

You can monitor connected user sessions and open files on your FSx for Windows File Server file system using the Shared Folders tool. The Shared Folders tool provides a central location to monitor who is connected to the file system, along with what files are opened and by whom. You can use this tool to do the following:

- Restore access to locked files.
- Disconnect a user session, which closes all files opened by that user.

You can use the Windows-native Shared Folders GUI tool and the Amazon FSx CLI for remote management on PowerShell to manage user sessions and open files on your FSx for Windows File Server file system.

Using the GUI to manage users and sessions

The following procedures detail how you can manage user sessions and open files on your Amazon FSx file system.

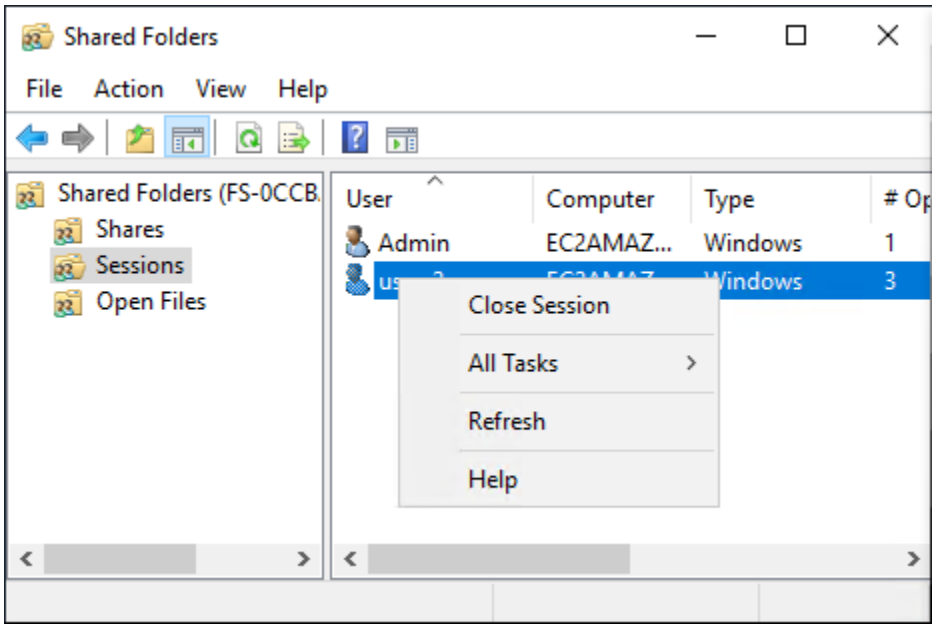
To launch the shared folders tool

1. Launch your Amazon EC2 instance and connect it to the Microsoft Active Directory that your Amazon FSx file system is joined to. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:
   - Seamlessly join a Windows EC2 instance
   - Manually join a Windows instance
2. Connect to your instance as a user that is a member of the file system administrators group. In AWS Managed Microsoft Active Directory, this group is called AWS Delegated FSx Administrators. In your self-managed Microsoft Active Directory, this group is called Domain Admins or the custom name for the administrators group that you provided during creation. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.

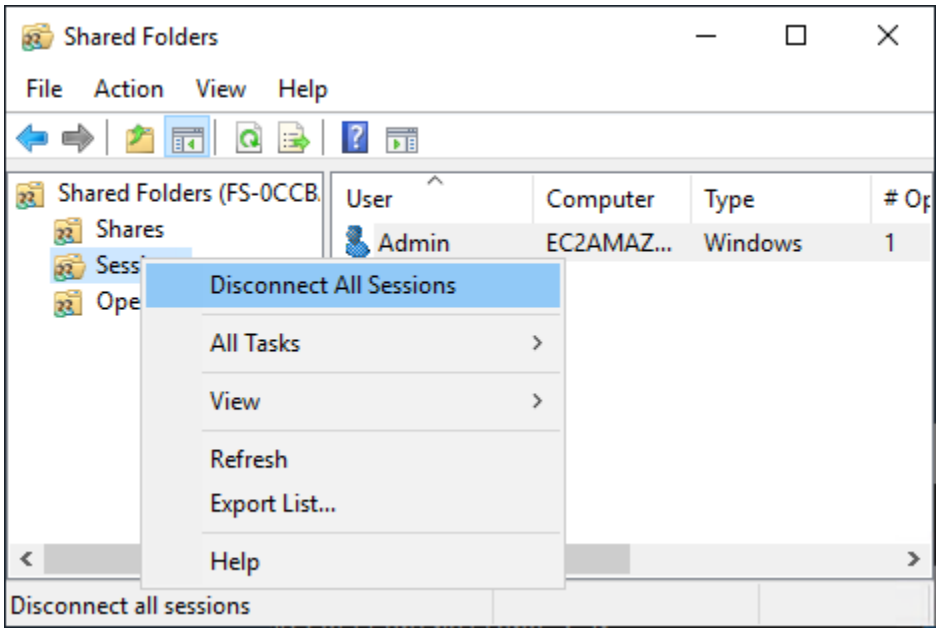
3. Open the **Start** menu and run **fsmgmt.msc** using `Run As Administrator`. Doing this opens the Shared Folders GUI tool.
4. For **Action**, choose **Connect to another computer**.
5. For **Another computer**, enter the DNS name of your Amazon FSx file system, for example `fs-01234567890123456.ad-domain.com`.
6. Choose **OK**. An entry for your Amazon FSx file system then appears in the list for the Shared Folders tool.

Managing user sessions

In the Shared Folders tool, choose **Sessions** to view all the user sessions that are connected to your FSx for Windows File Server file system. If a user or application is accessing a file share on your Amazon FSx file system, this snap-in shows you their session. You can disconnect sessions by opening the context (right-click) menu for a session and choosing **Close Session**.

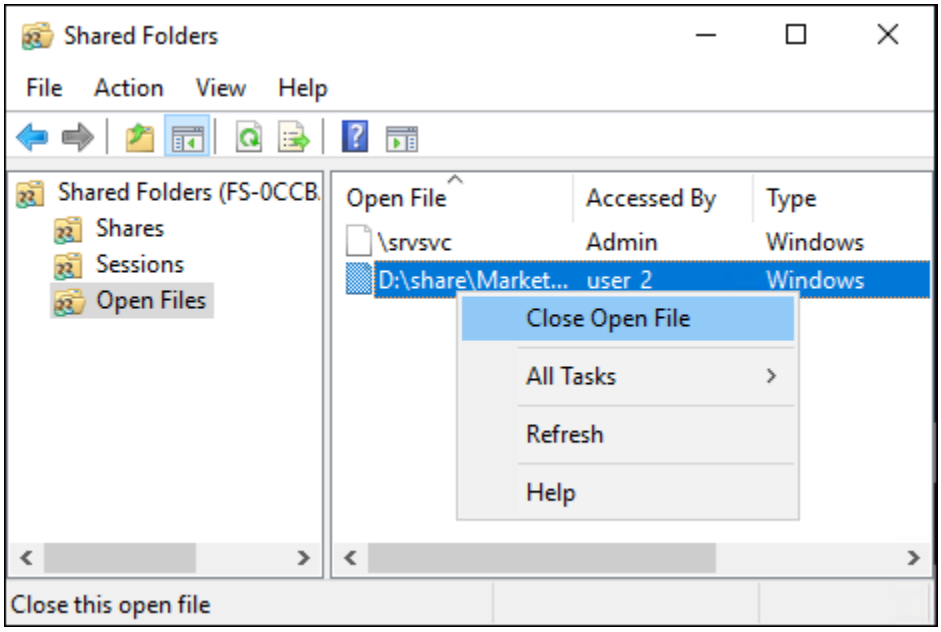


To disconnect all open sessions, open the context (right-click) menu for **Sessions**, choose **Disconnect All Sessions**, and confirm your action.

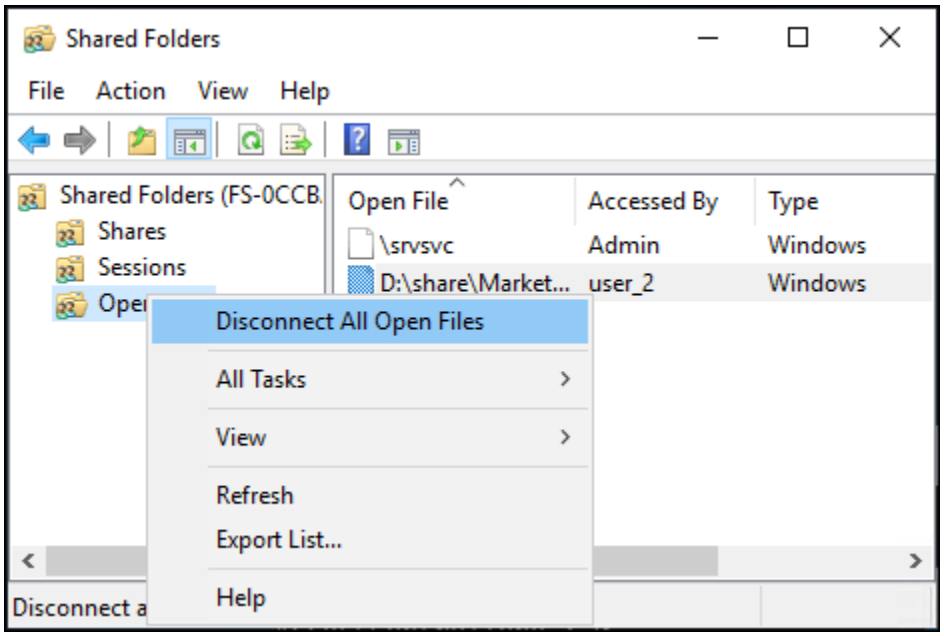


Managing open files

In the Shared Folders tool, choose **Open Files** to view all the files on the system that are currently open. The view also shows which users have the files or folders open. This information can be helpful in tracking down why other users cannot open certain files. You can close any file that any user has open simply by opening the context (right-click) menu for the file's entry in the list and choosing **Close Open File**.



To disconnect all open files on the file system, the context (right-click) menu for **Open Files** and choose **Disconnect All Open Files**, and confirm your action.



Using PowerShell to manage user sessions and open files

You can manage active user sessions and open files on your file system using the Amazon FSx CLI for remote management on PowerShell. To learn how to use this CLI, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Following are commands that you can use for user session and open file management.

Command	Description
Get-FSxSmbSession	Retrieves information about the Server Message Block (SMB) sessions that are currently established between the file system and the associated clients.
Close-FSxSmbSession	Ends an SMB session.
Get-FSxSmbOpenFile	Retrieves information about files that are open for the clients connected to the file system.
Close-FSxSmbOpenFile	Closes a file that is open for one of the clients of the SMB server.

The online help for each command provides a reference of all command options. To access this help, run the command with a **-?**, for example **Get-FSxSmbSession -?**.

Data deduplication

Large datasets often have redundant data, which increases the data storage costs. For example, with user file shares, multiple users can store many copies or versions of the same file. With software development shares, many binaries remain unchanged from build to build.

You can reduce your data storage costs by turning on data deduplication for your file system. *Data deduplication* reduces or eliminates redundant data by storing duplicated portions of the dataset only once. Data compression is enabled by default when you use data deduplication, further reducing the amount of data storage by compressing the data after deduplication. Data deduplication runs as a background process that continually and automatically scans and optimizes your file system, and it is transparent to your users and connected clients.

The storage savings that you can achieve with data deduplication depends on the nature of your dataset, including how much duplication exists across files. Typical savings average 50–60 percent for general-purpose file shares. Within shares, savings range from 30–50 percent for user documents to 70–80 percent for software development datasets. You can measure potential deduplication savings using the `Measure-FSxDedupFileMetadata` command described below.

You can also customize data deduplication to meet your specific storage needs. For example, you can configure deduplication to run only on certain file types, or you can create a custom job schedule. Because deduplication jobs can consume file server resources, we recommend monitoring the status of your deduplication jobs using the `Get-FSxDedupStatus` command described below.

For more information about data deduplication, see the Microsoft Understanding Data Deduplication documentation.

Note
If you encounter issues with getting data deduplication jobs to run successfully, see Troubleshooting data deduplication (p. 220).

Enabling data deduplication

You enable data deduplication on an Amazon FSx for Windows File Server file share using the `Enable-FSxDedup` command, as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

When you enable data deduplication, a default schedule and configuration are created. You can create, modify, and remove schedules and configurations using the commands below.

Note that creating new, custom deduplication job schedules does not override or remove the existing default schedule. Before creating a custom deduplication job, you may want to disable the default job if you don't need it.

You can use the `Disable-FSxDedup` command to disable data deduplication entirely on your file system.

Note
When you increase a file system's storage capacity, Amazon FSx cancels existing data deduplication jobs during the storage optimization process that migrates data from the old disks to the new, larger disks. During this period, the `OptimizedFilesSavingsRate` value is 0. Amazon FSx resumes data deduplication once the storage capacity increase optimization job completes. For more information about increasing storage capacity and storage optimization, see Managing storage capacity (p. 123).

Creating a data deduplication schedule

Even though the default schedule works well in most cases, you can create a new deduplication schedule by using the `New-FsxDedupSchedule` command, shown as follows. Data deduplication schedules use UTC time.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
New-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days Mon,Wed,Sat -Start
 08:00 -DurationHours 7
}
```

This command creates a schedule named `CustomOptimization` that runs on days Monday, Wednesday, and Saturday, starting the job at 8:00 am (UTC) each day, with a maximum duration of 7 hours, after which the job stops if it is still running.

Modifying a data deduplication schedule

You can modify an existing deduplication schedule by using the `Set-FsxDedupSchedule` command, shown as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days Mon,Tues,Wed,Sat -
Start 09:00 -DurationHours 9
}
```

This command modifies the existing `CustomOptimization` schedule to run on days Monday to Wednesday and Saturday, starting the job at 9:00 am (UTC) each day, with a maximum duration of 9 hours, after which the job stops if it is still running.

To modify the minimum file age before optimizing setting, use the `Set-FSxDedupConfiguration` command.

Viewing the amount of saved space

To view the amount of disk space you are saving from running data deduplication, use the `Get-FSxDedupStatus` command, as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FsxRemoteAdmin -ScriptBlock {
Get-FSxDedupStatus } | select
 OptimizedFilesCount,OptimizedFilesSize,SavedSpace,OptimizedFilesSavingsRate

OptimizedFilesCount OptimizedFilesSize SavedSpace OptimizedFilesSavingsRate
------------------- ------------------ ---------- -------------------------
              12587           31163594   25944826                        83
```

Note
The values shown in the command response for following parameters are not reliable, and you should not use these values: Capacity, FreeSpace, UsedSpace, UnoptimizedSize, and SavingsRate.

Managing data deduplication

You can manage data deduplication on your file system using the Amazon FSx CLI for remote management on PowerShell. To learn how to use this CLI, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Following are commands that you can use for data deduplication.

Data deduplication command	Description
Enable-FSxDedup	Enables data deduplication on the file share. Data compression after deduplication is enabled by default when you enable data deduplication.
Disable-FSxDedup	Disables data deduplication on the file share.
Get-FSxDedupConfiguration	Retrieves deduplication configuration information, including Minimum file size and age for optimization, compression settings, and Excluded file types and folders.
Set-FSxDedupConfiguration	Changes the deduplication configuration settings, including minimum file size and age for optimization, compression settings, and excluded file types and folders.
Get-FSxDedupStatus	Retrieves the deduplication status, and includes read-only properties that describe optimization savings and status on the file system, times, and completion status for the last jobs on the file system.
Get-FSxDedupMetadata	Retrieves deduplication optimization metadata.
Update-FSxDedupStatus	Computes and retrieves updated data deduplication savings information.
Measure-FSxDedupFileMetadata	Measures and retrieves the potential storage space that you can reclaim on your file system if you delete a group of folders. Files often have chunks that are shared across other folders, and the deduplication engine calculates which chunks are unique and would be deleted.
Get-FSxDedupSchedule	Retrieves deduplication schedules that are currently defined.
New-FSxDedupSchedule	Creates and customizes a data deduplication schedule.
Set-FSxDedupSchedule	Changes configuration settings for existing data deduplication schedules.
Remove-FSxDedupSchedule	Deletes a deduplication schedule.
Get-FSxDedupJob	Gets status and information for all currently running or queued deduplication jobs.
Stop-FSxDedupJob	Cancel one or more specified data deduplication jobs.

The online help for each command provides a reference of all command options. To access this help, run the command with **-?**, for example **Enable-FSxDedup -?**.

Storage quotas

You can configure user storage quotas on your file systems to limit how much data storage that users can consume. After you set quotas, you can track quota status to monitor usage and see when users surpass their quotas.

You can also enforce quotas by stopping users who reach their quotas from writing to the storage space. When you enforce quotas, a user that exceeds their quota receives an "insufficient disk space" error message.

You can set these thresholds for quota settings:

- Warning - used to track whether a user or group is approaching their quota limit, relevant for tracking only.
- Limit - the storage quota limit for a user or group.

You can configure default quotas that are applied to new users who access a file system and quotas that apply to specific users or groups. You can also view a report of how much storage each user or group is consuming and whether they're surpassing their quotas.

Storage consumption at a user level is tracked based on file ownership. Storage consumption is calculated using logical file size, not the actual physical storage space that files occupy. User storage quotas are tracked at the time when data is written to a file.

Managing user storage quotas

You can manage user storage quotas on your file system using the Amazon FSx CLI for remote management on PowerShell. To learn how to use this CLI, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Following are commands that you can use to manage user storage quotas.

User storage quotas command	Description
Enable-FSxUserQuotas	Starts tracking or enforcing user storage quotas, or both.
Disable-FSxUserQuotas	Stops tracking and enforcement for user storage quotas.
Get-FSxUserQuotaSettings	Retrieves the current user-storage quota settings for the file system.
Get-FSxUserQuotaEntries	Retrieves the current user-storage quota entries for individual users and groups on the file system.
Set-FSxUserQuotas	Set the user storage quota for an individual user or group. Quota values are specified in bytes.

The online help for each command provides a reference of all command options. To access this help, run the command with **-?**, for example **Enable-FSxUserQuotas -?**.

Shadow copies

Using the set of custom PowerShell commands defined by Amazon FSx, you can manage all aspects of shadow copies on your FSx for Windows File Server file systems.

Topics

Setting shadow copy storage

Shadow copies consume storage space on the same file system of which the shadow copies are taken. When you configure shadow copy storage, you define the maximum amount of storage that shadow copies can consume on the file system using the `Set-FsxShadowStorage` custom PowerShell command. You specify the maximum size that shadow copies can grow to using the `-Maxsize` or the `-Default` command options.

Using `-Maxsize`, you can define shadow copy storage as follows:

- In bytes: `Set-FsxShadowStorage -Maxsize 2500000000`
- In kilobytes, megabytes, gigabytes, or other units: `Set-FsxShadowStorage -Maxsize (2500MB)` or `Set-FsxShadowStorage -Maxsize (2.5GB)`
- As a percentage of the overall storage: `Set-FsxShadowStorage -Maxsize "20%"`
- As unbounded: `Set-FsxShadowStorage -Maxsize "UNBOUNDED"`

Use `-Default` to set shadow storage to use up to 10 percent of the file system: `Set-FsxShadowStorage -Default`. To learn more about using the default option, see Setting up shadow copies using default settings (p. 85).

To set the amount of shadow copy storage on an FSx for Windows File Server file system

1. Connect to a compute instance that has network connectivity with your file system as a user that is a member of the file system administrators group. In AWS Managed Microsoft AD, that group is **AWS Delegated FSx Administrators**. In your self-managed Microsoft AD, that group is **Domain Admins** or the custom group that you specified for administration when you created your file system. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
2. Open a Windows PowerShell window on the compute instance.
3. Use the following command to open a remote PowerShell session on your Amazon FSx file system. Replace *`FSxFileSystem-Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer. You can find the Windows Remote PowerShell endpoint in the Amazon FSx console, in the **Network & Security** section of the file system details screen, or in the response of the `DescribeFileSystem` API operation.

   ```
   PS C:\Users\delegateadmin> enter-pssession -computername FSxFileSystem-Remote-
   PowerShell-Endpoint -configurationname fsxremoteadmin
   ```

4. Verify that shadow copy storage is not already configured on the file system using the following command.

   ```
   [fs-1234567890abcef12]: PS>Get-FsxShadowStorage
   No Fsx Shadow Storage Configured
   ```

5. Set the amount of shadow storage to 10 percent of the volume using the `-Default` option.

```
[fs-1234567890abcef12]: PS>Set-FsxShadowStorage -Default
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace     MaxSpace
-------------- ---------     --------
             0         0  32530536858
```

Viewing your shadow copy storage

You can view the amount of storage currently consumed by shadow copies on your file system using the `Get-FsxShadowStorage` command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-1234567890abcef12]: PS>Get-FsxShadowStorage
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace     MaxSpace
-------------- ---------     --------
    1619869696  14417920  32530536858
```

The output shows the shadow storage configuration, as follows:

- `AllocatedSpace` – the amount of storage on the file system in bytes currently allocated to shadow copies. Initially, this value is 0.
- `UsedSpace` – the amount of storage, in bytes, currently used by shadow copies. Initially, this value is 0.
- `MaxSpace` – the maximum amount of storage, in bytes, to which shadow storage can grow. This is the value that you set for shadow copy storage (p. 118) using the `Set-FsxShadowStorage` command.

When the `UsedSpace` amount reaches the maximum shadow copy storage amount configured (`MaxSpace`), the next shadow copy that you take replaces the oldest shadow copy. If you don't want to lose your oldest shadow copies, monitor your shadow copy storage to make sure that you have sufficient storage space for new shadow copies. If you need more space, you can delete existing shadow copies (p. 122) or increase the maximum amount of shadow copy storage (p. 118).

> **Note**
> When shadow copies are automatically or manually created, they use as a storage limit the amount of shadow copy storage that you configured. Shadow copies don't use the available storage space shown by the CloudWatch `FreeStorageCapacity` metric as a storage limit.

Deleting shadow copy storage, schedule, and all shadow copies

You can delete your shadow copy configuration, including all existing shadow copies, along with the shadow copy schedule. At the same time, you can release the shadow copy storage on the file system.

To do this, enter the `Remove-FsxShadowStorage` command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowStorage

Confirm
Are you sure you want to perform this action?
```

```
Performing the operation "Remove-FsxShadowStorage" on target "Removing all Shadow Copies,
 Shadow Copy Schedule, and Shadow Storage".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y": Y
FSx Shadow Storage Configuration
Removing Shadow Copy Schedule
Removing Shadow Copies
All shadow copies removed.
Removing Shadow Storage
Shadow Storage removed successfully.
```

Creating a custom shadow copy schedule

Shadow copy schedules use scheduled task triggers in Microsoft Windows to specify when shadow copies are automatically taken. A shadow copy schedule can have multiple triggers, providing you with a lot of scheduling flexibility. Only one shadow copy schedule can exist at a time. Before you can create a shadow copy schedule, you must first set the amount of shadow copy storage (p. 118).

When you run the `Set-FsxShadowCopySchedule` command on a file system, you overwrite any existing shadow copy schedule. Optionally, you can specify the time zone for a trigger using Windows time zones and the `-TimezoneId` option. For a list of Windows time zones, see Microsoft's Default Timezone documentation or run the following at a Windows command prompt: `tzutil /l`. To learn more about Windows task triggers, see Task Triggers in Microsoft Windows Developer Center documentation.

You can also use the `-Default` option to quickly set up a default shadow copy schedule. To learn more, see Setting up shadow copies using default settings (p. 85).

To create a custom shadow copy schedule

1. Create a set of Windows scheduled task triggers to define when shadow copies are taken in the shadow copy schedule. Use the `new-scheduledTaskTrigger` command in a PowerShell on your local machine to set multiple triggers.

   This following example creates a custom shadow copy schedule that takes shadow copies every Monday–Friday, at 6:00 AM and at 6:00 PM UTC. By default, times are in UTC, unless you specify a time zone in the Windows scheduled task triggers you create.

   ```
   PS C:\Users\delegateadmin> $trigger1 = new-scheduledTaskTrigger -weekly -DaysOfWeek
    Monday,Tuesday,Wednesday,Thursday,Friday -at 06:00
   PS C:\Users\delegateadmin> $trigger2 = new-scheduledTaskTrigger -weekly -DaysOfWeek
    Monday,Tuesday,Wednesday,Thursday,Friday -at 18:00
   ```

2. Use `invoke-command` to run the `scriptblock` command. Doing so writes a script that sets the shadow copy schedule with the `new-scheduledTaskTrigger` value that you just created. Replace *`FSxFileSystem-Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer. You can find the Windows Remote PowerShell endpoint in the Amazon FSx console, in the **Network & Security** section of the file system details screen, or in the response of the `DescribeFileSystem` API operation.

   ```
   PS C:\Users\delegateadmin> invoke-command -ComputerName FSxFileSystem-Remote-
   PowerShell-Endpoint -ConfigurationName FSxRemoteAdmin -scriptblock {
   ```

3. Enter the following line at the >> prompt to set your shadow copy schedule using the `set-fsxshadowcopyschedule` command.

   ```
   >> set-fsxshadowcopyschedule -scheduledtasktriggers $Using:trigger1,$Using:trigger2 -
   Confirm:$false }
   ```

The response displays the shadow copy schedule that you configured on the file system.

```
FSx Shadow Copy Schedule


Start Time:     : 2019-07-16T06:00:00+00:00
Days of Week    : Monday,Tuesday,Wednesday,Thursday,Friday
WeeksInterval   : 1
PSComputerName : fs-0123456789abcdef1
RunspaceId      : 12345678-90ab-cdef-1234-567890abcde1

Start Time:     : 2019-07-16T18:00:00+00:00
Days of Week    : Monday,Tuesday,Wednesday,Thursday,Friday
WeeksInterval   : 1
PSComputerName : fs-0123456789abcdef1
RunspaceId      : 12345678-90ab-cdef-1234-567890abcdef
```

Viewing your shadow copy schedule

To view the existing shadow copy schedule on your file system, enter the following command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-0123456789abcdef1]PS> Get-FsxShadowCopySchedule
FSx Shadow Copy Schedule

Start Time                Days of week                              WeeksInterval
----------                ------------                              -------------
2019-07-16T07:00:00+00:00 Monday,Tuesday,Wednesday,Thursday,Friday              1
2019-07-16T12:00:00+00:00 Monday,Tuesday,Wednesday,Thursday,Friday              1
```

Deleting a shadow copy schedule

To delete the existing shadow copy schedule on your file system, enter the following command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopySchedule

Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-FsxShadowCopySchedule" on target "Removing FSx Shadow Copy
 Schedule".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y"): Y
[fs-0123456789abcdef1]PS>
```

Creating a shadow copy

To manually create a shadow copy, enter the following command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-0123456789abcdef1]PS>New-FsxShadowCopy
```

```
Shadow Copy {ABCDEF12-3456-7890-ABCD-EF1234567890} taken successfully
```

Viewing existing shadow copies

To view the set of existing shadow copies on your file system, enter the following command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

```
[fs-0123456789abcdef1]PS>Get-FsxShadowCopies
FSx Shadow Copies: 2 total

Shadow Copy ID                          Creation Time
--------------                          -----------------
{ABCDEF12-3456-7890-ABCD-EF1234567890} 6/17/2019 7:11:09 AM
{FEDCBA21-6543-0987-0987-EF3214567892} 6/19/2019 11:24:19 AM
```

Deleting shadow copies

You can delete one or more existing shadow copies on your file system using the `Remove-FsxShadowCopies` command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Specify which shadow copies to delete by using one of the following required options:

- `-Oldest` deletes the oldest shadow copy
- `-All` deletes all existing shadow copies
- `-ShadowCopyId` deletes a specific shadow copy by ID.

You can use only one option with the command. An error occurs if you don't specify which shadow copy to delete, if you specify multiple shadow copy IDs, or if you specify an invalid shadow copy ID.

To delete the oldest shadow copy on your file system, enter the following command in a remote PowerShell session on your file system.

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopies -Oldest
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-FSxShadowCopies" on target "Removing oldest shadow copy".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y": Y
Shadow Copy {ABCDEF12-3456-7890-ABCD-EF1234567890} deleted
```

To delete a specific shadow copy on your file system, enter the following command in a remote PowerShell session on your file system.

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopies -ShadowCopyId "{ABCDEF12-3456-7890-ABCD-
EF1234567890}"
Are you sure you want to perform this action?
Performing the operation "Remove-FSxShadowCopies" on target "Removing shadow copy
 {ABCDEF12-3456-7890-ABCD-EF1234567890}".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y":>Y
Shadow Copy \\AMZNFSXABCDE123\root\cimv2:Wind32_ShadowCopy.ID{ABCDEF12-3456-7890-ABCD-
EF1234567890}".ID deleted.
```

Managing encryption in transit

You can use a set of custom PowerShell commands to control the encryption of your data in transit between your FSx for Windows File Server file system and clients. You can limit file system access to only clients supporting SMB encryption so that data-in-transit is always encrypted. When enforcement is turned on for encryption of data-in-transit, users accessing the file system from clients that do not support SMB 3.0 encryption will not be able to access file shares for which encryption is turned on.

You can also control encryption of data-in-transit on a file share-level instead of file server-level. You can use file share-level encryption controls to have a mix of encrypted and unencrypted file shares on the same file system if you want to enforce encryption in-transit for some file shares that have sensitive data, and allow all users to access some other file shares. Server-wide encryption has precedence over share level encryption. If global encryption is enabled, you cannot selectively disable encryption for certain shares.

You can manage user in-transit encryption on your file system using the Amazon FSx CLI for remote management on PowerShell. To learn how to use this CLI, see Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Following are commands that you can use to manage user in-transit encryption on your file system.

Encryption in Transit Command	Description
Get-FSxSmbServerConfiguration	Retrieves the Server Message Block (SMB) server configuration.
Set-FSxSmbServerConfiguration	This command has two options for configurig in-transit encryption: • `-EncryptData $True\|$False` – Sets in-transit data encryption on or off. • `-RejectUnencryptedAccess $True\|$False` – Allows or disallows access to clients that do not support encryption.

The online help for each command provides a reference of all command options. To access this help, run the command with **-?**, for example **Get-FSxSmbServerConfiguration -?**.

Managing storage capacity

As you need additional storage, you can increase the storage capacity that is configured on your FSx for Windows File Server file system. You can do so using the Amazon FSx console, the Amazon FSx API, or the AWS Command Line Interface (AWS CLI).

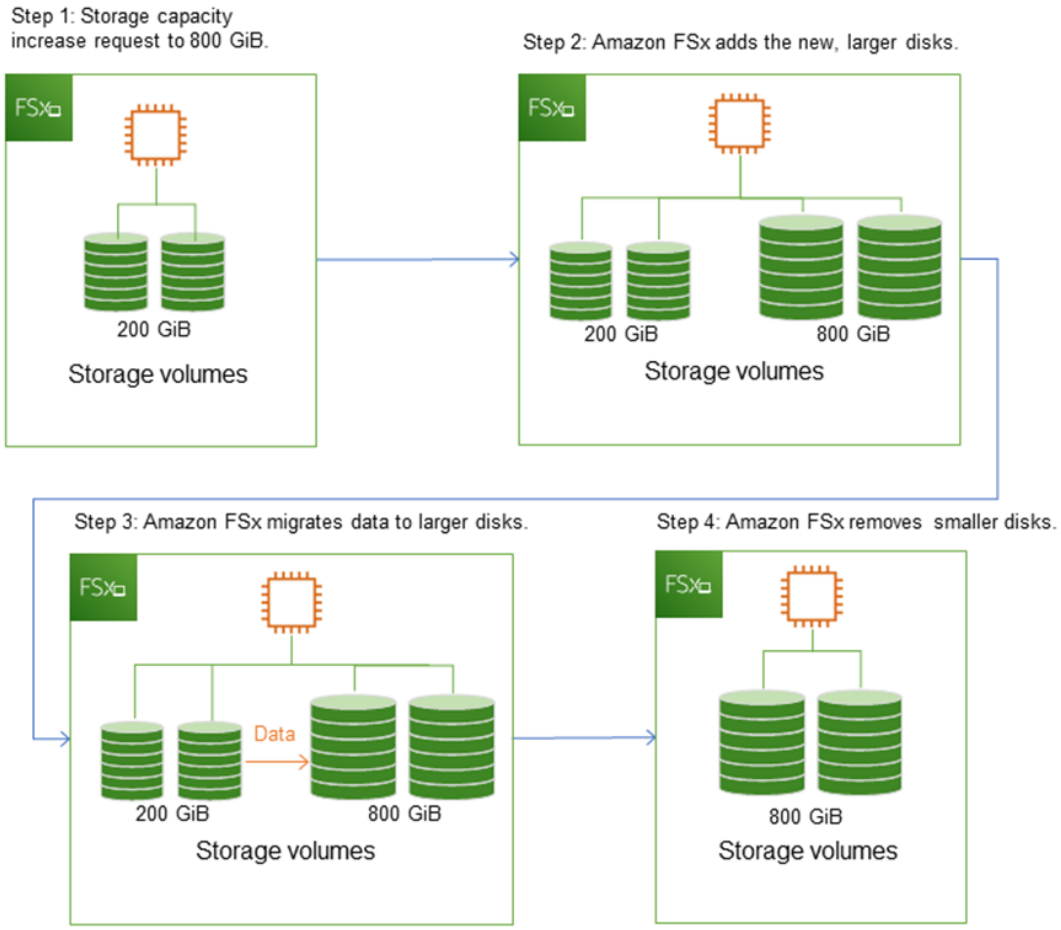
> **Note**
> You can only *increase* the amount of storage capacity for a file system; you cannot decrease storage capacity.

> **Note**
> Increasing storage capacity is not available for file systems created before June 23, 2019 or file systems restored from a backup belonging to a file system that was created before June 23, 2019.

When you increase the storage capacity of your Amazon FSx file system, behind the scenes, Amazon FSx adds a new, larger set of disks to your file system. The new capacity is available for use within minutes. When the new storage capacity becomes available, you are billed only for the new storage capacity.

Amazon FSx runs a storage optimization process in the background to transparently migrate data from the old disks to the new, larger disks. For most file systems, storage optimization takes a few hours up to a few days, with minimal noticeable impact on your workload performance.

The following illustration shows the four main steps of the process that Amazon FSx uses when increasing a file system's storage capacity.



You can track the storage optimization progress at any time using the Amazon FSx console, CLI, and API. For more information, see Monitoring storage capacity increases (p. 127).

Topics

Important points to know when increasing storage capacity

Here are a few important items to consider when increasing storage capacity:

- **Increase only** – You can only *increase* the amount of storage capacity for a file system; you cannot decrease storage capacity.
- **Minimum increase** – Each storage capacity increase must be a minimum of 10 percent of the file system's current storage capacity, up to the maximum allowed value of 65,536 GiB.
- **Minimum throughput capacity** – To increase storage capacity, a file system must have a minimum throughput capacity of 16 MB/s. This is because the storage optimization step is a throughput-intensive process.
- **Time between increases** – You can't make further storage capacity increases on a file system until 6 hours after the last increase was requested, or until the storage optimization process has completed, whichever time is longer. Storage optimization can take from a few hours up to a few days to complete. To minimize the time it takes for storage optimization to complete, we recommend increasing your file system's throughput capacity before increasing storage capacity (the throughput capacity can be scaled back down after storage scaling completes), and increasing storage capacity when there is minimal traffic on the file system.

When to increase storage capacity

Increase your file system's storage capacity when it's running low on free storage capacity. Use the `FreeStorageCapacity` CloudWatch metric to monitor the amount of free storage available on the file system. You can create an Amazon CloudWatch alarm on this metric and get notified when it drops below a specific threshold. For more information, see Monitoring with Amazon CloudWatch (p. 145).

You can automatically increase your file system's storage capacity when the amount free storage capacity falls below a defined threshold you specify. Use the AWS-developed custom AWS CloudFormation template to deploy all the components required to implement the automated solution. For more information, see Increasing storage capacity dynamically (p. 129).

Storage capacity increases and file system performance

Most workloads experience minimal performance impact while Amazon FSx runs the storage optimization process in the background after the new storage capacity is available. Write-heavy applications with large active datasets could temporarily experience up to a one-half reduction in the write performance. For these cases, you can first increase your file system's throughput capacity *before* increasing storage capacity. This enables you to continue providing the same level of throughput to meet your application's performance needs. For more information, see Managing throughput capacity (p. 133).

How to increase storage capacity

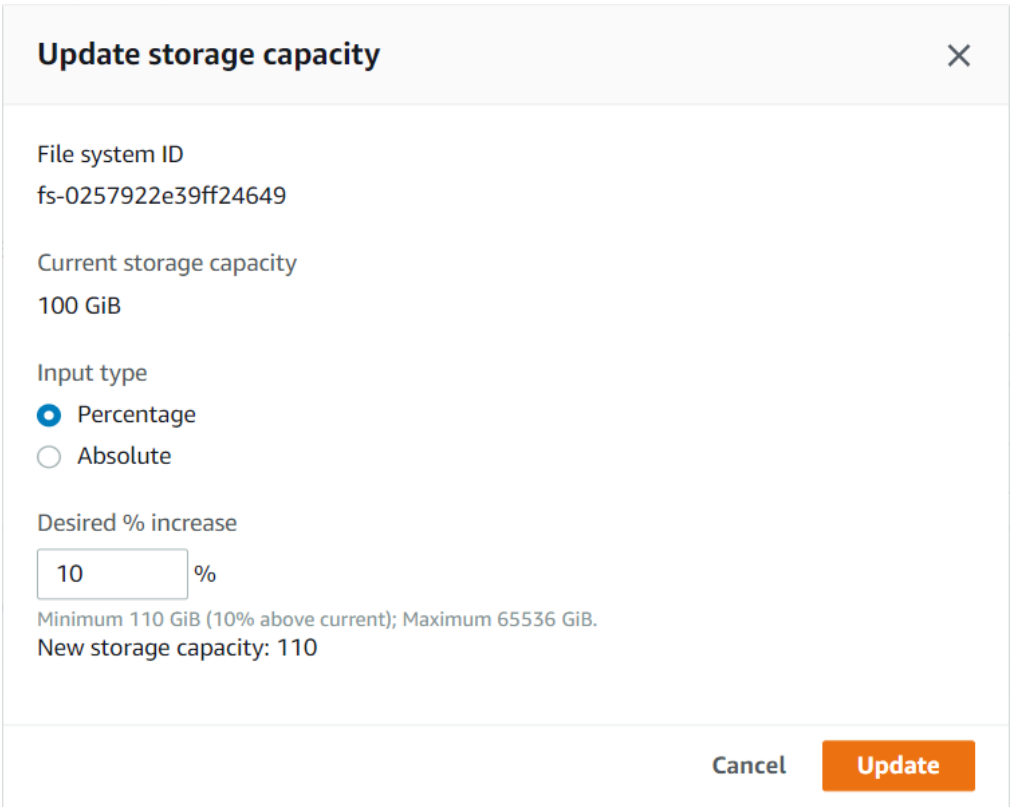
You can increase a file system's storage capacity using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

To increase storage capacity for a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.

2. Navigate to **File systems** and choose the Windows file system that you want to increase storage capacity for.
3. For **Actions**, choose **Update storage**. Or, in the **Summary** panel, choose **Update** next to the file system's **Storage capacity**.

   The **Update storage capacity** window appears.



4. For **Input type**, choose **Percentage** to enter the new storage capacity as a percentage change from the current value, or choose **Absolute** to enter the new value in GiB.
5. Enter the **Desired storage capacity**.

   Note
   The desired capacity value must be at least 10 percent larger than the current value, up to the maximum value of 65,536 GiB.
6. Choose **Update** to initiate the storage capacity update.
7. You can monitor the update progress on the **File systems** detail page, in the **Updates** tab.

To increase storage capacity for a file system (CLI)

To increase the storage capacity for an FSx for Windows File Server file system, use the AWS CLI command update-file-system. Set the following parameters:

- `--file-system-id` to the ID of the file system you are updating.
- `--storage-capacity` to a value that is at least 10 percent greater than the current value.

You can monitor the progress of the update by using the AWS CLI command describe-file-systems. Look for the `administrative-actions` in the output.

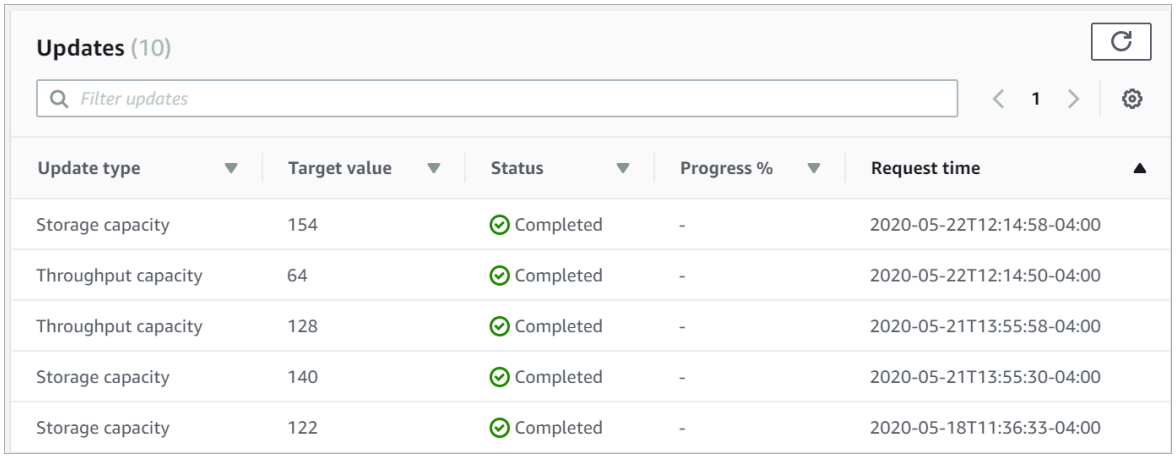
For more information, see AdministrativeAction.

Monitoring storage capacity increases

You can monitor the progress of a storage capacity increase using the Amazon FSx console, the API, or the AWS CLI.

Monitoring increases in the console

In the **Updates** tab in the **File system details** window, you can view the 10 most recent updates for each update type.



For storage capacity updates, you can view the following information.

Update type

Supported types are **Storage capacity**, **Storage optimization**, and **Throughput capacity**.

Target value

The desired value to update the file system's storage capacity to.

Status

The current status of the update. For storage capacity updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Updated optimizing** – Amazon FSx has increased the file system's storage capacity. The storage optimization process is now moving the file system data to the new larger disks.
- **Completed** – The storage capacity increase completed successfully.
- **Failed** – The storage capacity increase failed. Choose the question mark (**?**) to see details on why the storage update failed.

Progress %

Displays the progress of the storage optimization process as percent complete.

Request time

The time that Amazon FSx received the update action request.

Monitoring increases with the AWS CLI and API

You can view and monitor file system storage capacity increase requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array

lists the 10 most recent update actions for each administrative action type. When you increase a file system's storage capacity, two `AdministrativeActions` are generated: a `FILE_SYSTEM_UPDATE` and a `STORAGE_OPTIMIZATION` action.

The following example shows an excerpt of the response of a **describe-file-systems** CLI command. The file system has a storage capacity of 300 GB, and there is a pending administrative action to increase the storage capacity to 1000 GB.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 300,
            "AdministrativeActions": [
                {
                     "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                     "RequestTime": 1581694764.757,
                     "Status": "PENDING",
                     "TargetFileSystemValues": {
                         "StorageCapacity": 1000
                     }
                },
                {
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
                    "Status": "PENDING",
                }
            ]
```

Amazon FSx processes the `FILE_SYSTEM_UPDATE` action first, adding the new larger storage disks to the file system. When the new storage is available to the file system, the `FILE_SYSTEM_UPDATE` status changes to `UPDATED_OPTIMIZING`. The storage capacity shows the new larger value, and Amazon FSx begins processing the `STORAGE_OPTIMIZATION` administrative action. This is shown in the following excerpt of the response of a **describe-file-systems** CLI command.

The `ProgressPercent` property displays the progress of the storage optimization process. After the storage optimization process completes successfully, the status of the `FILE_SYSTEM_UPDATE` action changes to `COMPLETED`, and the `STORAGE_OPTIMIZATION` action no longer appears.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 1000,
            "AdministrativeActions": [
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "RequestTime": 1581694764.757,
                    "Status": "UPDATED_OPTIMIZING",
                    "TargetFileSystemValues": {
                        "StorageCapacity": 1000
                }
                },
                {
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
```

```
                    "Status": "IN_PROGRESS",
                    "ProgressPercent": 50,
                }
            ]
```

If the storage capacity increase fails, the status of the `FILE_SYSTEM_UPDATE` action changes to `FAILED`. The `FailureDetails` property provides information about the failure, shown in the following example.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 300,
            "AdministrativeActions": [
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "FailureDetails": {
                        "Message": "string"
                    },
                    "RequestTime": 1581694764.757,
                    "Status": "FAILED",
                    "TargetFileSystemValues":
                        "StorageCapacity": 1000
                }
            ]
```

For information about troubleshooting failed actions, see Storage or throughput capacity updates fail (p. 218).

Increasing the storage capacity of an FSx for Windows File Server file system dynamically

You can use the following solution to dynamically increase the storage capacity of an FSx for Windows File Server file system when the amount of free storage capacity falls below a defined threshold that you specify. This AWS CloudFormation template automatically deploys all the components that are required to define the free storage capacity threshold, the Amazon CloudWatch alarm based on this threshold, and the AWS Lambda function that increases the file system's storage capacity.

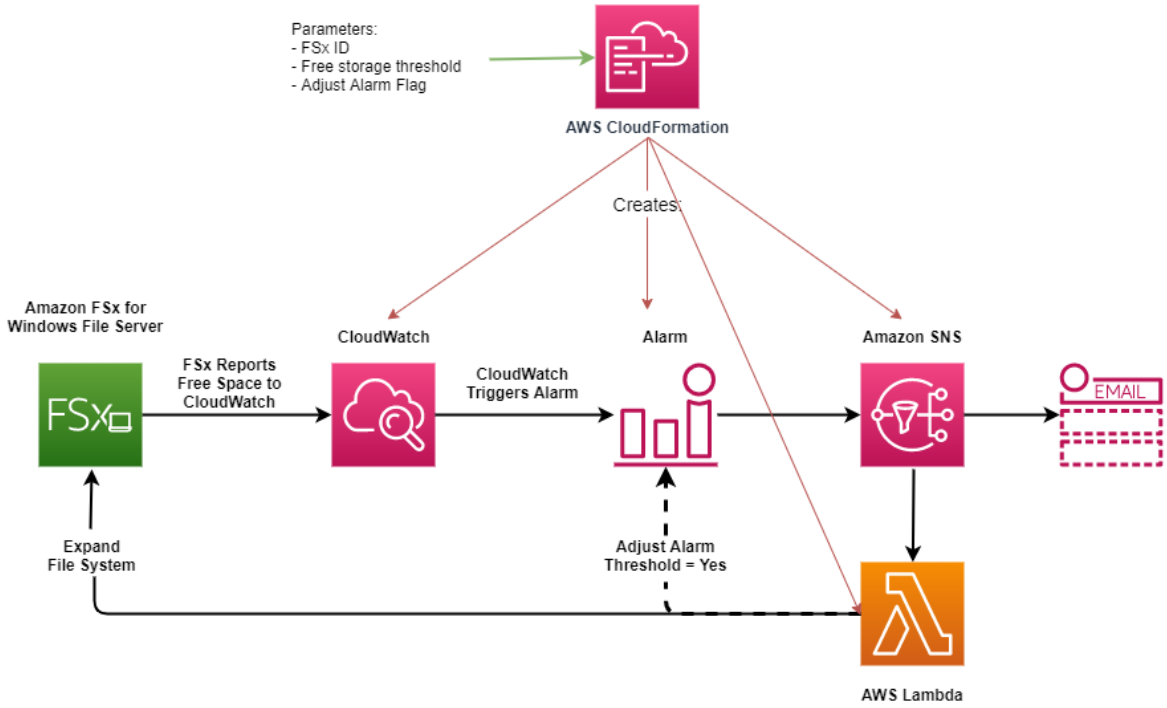
The solution automatically deploys all the components needed, and takes in the following parameters:

- The file system ID
- The free storage capacity threshold (numerical value)
- Unit of measurement (percentage [default] or GiB)
- The percentage by which to increase the storage capacity (%)
- The email address for the SNS subscription
- Adjust alarm threshold (Yes/No)

Topics

Architecture overview

Deploying this solution builds the following resources in the AWS Cloud.



The diagram illustrates the following steps:

1. The AWS CloudFormation template deploys a CloudWatch alarm, an AWS Lambda function, an Amazon Simple Notification Service (Amazon SNS) queue, and all required AWS Identity and Access Management (IAM) roles. The IAM role gives the Lambda function permission to invoke the Amazon FSx API operations.
2. CloudWatch triggers an alarm when the file system's free storage capacity goes below the specified threshold, and sends a message to the Amazon SNS queue.
3. The solution then triggers the Lambda function that is subscribed to this Amazon SNS topic.
4. The Lambda function calculates the new file system storage capacity based on the specified percent increase value and sets the new file system storage capacity.
5. The Lambda function can optionally adjust the free storage capacity threshold so that it is equal to a specified percentage of the file system's new storage capacity.
6. The original CloudWatch alarm state and results of the Lambda function operations are sent to the Amazon SNS queue.

To receive notifications about the actions that are performed as a response to the CloudWatch alarm, you must confirm the Amazon SNS topic subscription by following the link provided in the **Subscription Confirmation** email.

AWS CloudFormation template

This solution uses AWS CloudFormation to automate deploying the components that are used to automatically increase the storage capacity of an FSx for Windows File Server file system. To use this solution, download the IncreaseFSxSize AWS CloudFormation template.

The template uses the **Parameters** described as follows. Review the template parameters and their default values, and modify them for the needs of your file system.

FileSystemId

No default value. The ID of the file system for which you want to automatically increase the storage capacity.

LowFreeDataStorageCapacityThreshold

No default value. Specifies the initial free storage capacity threshold at which to trigger an alarm and automatically increase the file system's storage capacity, specified in GiB or as a percentage (%) of the file system's current storage capacity. When expressed as a percentage, the CloudFormation template re-calculates to GiB to match the CloudWatch alarm settings.

LowFreeDataStorageCapacityThresholdUnit

Default is **%**. Specifies the units for the `LowFreeDataStorageCapacityThreshold`, either in GiB or as a percentage of the current storage capacity.

AlarmModificationNotification

Default is **Yes**. If set to Yes, the initial `LowFreeDataStorageCapacityThreshold`, is increased proportionally to the value of `PercentIncrease` for subsequent alarm thresholds.

For example, when `PercentIncrease` is set to 20, and AlarmModificationNotification is set to Yes, the available free space threshold (`LowFreeDataStorageCapacityThreshold`) specified in GiB is increased by 20% for subsequent storage capacity increase events.

EmailAddress

No default value. Specifies the email address to use for the SNS subscription and will receive the storage capacity threshold alerts.

PercentIncrease

No default value. Specifies the amount by which to increase the storage capacity, expressed as a percentage of the current storage capacity.

Automated deployment with AWS CloudFormation

The following procedure configures and deploys an AWS CloudFormation stack to automatically increase the storage capacity of an FSx for Windows File Server file system. It takes about 5 minutes to deploy.

Note
Implementing this solution incurs billing for the associated AWS services. For more information, see the pricing details pages for those services.

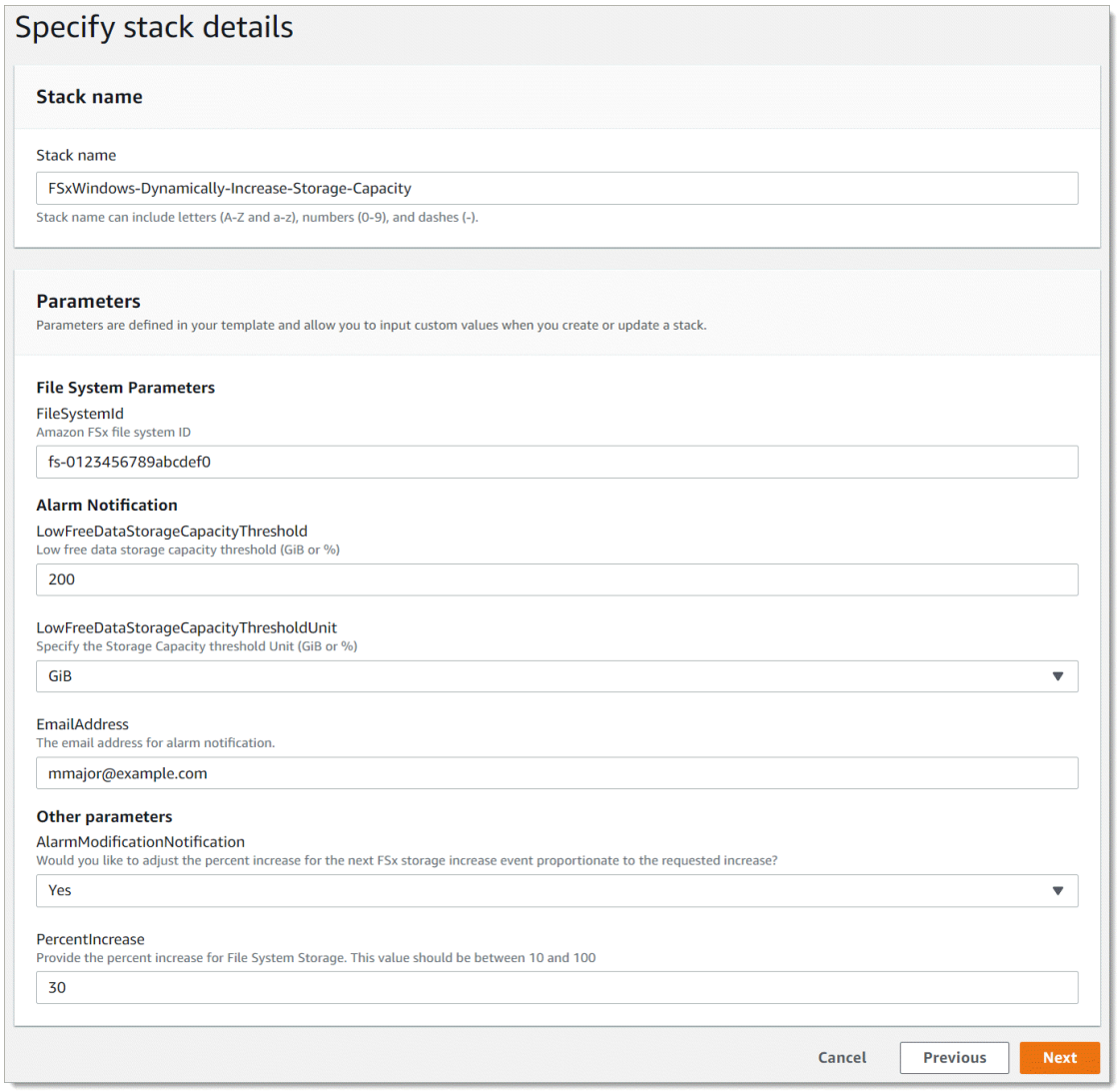
Before you start, you must have the ID of the Amazon FSx file system running in an Amazon Virtual Private Cloud (Amazon VPC) in your AWS account. For more information about creating Amazon FSx resources, see Getting started with Amazon FSx (p. 7).

To launch the automatic storage capacity increase solution stack

1. Download the IncreaseFSxSize AWS CloudFormation template. For more information about creating a CloudFormation stack, see Creating a stack on the AWS CloudFormation console in the *AWS CloudFormation User Guide*.

   Note
   Amazon FSx is currently only available in specific AWS Regions. You must launch this solution in an AWS Region where Amazon FSx is available. For more information, see Amazon FSx endpoints and quotas in the *AWS General Reference*.

2. In **Specify stack details**, enter the values for your automatic storage capacity increase solution.



3. Enter a **Stack name**.
4. For **Parameters**, review the parameters for the template and modify them for the needs of your file system. Then choose **Next**.
5. Enter any **Options** settings that you want for your custom solution, and then choose **Next**.
6. For **Review**, review and confirm the solution settings. You must select the check box acknowledging that the template creates IAM resources.
7. Choose **Create** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in about 5 minutes.

Updating the stack

After the stack is created, you can update it by using the same template and providing new values for the parameters. For more information, see Updating stacks directly in the *AWS CloudFormation User Guide*.

Managing throughput capacity

Every FSx for Windows File Server file system has a throughput capacity that is configured when you create the file system. You can modify your file system's throughput capacity at any time, as needed. Throughput capacity is one factor that determines the speed at which the file server hosting the file system can serve file data. Higher levels of throughput capacity also come with higher levels of I/O operations per second (IOPS) and more memory for caching of data on the file server. For more information, see FSx for Windows File Server performance (p. 153).

When you modify your file system's throughput capacity, behind the scenes, Amazon FSx switches out the file system's file server. For Multi-AZ file systems, it results in an automatic failover and failback while Amazon FSx switches out the preferred and secondary file servers. For single-AZ systems, your file system will be unavailable for a few minutes during throughput capacity scaling. You are billed for the new amount of throughput capacity once it is available to your file system.

> **Note**
> During a maintenance operation on the back end, system modifications (such as a modification to your throughput capacity) may be delayed. Maintenance can cause these changes to queue up until they are next to be processed.

Topics

When to modify throughput capacity

Amazon FSx integrates with Amazon CloudWatch, enabling you to monitor your file system's ongoing throughput usage levels. The performance (throughput and IOPS) that you can drive through your file system depends on your specific workload's characteristics, in addition to your file system's throughput capacity, storage capacity, and storage type. You can use CloudWatch metrics to determine which of these dimensions to change to improve performance. For more information, see Monitoring with Amazon CloudWatch (p. 145).

For Multi-AZ file systems, throughput capacity scaling results in an automatic failover and failback while Amazon FSx switches out the preferred and secondary file servers, and any data changes during this time need to be synchronized between file servers. Your file system will continue to be available during this time, but in order to reduce the duration of data synchronization, we recommend modifying throughput capacity during idle periods when there is minimal load on your file system.

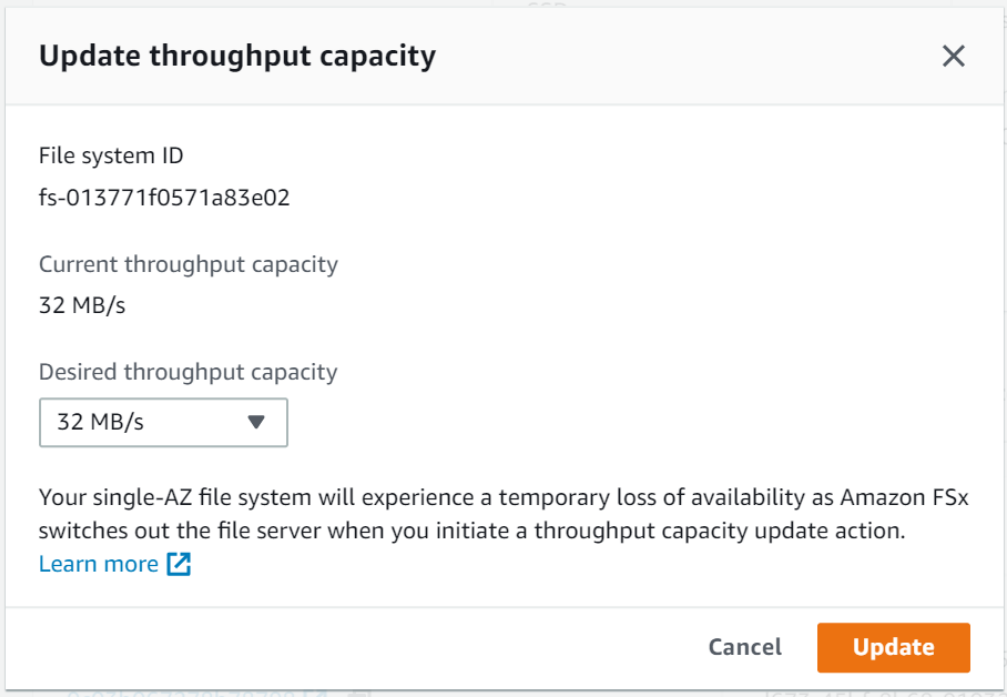
How to modify throughput capacity

You can modify a file system's throughput capacity using the Amazon FSx console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx API.

To modify a file system's throughput capacity (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to increase the throughput capacity for.
3. For **Actions**, choose **Update throughput**. Or, in the **Summary** panel, choose **Update** next to the file system's **Throughput capacity**.

   The **Update throughput capacity** window appears.
4. Choose the new value for **Throughput capacity** from the list.



5. Choose **Update** to initiate the throughput capacity update.

   Note
   Multi-AZ file systems fail over and fail back when updating throughput scaling, and are fully available. Single-AZ file systems experience a very brief period of unavailability during the update.

6. You can monitor the update progress on the **File systems** detail page, in the **Updates** tab.

   You can monitor the progress of the update by using the Amazon FSx console, the AWS CLI, and the API. For more information, see Monitoring throughput capacity changes (p. 134).

To modify a file system's throughput capacity (CLI)

To modify a file system's throughput capacity, use the AWS CLI command update-file-system. Set the following parameters:

- `--file-system-id` to the ID of the file system that you are updating.
- `ThroughputCapacity` to the desired value to update the file system to.

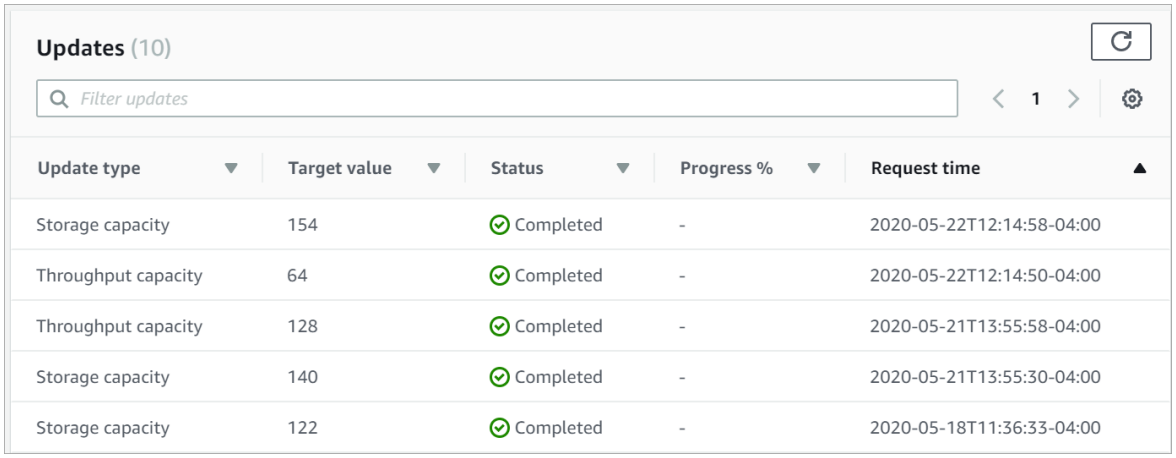
You can monitor the progress of the update by using the Amazon FSx console, the AWS CLI, and the API. For more information, see Monitoring throughput capacity changes (p. 134).

Monitoring throughput capacity changes

You can monitor the progress of a throughput capacity modification using the Amazon FSx console, the API, and the AWS CLI.

Monitoring throughput capacity changes in the console

In the **Updates** tab in the **File system details** window, you can view the 10 most recent update actions for each update action type.



For throughput capacity update actions, you can view the following information.

Update type

Supported types are **Throughput capacity**, **Storage capacity**, and **Storage optimization**.

Target value

The desired value to change the file system's throughput capacity to.

Status

The current status of the update. For throughput capacity updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Completed** – The throughput capacity update completed successfully.
- **Failed** – The throughput capacity update failed. Choose the question mark (**?**) to see details on why the throughput update failed.

Request time

The time that Amazon FSx received the update request.

Monitoring changes with the AWS CLI and API

You can view and monitor file system throughput capacity modification requests using the describe-file-systems CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you modify a file system's throughput capacity, a `FILE_SYSTEM_UPDATE` administrative action is generated.

The following example shows the response excerpt of a `describe-file-systems` CLI command. The file system has a throughput capacity of 8 MB/s, and the target throughput capacity of 256 MB/s.

```
.
.
.
    "ThroughputCapacity": 8,
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1581694764.757,
        "Status": "PENDING",
```

```
        "TargetFileSystemValues": {
          "WindowsConfiguration": {
            "ThroughputCapacity": 256
          }
        }
    }
]
```

When Amazon FSx completes processing the action successfully, the status changes to `COMPLETED`. The new throughput capacity is then available to the file system, and shows in the `ThroughputCapacity` property. This is shown in the following response excerpt of a **describe-file-systems** CLI command.

```
.
.
.
    "ThroughputCapacity": 256,
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1581694764.757,
        "Status": "COMPLETED",
        "TargetFileSystemValues": {
          "WindowsConfiguration": {
            "ThroughputCapacity": 256
          }
        }
    }
]
```

If the throughput capacity modification fails, the status changes to `FAILED`, and the `FailureDetails` property provides information about the failure. For information about troubleshooting failed actions, see Storage or throughput capacity updates fail (p. 218).

Tag your Amazon FSx resources

To help you manage your file systems and other Amazon FSx resources, you can assign your own metadata to each resource in the form of tags. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Topics

Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's Amazon FSx file systems that helps you track each instance's owner and stack level.

We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper Tagging Best Practices.

Tags don't have any semantic meaning to Amazon FSx and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

If you're using the Amazon FSx API, the AWS CLI, or an AWS SDK, you can use the `TagResource` API action to apply tags to existing resources. Additionally, some resource-creating actions enable you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, we roll back the resource creation process. This ensures that resources are either created with tags or not created at all, and that no resources are left untagged at any time. By tagging resources at the time of creation, you can eliminate the need to run custom tagging scripts after resource creation. For more information about enabling users to tag resources on creation, see Grant permission to tag resources during creation (p. 182).

Tagging your resources

You can tag Amazon FSx resources that exist in your account. If you're using the Amazon FSx console, you can apply tags to resources by using the Tags tab on the relevant resource screen. When you create resources, you can apply the Name key with a value, and you can apply tags of your choice when creating a new file system. The console may organize resources according to the Name tag, but this tag doesn't have any semantic meaning to the Amazon FSx service.

You can apply tag-based resource-level permissions in your IAM policies to the Amazon FSx API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `TagResource` and `UntagResource` Amazon FSx API actions in your IAM policies to control which tag keys and values are set on your existing resources.

For more information about tagging your resources for billing, see Using cost allocation tags in the *AWS Billing User Guide*.

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- The allowed characters for Amazon FSx tags are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.

You can't delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete a file system that you tagged with a tag key called `DeleteMe`, you must use the `DeleteFileSystem` action with the file system resource identifier, such as fs-1234567890abcdef0.

When you tag public or shared resources, the tags you assign are available only to your AWS account; no other AWS account will have access to those tags. For tag-based access control to shared resources, each AWS account must assign its own set of tags to control access to the resource.

Permissions and tag

For more information about the permissions required to tag Amazon FSx resources at creation, see Grant permission to tag resources during creation (p. 182). For more information about using tags to restrict access to Amazon FSx resources in IAM policies, see Using tags to control access to your Amazon FSx resources (p. 186).

Working with Amazon FSx maintenance windows

Amazon FSx for Windows File Server performs routine software patching for the Microsoft Windows Server software it manages. The maintenance window is your opportunity to control what day and time of the week this software patching occurs.

Patching occurs infrequently, typically once every several weeks. Patching should require only a fraction of your 30-minute maintenance window. During these few minutes of time, you should expect that your Single-AZ file system will be unavailable, and your Multi-AZ file systems will automatically fail over and fail back.

You choose the maintenance window during file system creation. If you have no time preference, then a 30-minute default window is assigned.

> **Note**
> To ensure data integrity during maintenance activity, Amazon FSx for Windows File Server completes any pending write operations to the underlying storage volumes hosting your file system before maintenance begins.

You can use the Amazon FSx Management Console, AWS CLI, AWS API, or one of the AWS SDKs to change the maintenance window for your file systems.

To change the weekly maintenance window (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose **File systems** in the left hand navigation column.
3. Choose the file system that you want to change the weekly maintenance window for. The file system details page displays.
4. Choose **Administration** to display the file system administration **Settings** panel.
5. Choose **Update** to display the **Change maintenance window** window.
6. Enter the new day and time that you want the weekly maintenance window to start.
7. Choose **Save** to save your changes. The new maintenance start time is displayed in the **Administration Settings** panel.

To change the weekly maintenance window using the CLI or API using the UpdateFileSystem operation, see Walkthrough 3: Update an existing file system (p. 162).

Best practices for administering Amazon FSx file systems

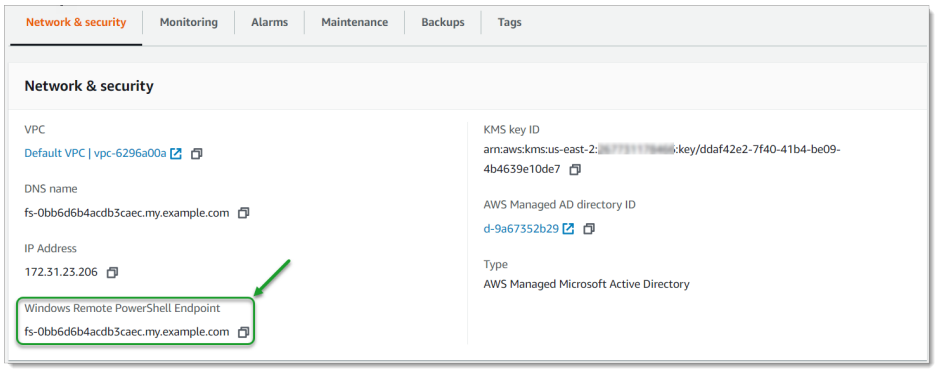
Amazon FSx provides several features that can help you implement best practices for administering your file systems, including:

- optimizing storage consumption
- enabling end-users to recover files and folders to previous versions
- enforcing encryption for all connected clients

Use the following Amazon FSx CLI for Remote Management on PowerShell commands to quickly implement these best practices on your file systems.

To run these commands, you must know the *Windows Remote PowerShell Endpoint* for your file system. To find this endpoint, follow these steps:

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose your file system. On the **Network & security** tab, locate the **Windows Remote PowerShell Endpoint**, as shown following.



For more information, see Administering file systems (p. 88) and Getting started with the Amazon FSx CLI for remote management on PowerShell (p. 88).

Topics

- One-time administrative setup tasks (p. 139)
- Ongoing administration tasks to monitor your file system (p. 141)

One-time administrative setup tasks

The following are tasks that you can quickly set up once for your file system.

Managing storage consumption

Use the following commands to manage your file system storage consumption.

- To turn on data deduplication with the default schedule, run the following command.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Enable-FsxDedup }
```

Optionally, use the following command to get data deduplication operating on your files soon after a file is created, without requiring any minimum file age.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FSxDedupConfiguration -MinimumFileAgeDays 0 }
```

For more information, see Data deduplication (p. 113).

- Use the following command to turn on user storage quotas in "Track" mode, which is for reporting purposes only and not for enforcement.

```
$QuotaLimit = Quota limit in bytes
$QuotaWarningLimit = Quota warning threshold in bytes
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Enable-FSxUserQuotas -Track -DefaultLimit
 $Using:QuotaLimit -DefaultWarningLimit $Using:QuotaWarningLimit }
```

For more information, see Storage quotas (p. 116).

Turning on shadow copies to enable end-users to recover files and folders to previous versions

Turn on shadow copies with the default schedule (weekdays 7 AM and 12 noon), as follows.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxShadowStorage -Default }

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxShadowCopySchedule -Default -Confirm:$False}
```

For more information, see Shadow copies (p. 117).

Enforcing encryption in transit

The following command enforces encryption for clients connecting to your file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxSmbServerConfiguration -EncryptData $True -
RejectUnencryptedAccess $True -Confirm:$False}
```

You can close all open sessions and force clients currently connected to reconnect using encryption.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Close-FSxSmbSession -Confirm:$False}
```

For more information, see Managing encryption in transit (p. 123) and User sessions and open files (p. 110).

Ongoing administration tasks to monitor your file system

The following ongoing tasks help you monitor your file system's disk usage, user quotas, and open files.

Monitoring deduplication status

Monitor deduplication status, including the savings rate achieved on your file system, as follows.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -
ConfigurationName FsxRemoteAdmin -ScriptBlock { Get-FSxDedupStatus } | select
 OptimizedFilesCount,OptimizedFilesSize,SavedSpace,OptimizedFilesSavingsRate
```

Monitoring user-level storage consumption

Get a report of the current user storage quota entries, including how much space they're consuming and whether they're violating the limit and the warning threshold.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Get-FSxUserQuotaEntries }
```

Monitoring and closing open files

Manage open files by looking for files left open, and closing them. Use the following command to check for open files.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Get-FSxSmbOpenFile}
```

Use the following command to close open files.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Close-FSxSmbOpenFile -Confirm:$false}
```

Grouping multiple file systems with DFS Namespaces

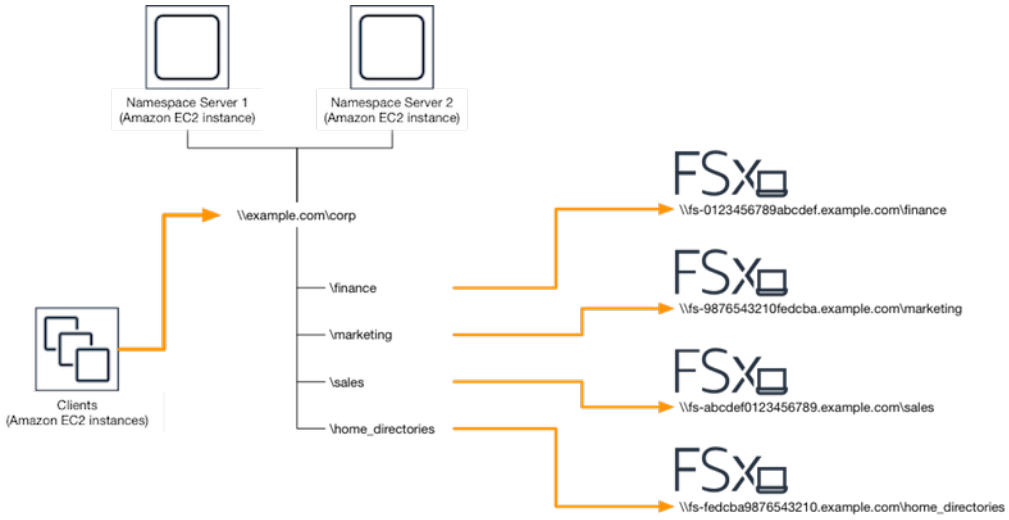
Amazon FSx for Windows File Server supports the use of Microsoft's Distributed File System (DFS) Namespaces. You can use DFS Namespaces to group file shares on multiple file systems into one common folder structure (a namespace) that you use to access the entire file dataset. DFS Namespaces can help you to organize and unify access to your file shares across multiple file systems. DFS Namespaces can also help to scale file data storage beyond what each file system supports (64 TB) for large file datasets—up to hundreds of petabytes.

Setting up DFS Namespaces for grouping multiple file systems

You can use DFS Namespaces to group multiple file systems under a single namespace. In the example that follows, the domain-based namespace (example.com\corp) is created on two namespace servers, consolidating file shares stored on multiple Amazon FSx file systems (finance, marketing, sales, home_directories). This allows your users to access file shares using a common namespace. Given this, they don't need to specify file-system DNS names for each of the file systems hosting the file shares.

Note
Amazon FSx cannot be added to the root of the DFS share path.

These steps guide you through creating a single namespace (example.com\corp) on two namespace servers. You also set up four file shares under the namespace, each transparently redirecting users to shares hosted on separate Amazon FSx file systems.



To group multiple file systems into a common DFS namespace

1. If you don't already have DFS Namespace servers running, you can launch a pair of highly available DFS Namespace servers using the setup-DFSN-servers.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the *AWS CloudFormation User Guide*.

2. Connect to one of the DFS Namespace servers launched in the previous step as a user in the **AWS Delegated Administrators** group. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Access the DFS Management Console by opening. Open the **Start** menu and run **dfsmgmt.msc**. This opens the DFS Management GUI tool.
4. Choose **Action** then **New Namespace**, type in the computer name of the first DFS Namespace server you launched for **Server** and choose **Next**.
5. For **Name**, type in the namespace you're creating (for example, **corp**).
6. Choose **Edit Settings** and set the appropriate permissions based on your requirements. Choose **Next**.
7. Leave the default **Domain-based namespace** option selected, leave the **Enable Windows Server 2008 mode** option selected, and choose **Next**.

   Note
   Windows Server 2008 mode is the latest available option for Namespaces.
8. Review the namespace settings and choose **Create**.
9. With the newly created namespace selected under **Namespaces** in the navigation bar, choose **Action** then **Add Namespace Server**.
10. Type in the computer name of the second DFS Namespace server you launched for **Namespace server**.
11. Choose **Edit Settings**, set the appropriate permissions based on your requirements, and choose **OK**.
12. Open the context (right-click) menu for the namespace you just created, choose **New Folder**, type in the name of the folder (for example, `finance` for **Name**, and choose **OK**.
13. Type in the DNS name of the file share that you want the DFS Namespace folder to point to in UNC format (for example, `\\fs-0123456789abcdef0.example.com\finance`) for **Path to folder target** and choose **OK**.
14. If the share doesn't exist:

    a. Choose **Yes** to create it.
    b. From the **Create Share** dialog, choose **Browse**.
    c. Choose an existing folder, or create a new folder under **D$**, and choose **OK**.
    d. Set the appropriate share permissions, and choose **OK**.
15. From the **New Folder** dialog, choose **OK**. The new folder will be created under the namespace.
16. Repeat the last four steps for other folders you want to share under the same namespace.

Monitoring FSx for Windows File Server

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon FSx and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. However, before you start monitoring Amazon FSx, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

The next step is to establish a baseline for normal Amazon FSx performance in your environment, by measuring performance at various times and under different load conditions. As you monitor Amazon FSx, you should consider storing historical monitoring data. This stored data gives you a baseline to compare against with current performance data, identify normal performance patterns and performance anomalies, and devise methods to address issues.

For example, with Amazon FSx, you can monitor network throughput, I/O for read, write, and metadata operations, and the amount of available storage capacity for your file system. When performance falls outside your established baseline, you might need to change the size of your file system to optimize the file system for your workload.

To establish a baseline, you should, at a minimum, monitor the following items:

- Your file system's network throughput.
- The number of bytes for each file system operation, including data read, data write, and metadata operations.

Monitoring tools

AWS provides various tools that you can use to monitor Amazon FSx. You can configure some of these tools to do the monitoring for you, whereas some of the tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

Automated monitoring tools

You can use the following automated monitoring tools to watch Amazon FSx and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring with Amazon CloudWatch (p. 145).

- **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources. For more information, see What Is Amazon CloudWatch Logs? in the *Amazon CloudWatch Logs User Guide*.
- **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see Working with CloudTrail Log Files in the *AWS CloudTrail User Guide*.

Manual monitoring tools

Another important part of monitoring Amazon FSx involves manually monitoring those items that the Amazon CloudWatch alarms don't cover. The Amazon FSx, CloudWatch, and other AWS console dashboards provide an at-a-glance view of the state of your AWS environment. We recommend that you also check the log files on the file system.

- From the Amazon FSx console, you can find the following items for your file systems:
  - Free storage capacity
  - Total throughput (bytes/sec)
  - Total IOPS (operations/sec)
- The CloudWatch home page shows:
  - Current alarms and status
  - Graphs of alarms and resources
  - Service health status

  In addition, you can use CloudWatch to do the following:
  - Create customized dashboards to monitor the services you use.
  - Graph metric data to troubleshoot issues and discover trends.
  - Search and browse all your AWS resource metrics.
  - Create and edit alarms to be notified of problems.

Monitoring with Amazon CloudWatch

You can monitor file systems using Amazon CloudWatch, which collects and processes raw data from FSx for Windows File Server into readable, near real-time metrics. These statistics are retained for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. By default, Amazon FSx for Windows File Server metric data is automatically sent to CloudWatch at 1-minute periods. For more information about CloudWatch, see What Is Amazon CloudWatch? in the *Amazon CloudWatch User Guide*.

Amazon FSx CloudWatch metrics are reported as raw *Bytes*. Bytes are not rounded to either a decimal or binary multiple of the unit.

Amazon FSx for Windows File Server publishes the following metrics into the `AWS/FSx` namespace in CloudWatch. For each metric, FSx for Windows File Server emits a data point per file system per minute.

Metric	Description
`DataReadBytes`	The number of bytes for file system read operations. The `Sum` statistic is the total number of bytes associated with read operations during the period. To calculate the average throughput (Bytes

Metric	Description
	per second) for a period, divide the `Sum` statistic by the number of seconds in the period. Units: Bytes Valid statistics: `Sum`
`DataWriteBytes`	The number of bytes for file system write operations. The `Sum` statistic is the total number of bytes associated with write operations during the period. To calculate the average throughput (Bytes per second) for a period, divide the `Sum` statistic by the number of seconds in the period. Units: Bytes Valid statistics: `Sum`
`DataReadOperations`	The number of read operations. The `Sum` statistic is the count of read operations during the time period. To calculate the average number of read operations (operations per second) for a period, divide the `Sum` statistic by the number of seconds in the period. Units: Count Valid statistics: `Sum`
`DataWriteOperations`	The number of write operations. The `Sum` statistic is the count of write operations during the time period. To calculate the average number of write operations (operations per second) for a period, divide the `Sum` statistic by the number of seconds in the period. Units: Count Valid statistics: `Sum`
`MetadataOperations`	The number of metadata operations. The `Sum` statistic is the count of metadata operations during the time period. To calculate the average number of metadata operations (operations per second) for a period, divide the `Sum` statistic by the number of seconds in the period. Units: Count Valid statistics: `Sum`
`FreeStorageCapacity`	The amount of available storage capacity. Units: Bytes Valid statistics: `Average`, `Minimum`

FSx for Windows File Server dimensions

FSx for Windows File Server metrics use the `FSx` namespace and provide metrics for a single dimension, `FileSystemId`. You can find a file system's ID using the describe-file-systems AWS CLI command or the DescribeFileSystems API command. A file system ID takes the form of *`fs-0123456789abcdef0`*.

How to use FSx for Windows File Server metrics

The metrics reported by Amazon FSx provide information that you can analyze in different ways. The list following shows some common uses for the metrics. These are suggestions to get you started, not a comprehensive list.

How do I determine...	Relevant metrics
My file system's IOPS?	Total IOPS = SUM(DataReadOperations + DataWriteOperations + MetadataOperations)/Period (in seconds)
My file system's throughput?	SUM(DataReadBytes + DataWriteBytes)/Period (in seconds)

Note
We recommend that you maintain an average throughput capacity utilization under 50% to ensure that you have enough spare throughput capacity for unexpected spikes in your workload, as well as for any background Windows storage operations (such as storage synchronization, deduplication, or shadow copies).

Accessing CloudWatch metrics

You can see Amazon FSx metrics for CloudWatch in the following ways.

- The Amazon FSx console.
- The CloudWatch console.
- The CloudWatch CLI (command line interface).
- The CloudWatch API.

The following procedures show you how to access the metrics using these various tools.

To view metrics using the Amazon FSx console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation pane, choose **File systems**, then choose the file system whose metrics you want to view.
3. Choose **Actions** and choose **View details**.
4. On the **Summary** page, choose **Monitoring** to see the metrics for your file system.

To view metrics using the CloudWatch console

1. Open the CloudWatch console.
2. In the navigation pane, choose **Metrics**.
3. Select the **FSx** namespace.

4. (Optional) To view a metric, enter its name in the search field.
5. (Optional) To filter by dimension, select **FileSystemId**.

To access metrics from the AWS CLI

- Use the `list-metrics` command with the `--namespace "AWS/FSx"` namespace. For more information, see the AWS CLI Command Reference.

Using the CloudWatch API

To access metrics from the CloudWatch API

- Call `GetMetricStatistics`. For more information, see Amazon CloudWatch API Reference.

Creating CloudWatch alarms to monitor Amazon FSx

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy.

Alarms invoke actions for sustained state changes only. CloudWatch alarms don't invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. You can create an alarm from the Amazon FSx console or the CloudWatch console.

The following procedures describe how to create alarms for Amazon FSx using the console, AWS CLI, and API.

To set alarms using the Amazon FSx console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation pane, choose **File systems**, and then choose the file system you want to create the alarm for.
3. Choose the **Actions** menu, and choose **View details**.
4. On the **Summary** page, choose **Alarms**.
5. Choose **Create CloudWatch alarm**. You are redirected to the CloudWatch console.
6. Choose **Select metrics**, and choose **Next**.
7. In the **Metrics** section, choose **FSX**.
8. Choose **File System Metrics**, choose the metric you want to set the alarm for, and then choose **Select metric**.
9. In the **Conditions** section, choose the conditions you want for the alarm, and choose **Next**.

   Note
   Metrics may not be published during file system maintenance for Single-AZ file systems. To prevent unnecessary and misleading alarm condition changes and to configure your alarms so that they are resilient to missing data points, see Configuring how CloudWatch alarms treat missing data in the *Amazon CloudWatch User Guide*.
10. If you want CloudWatch to send you an email or SNS notification when the alarm state triggers the action, choose an alarm state for **Whenever this alarm state is**.

    For **select an SNS topic**, choose an existing SNS topic. If you select **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms. Choose **Next**.

Note
If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.

11. Fill in the **Name**, **Description**, and **Whenever** values for the metric, and choose **Next**.
12. On the **Preview and create** page, review the alarm you're about to create, and then choose **Create Alarm**.

To set alarms using the CloudWatch console

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Choose **Create Alarm** to start the **Create Alarm Wizard**.
3. Choose **FSx Metrics**, and scroll through the Amazon FSx metrics to locate the metric you want to place an alarm on. To display just the Amazon FSx metrics in this dialog box, search on the file system ID of your file system. Select the metric to create an alarm on, and choose **Next**.
4. Fill in the **Name**, **Description**, and **Whenever** values for the metric.
5. If you want CloudWatch to send you an email when the alarm state is reached, for **Whenever this alarm**, choose **State is ALARM**. For **Send notification to**, choose an existing SNS topic. If you select **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms.

   Note
   If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.
6. At this point, the **Alarm Preview** area gives you a chance to preview the alarm you're about to create. Choose **Create Alarm**.

To set an alarm using the AWS CLI

- Call `put-metric-alarm`. For more information, see *AWS CLI Command Reference*.

To set an alarm using the CloudWatch API

- Call `PutMetricAlarm`. For more information, see *Amazon CloudWatch API Reference*.

Logging FSx for Windows File Server API Calls with AWS CloudTrail

Amazon FSx is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon FSx. CloudTrail captures all API calls for Amazon FSx as events. Captured calls include calls from the Amazon FSx console and from code calls to Amazon FSx API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon FSx. 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 Amazon FSx. You can also determine 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.

Amazon FSx Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When API activity occurs in Amazon FSx, 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 Amazon FSx, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following topics in the *AWS CloudTrail User Guide:*

- 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 Amazon FSx API calls are logged by CloudTrail. For example, calls to the `CreateFileSystem` and `TagResource` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element in the *AWS CloudTrail User Guide.*

Understanding Amazon FSx Log File Entries

A *trail* is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `TagResource` operation when a tag for a file system is created from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": “111122223333”,
        "arn": "arn:aws:sts::111122223333:root”,
```

```
        "accountId": “111122223333”,
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE”,
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T22:36:07Z"
            }
        }
    },
    "eventTime": "2018-11-14T22:36:07Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": “192.0.2.0”,
    "userAgent": “console.amazonaws.com”,
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789”
    },
    "responseElements": null,
    "requestID": “aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51”,
    "eventID": “bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p”,
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": “111122223333”
}
```

The following example shows a CloudTrail log entry that demonstrates the `UntagResource` action when a tag for a file system is deleted from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T23:40:54Z"
            }
        }
    },
    "eventTime": "2018-11-14T23:40:54Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789"
    },
    "responseElements": null,
    "requestID": "aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51",
    "eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": "111122223333"
}
```

FSx for Windows File Server performance

FSx for Windows File Server offers file systems to meet a variety of performance needs. Following is an overview of Amazon FSx file system performance, with a discussion of the available performance and throughput options and useful performance tips.

Topics

Overview

File system performance is measured by its latency, throughput, and I/O operations per second (IOPS).

Latency

FSx for Windows File Server file servers employ a fast, in-memory cache to achieve consistent sub-millisecond latencies for actively accessed data. For data that is not in the in-memory cache, that is, for file operations that need to be served by performing I/O on the underlying storage volumes, Amazon FSx provides sub-millisecond file operation latencies with solid state drive (SSD) storage, and single-digit millisecond latencies with hard disk drive (HDD) storage.

Throughput and IOPS

Amazon FSx file systems provide up to multiple GB/s of throughput and hundreds of thousands of IOPS. The specific amount of throughput and IOPS that your workload can drive on your file system depends on the throughput capacity and storage capacity configuration of your file system, along with the nature of your workload, including the size of the active working set.

Single-client performance

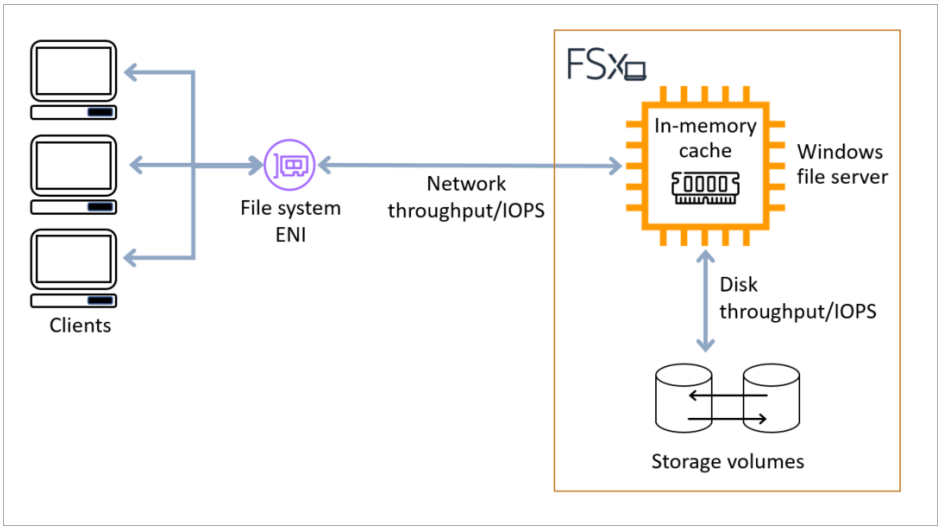
With Amazon FSx, you can get up to the full throughput and IOPS levels for your file system from a single client accessing it. Amazon FSx supports *SMB Multichannel*. This feature enables it to provide up to multiple GB/s throughput and hundreds of thousands of IOPS for a single client accessing your file system. SMB Multichannel uses multiple network connections between the client and server simultaneously to aggregate network bandwidth for maximal utilization.

Performance details

To understand the Amazon FSx performance model in detail, you can examine the architectural components of an Amazon FSx file system. Your client compute instances, whether they exist in AWS or on-premises, access your file system through an elastic network interface (ENI). This network interface resides in the Amazon VPC that you associate with your file system. Behind the file system ENI is the Windows file server that is serving data over the network to the clients accessing the file system. Amazon

FSx provides a fast in-memory cache on the file server to enhance performance for the most frequently accessed data. Behind the file server are the storage volumes, or disks, hosting your file system data.

These components are illustrated in the following diagram.



Corresponding with these architectural components–network interface, in-memory cache, and storage volumes–are the three primary performance characteristics of an FSx for Windows File Server file system that determine the overall throughput and IOPS performance.

- Network I/O performance: throughput/IOPS of requests between the clients and the file server (in aggregate)
- In-memory cache size on the file server: size of active working set that can be accommodated for caching
- Disk I/O performance: throughput/IOPS of requests between the file server and the storage volumes

There are two factors that determine these performance characteristics for your file system: the amount of storage capacity and throughput capacity that you configure for it. The first two performance characteristics – network I/O performance and in-memory cache size – are solely determined by throughput capacity, while the third one – disk I/O performance – is determined by a combination of throughput capacity and storage capacity.

File-based workloads are typically spiky, characterized by short, intense periods of high I/O with plenty of idle time between bursts. To support spiky workloads, in addition to the baseline speeds that a file system can sustain 24/7, Amazon FSx provides the capability to burst to higher speeds for periods of time for both network I/O and disk I/O operations. Amazon FSx uses a network I/O credit mechanism to allocate throughput and IOPS based on average utilization — file systems accrue credits when their throughput and IOPS usage is below their baseline limits, and can use these credits when they perform I/O operations.

Impact of storage capacity on performance

The type and amount of storage capacity impacts the performance of your file system. You need to configure the type and amount of storage capacity necessary for your file system to deliver the desired performance levels for your workload.

The maximum disk throughput and IOPS levels your file system can achieve is the lower of:

- the disk performance level provided by your file server, based on the throughput capacity you select for your file system

- the disk performance level provided by the type and amount of storage capacity you select for your file system

Your file system's storage provides the following levels of disk throughput and IOPS:

Storage type	Disk throughput (Megabytes/second per TiB of storage)	Disk IOPS (IOPs per TiB of storage)
SSD	750	3,000
HDD	12 baseline; 80 burst (up to a max. of 1 GB/s per file system)	12 baseline; 80 burst

You can increase a file system's storage capacity at any time. For more information, see Managing storage capacity (p. 123).

Impact of throughput capacity on performance

Every Amazon FSx file system has a throughput capacity that you configure when the file system is created. The throughput capacity determines the level of network I/O performance, that is, the speed at which the file server hosting your file system can serve file data over the network to clients accessing it. Higher levels of throughput capacity come with more memory for caching data on the file server, and higher levels of disk I/O performance supported by the file server.

When you create a file system using the Amazon Web Services Management Console, Amazon FSx automatically picks the recommended throughput capacity level for your file system based on the amount of storage capacity you select. While the recommended throughput capacity should be sufficient for most workloads, you have the option to override the recommendation and select a specific throughput capacity level to meet your application's needs. You can increase or decrease the amount of throughput capacity at any time after you create it. For more information, see Managing throughput capacity (p. 133).

The following table shows the full set of specifications for throughput capacity, along with baseline and burst levels, and amount of memory on the file server (memory that is available for caching, and for performing background activities such as data deduplication and shadow copies).

Note
The following table shows the set of choices you have for selecting the throughput capacity for your file system while using the Amazon FSx console. While you can select lower levels (8 MBps or 16 MBps) for throughput capacity when you use the Amazon FSx API or CLI, keep in mind that the 8 MBps and 16 MBps levels are meant for test and development workloads, not for production workloads. 8 MBps and 16 MBps throughput capacities do not support file access auditing.

FSx throughput capacity (MBps)	Network throughput capacity (MBps)		Network IOPS	Memory (GB)	Disk throughput (MBps)		Disk IOPS	
	Baseline	Burst (for a few minutes a day)			Baseline	Burst (for 30 mins a day)	Baseline	Burst (for 30 mins a day)

FSx throughput capacity (MBps)	Network throughput capacity (MBps)		Network IOPS	Memory (GB)	Disk throughput (MBps)		Disk IOPS	
32	32	600	Thousands	4	32	260	2K	12K
64	64	600	Tens of thousands	8	64	350	4K	16K
128	150	1,250		8	128	600	6K	20K
256	300	1,250	Hundreds of thousands	16	256	600	10K	20K
512	600	1,250		32	512	–	20K	–
1,024	1,500	–		72	1,024	–	40K	–
2,048	3,125	–		144	2,048	–	80K	–

Example: storage capacity and throughput capacity

The following example illustrates how storage capacity and throughput capacity impact file system performance.

A file system that is configured with 2 TiB of HDD storage capacity and 32 MBps of throughput capacity has the following throughput levels:

- Network throughput – 32 MBps baseline and 600 MBps burst (see throughput capacity table)
- Disk throughput – 24 MBps baseline and 160 MBps burst, which is the lower of the disk throughput levels of 32 MBps baseline and 260 MBps burst supported by the file server (based on throughput capacity), and the disk throughput levels of 24 MBps baseline (12 MBps per TB * 2 TB) and 160 MBps burst (80 MBps per TB * 2 TB) supported by the storage capacity.

Your workload accessing the file system will therefore be able to drive up to 32 MBps baseline and 600 MBps burst throughput for file operations performed on actively accessed data cached in the file server in-memory cache, and up to 24 MBps baseline and 160 MBps burst throughput for file operations that need to go all the way to the disk, for example, due to cache misses.

Measuring performance using CloudWatch metrics

You can use Amazon CloudWatch to measure and monitor your file system's throughput and IOPS. For more information, see How to use FSx for Windows File Server metrics (p. 147).

Amazon FSx Walkthroughs

Following, you can find a number of task-oriented walkthroughs that guide you through various processes.

Topics

Walkthrough 1: Prerequisites for getting started

Before you can complete the getting started exercise, you must already have a Microsoft Windows–based Amazon EC2 instance joined to your AWS Directory Service directory. You must also be signed into the instance over Windows Remote Desktop Protocol as the Admin user for your directory. The following walkthrough shows you how to perform these necessary prerequisite actions.

Topics

Step 1: Set up Active Directory

With Amazon FSx, you can operate fully managed file storage for Windows-based workloads. Likewise, AWS Directory Service provides fully managed directories to use in your workload deployment. If you have an existing corporate AD domain running in AWS in a virtual private cloud (VPC) using EC2 instances, you can enable user-based authentication and access control. You do this by establishing a trust relationship between your AWS Managed Microsoft AD and your corporate domain. For Windows authentication in Amazon FSx, you only need a one-way directional forest trust, where the AWS managed forest trusts the corporate domain forest.

Your corporate domain takes the role of the trusted domain, and the AWS Directory Service managed domain takes the role of the trusting domain. Validated authentication requests travel between the domains in only one direction—allowing accounts in your corporate domain to authenticate against resources shared in the managed domain. In this case, Amazon FSx interacts only with the managed domain. The managed domain then passes on the authentication requests to your corporate domain.

> **Note**
> You can also use an external trust type with Amazon FSx for trusted domains.

Your Active Directory security group must enable inbound access from the Amazon FSx file system's security group.

To create an AWS Directory Services for Microsoft AD

- If you don't already have one, use the AWS Directory Service to create your AWS Managed Microsoft AD directory. For more information, see Create Your AWS Managed Microsoft AD directory in the *AWS Directory Service Administration Guide*.

  Important
  Remember the password you assign to your Admin user; you need it later in this getting started exercise. If you forget the password, you need to repeat steps in this exercise with the new AWS Directory Service directory and Admin user.

- If you have an existing AD, create a trust relationship between your AWS Managed Microsoft AD and your existing AD. For more information, see When to Create a Trust Relationship in the *AWS Directory Service Administration Guide.*

Step 2: Launch a Windows instance in the Amazon EC2 console

You can launch a Windows instance using the AWS Management Console as described in the following procedure. This is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see Launching an Instance.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the console dashboard, choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Select the AMI for Windows Server 2016 Base or Windows Server 2012 R2 Base. Notice that these AMIs are marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the `t2.micro` type, which is selected by default. Notice that this instance type is eligible for the free tier.
5. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, a security group appears that the wizard created and selected for you. You can use this security group, or you can choose the security group that you created when getting set up using the following steps:

   a. Choose **Edit security groups**.

   b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.

   c. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

   Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of

your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Warning
Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
10. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. (If the **Public DNS (IPv4)** column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) in the top right corner of the page and then select **Public DNS (IPv4)**.)
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

Important
Make a note of the ID of the security group that was created when you launched this instance. You'll need it when you create your Amazon FSx file system.

Now that your instance is launched, you can connect to your instance.

Step 3: Connect to your instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

The name of the administrator account depends on the language of the operating system. For example, for English it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see Localized Names for Administrator Account in Windows in the Microsoft TechNet Wiki.

If you joined your instance to a domain, you can connect to your instance using domain credentials you defined in AWS Directory Service. On the Remote Desktop login screen, don't use the local computer name and the generated password. Instead, use the fully qualified user name for the administrator and the password for this account. An example is **`corp.example.com\Admin`**.

The license for the Windows Server operating system (OS) allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your Windows instance. If you need more than two simultaneous remote connections, you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs. For more information, see Configure the Number of Simultaneous Remote Connections Allowed for a Connection.

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect to Your Instance** dialog box, choose **Get Password** (it takes a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect to Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.

5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect to Your Instance** dialog box.

   - If you opened the .rdp file, you see the **Remote Desktop Connection** dialog box.
   - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.
7. You may get a warning that the publisher of the remote connection is unknown. You can continue to connect to your instance.
8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and type the user name and password manually.

   Note
   Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.
9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.

   a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
   b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.
   c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
   d. In the system log output, look for an entry labeled `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
   e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
   f. [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

Now that you're connected to your instance, you can join the instance to your AWS Directory Service directory.

Step 4: Join your instance to your AWS Directory Service directory

The following procedure shows you how to manually join an existing Amazon EC2 Windows instance to your AWS Directory Service directory.

To join a Windows instance to your AWS Directory Service directory

1. Connect to the instance using any Remote Desktop Protocol client.
2. Open the TCP/IPv4 properties dialog box on the instance.

   a. Open **Network Connections**.

Tip
You can open **Network Connections** directly by running the following from a command prompt on the instance.

```
%SystemRoot%\system32\control.exe ncpa.cpl
```

b. Open the context (right-click) menu for any enabled network connection and then choose **Properties**.

c. In the connection properties dialog box, open (double-click) **Internet Protocol Version 4**.

3. (Optional) Select **Use the following DNS server addresses**, change the **Preferred DNS server** and **Alternate DNS server** addresses to the IP addresses of the AWS Directory Service–provided DNS servers, and choose **OK**.
4. Open the **System Properties** dialog box for the instance, choose the **Computer Name** tab, and choose **Change**.

   Tip
   You can open the **System Properties** dialog box directly by running the following from a command prompt on the instance.

   ```
   %SystemRoot%\system32\control.exe sysdm.cpl
   ```

5. In the **Member of** box, choose **Domain**, enter the fully qualified name of your AWS Directory Service directory, and choose **OK**.
6. When prompted for the name and password for the domain administrator, enter the user name and password of the Admin account.

   Note
   You can enter either the fully qualified name of your domain or the NetBios name, followed by a backslash (\), and then the user name, in this case, **Admin**. For example, **corp.example.com\Admin** or **corp\Admin**.

7. After you receive the message welcoming you to the domain, restart the instance to have the changes take effect.
8. Reconnect to your instance over RDP, and sign into the instance using the user name and password for your AWS Directory Service directory's Admin user.

Now that your instance has been joined to the domain, you're ready to create your Amazon FSx file system. You can then go on to finish the other tasks in the getting started exercise. For more information, see Getting started with Amazon FSx (p. 7).

Walkthrough 2: Create a file system from a backup

With Amazon FSx, you can create a file system from a backup. When you do so, you can change any of the following elements to better suit the use case you have for your newly created file system:

- Storage type
- Throughput capacity
- VPC
- Availability Zone
- Subnet
- VPC security groups
- Active Directory Configuration

- AWS KMS encryption key
- Daily automatic backup start time
- Weekly maintenance window

The following procedure guides you through the process of creating a new file system from a backup. Before you can create this file system, you must have an existing backup. For more information, see Working with backups (p. 78)

To create a file system from an existing backup

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation list at right, choose **Backups**.
3. From the table on the dashboard, choose the backup that you want to use for creating a new file system.

   Note
   You can only restore your backup to a file system of the same storage capacity as the original. You can increase your restored file system's storage capacity after it becomes available. For more information, see Managing storage capacity (p. 123).
4. Choose **Restore backup**. This will begin the create file system wizard.
5. Choose the settings that you'd like to change for this new file system. The storage type is set to **SSD** by default, but you can change it to **HDD** under the following conditions:

   - The file system deployment type is **Multi-AZ** or **Single-AZ 2**.
   - The storage capacity is at least 2,000 GiB.
6. Choose **Review summary** to review your settings before creating the file system.
7. Choose **Create file system**.

You've now successfully created your new file system from an existing backup.

Walkthrough 3: Update an existing file system

There are three elements that you can update with the procedures in this walkthrough. All other elements of your file system that you can update, you can do so from the console. These procedures assume you have the AWS CLI installed and configured on your local computer. For more information, see Install and Configure in the *AWS Command Line Interface User Guide*.

- **AutomaticBackupRetentionDays** – the number of days that you want to retain automatic backups for your file system.
- **DailyAutomaticBackupStartTime** – the time of the day in Coordinated Universal Time (UTC) that you want the daily automatic backup window to start. The window is 30 minutes starting from this specified time. This window can't overlap with the weekly maintenance backup window.
- **WeeklyMaintenanceStartTime** – the time of the week that you want the maintenance window to start. Day 1 is Monday, 2 is Tuesday, and so on. The window is 30 minutes starting from this specified time. This window can't overlap with the daily automatic backup window.

The following procedures outlines how to update your file system with the AWS CLI.

To update how long automatic backups are retained for your file system

1. Open a command prompt or terminal on your computer.

2. Run the following command, replacing the file system ID with the ID for your file system, and the number of days that you want to retain your automatic backups for.

```
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-
configuration AutomaticBackupRetentionDays=30
```

To update the daily backup window of your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the time with when you want to begin the window.

```
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-
configuration DailyAutomaticBackupStartTime=01:00
```

To update the weekly maintenance window of your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the date and time with when you want to begin the window.

```
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-
configuration WeeklyMaintenanceStartTime=1:01:30
```

Walkthrough 4: Using Amazon FSx with Amazon AppStream 2.0

By supporting the Server Message Block (SMB) protocol, Amazon FSx for Windows File Server supports accessing your file system from Amazon EC2, VMware Cloud on AWS, Amazon WorkSpaces, and Amazon AppStream 2.0 instances. AppStream 2.0 is a fully managed application streaming service. You centrally manage your desktop applications on AppStream 2.0 and securely deliver them to a browser on any computer. For more information on AppStream 2.0, see the *Amazon AppStream 2.0 Administration Guide*.

Use this walkthrough as a guide through how to use Amazon FSx with AppStream 2.0 for two use cases: providing personal persistent storage to each user and providing a shared folder across users to access common files.

Providing personal persistent storage to each user

You can use Amazon FSx to provide every user in your organization a unique storage drive within AppStream 2.0 streaming sessions. A user will have permissions to access only their folder. The drive is automatically mounted at the start of a streaming session and files added or updated to the drive are automatically persisted between streaming sessions.

There are three procedures you'll need to perform to complete this task.

To create home folders for domain users using Amazon FSx

1. Create an Amazon FSx file system. For more information, see Getting started with Amazon FSx (p. 7).

2. After the file system is available, create a folder for every domain AppStream 2.0 user within your Amazon FSx file system. The example following uses the domain user name of the user as the name of the corresponding folder. Doing this means that you can build the UNC name of the file share to map easily using the Windows environment variable `%username%`.
3. Share each of these folders out as a shared folder. For more information, see File shares (p. 95).

To launch a domain-joined AppStream 2.0 image builder

1. Sign into the AppStream 2.0 console: https://console.aws.amazon.com/appstream2
2. Choose **Directory Configs** from the navigation menu, and create a Directory Config object. For more information, see Using Active Directory with AppStream 2.0 in the *Amazon AppStream 2.0 Administration Guide*.
3. Choose **Images**, **Image Builder**, and launch a new image builder.
4. Choose the directory config object created earlier in the image builder launch wizard to join the image builder to your Active Directory domain.
5. Launch the image builder in the same VPC as that of your Amazon FSx file system. Make sure to associate the image builder with the same AWS Managed Microsoft AD directory to which your Amazon FSx file system is joined. The VPC security groups that you associate with the image builder must allow access to your Amazon FSx file system.
6. Once the image builder is available, connect to the image builder and login using your domain administrator account.
7. Install your applications.

To link Amazon FSx file shares with AppStream 2.0

1. In the image builder, create a batch script with the following command and store it in a known file location (for example: C:\Scripts\map-fs.bat). The following example uses S: as the drive letter to map the shared folder on your Amazon FSx file system. You use the DNS name of your Amazon FSx file system or a DNS alias associated with the file system in this script, which you can get from the file system details view in the Amazon FSx console.

   If you're using the file system's DNS name:

   ```
   @echo off
   net use S: /delete
   net use S: \\file-system-DNS-name\users\%username%
   ```

   If you're using a DNS alias associated with the file system:

   ```
   @echo off
   net use S: /delete
   net use S: \\fqdn-DNS-alias\users\%username%
   ```

2. Open a PowerShell prompt and run `gpedit.msc`.
3. From **User Configuration** choose **Windows Settings** and then **Logon**.
4. Navigate to the batch script that you created in the first step of this procedure, and choose it.
5. From **Computer Configuration**, choose **Windows Administrative Templates**, **System**, and then **Group Policy**.
6. Choose the policy **Configure Logon Script delay**. Enable the policy and reduce the time delay to `0`. This setting helps to ensure that the user logon script is executed immediately when the user starts a streaming session.
7. Create your image and assign it to an AppStream 2.0 fleet. Ensure that you also join the AppStream 2.0 fleet to the same Active Directory domain that you used for image builder. Launch the fleet

in the same VPC that is used by your Amazon FSx file system. The VPC security groups that you associate with the fleet must provide access to your Amazon FSx file system.

8. Launch a streaming session using SAML SSO. To connect to an fleet that is joined to Active Directory, configure single sign-on federation using a SAML provider. For more information, see Single Sign-on Access to AppStream 2.0 Using SAML 2.0 in the *Amazon AppStream 2.0 Administration Guide*.
9. Your Amazon FSx file share is mapped to the S: drive letter within the streaming session.

Providing a shared folder across users

You can use Amazon FSx to provide a shared folder to users in your organization. A shared folder can be used to maintain common files (for example, demo files, code examples, instruction manuals, etc.) needed by all users.

There are three procedures you'll need to perform to complete this task.

To create a shared folder using Amazon FSx

1. Create an Amazon FSx file system. For more information, see Getting started with Amazon FSx (p. 7).
2. Every Amazon FSx file system includes a shared folder by default that you can access using the address *file-system-DNS-name*\share, or *fqdn-DNS-alias*\share if you are using DNS aliases. You can use the default share or create a different shared folder. For more information, see File shares (p. 95).

To launch an AppStream 2.0 image builder

1. From the AppStream 2.0 console, launch a new image builder or connect to an existing image builder. Launch the image builder in the same VPC that is used by your Amazon FSx file system. The VPC security groups that you associate with the image builder must allow access to your Amazon FSx file system.
2. Once the image builder is available, connect to the image builder as the Administrator user.
3. Install or update your applications as Administrator.

To link the shared folder with AppStream 2.0

1. Create a batch script, as described in the previous procedure, to automatically mount the shared folder whenever a user launches a streaming session. To complete the script, you need the file system's DNS name or a DNS alias that is associated with the file system (which you can obtain from the file system details view in the Amazon FSx Console), and credentials for accessing the shared folder.

   If you're using the file system's DNS name:

   ```
   @echo off
   net use S: /delete
   net use S: \\file-system-DNS-name\share /user:username password
   ```

   If you're using a DNS alias associated with the file system:

   ```
   @echo off
   net use S: /delete
   net use S: \\fqdn-DNS-alias\share /user:username password
   ```

2. Create a Group Policy to execute this batch script at every user logon. You can follow the same instructions as described in the previous section.
3. Create your image and assign it to your fleet.
4. Launch a streaming session. You should now see the shared folder automatically mapped to the drive letter.

Walkthrough 5: Using DNS aliases to access your file system

FSx for Windows File Server provides a default Domain Name System (DNS) name for every file system that you can use to access the data on your file system. You can also access your file systems using a DNS alias of your choosing. With DNS aliases, you can continue using existing DNS names to access data stored on Amazon FSx when migrating file system storage from on-premises to Amazon FSx, without needing to update any tools or applications. You can associate up to 50 DNS aliases with a file system at any one time.

To access your Amazon FSx file systems using DNS aliases, you must perform the following three steps:

1. Associate DNS aliases with your Amazon FSx file system.
2. Configure service principal names (SPNs) for your file system's computer object. (This is required to get Kerberos authentication when accessing your file system using DNS aliases.)
3. Update or create a DNS CNAME record for the file system and the DNS alias.

Topics

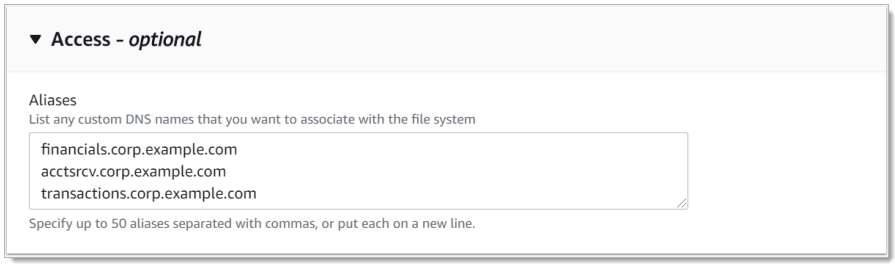
Step 1: Associate DNS aliases with your Amazon FSx file system

You can associate DNS aliases with existing FSx for Windows File Server file systems, when you create new file systems, and when you create a new file system from a backup using the Amazon FSx console, CLI, and API. If you are creating an alias with a different domain name input the full name, including parent domain, to associate an alias.

This procedure describes how to associate DNS aliases when creating a new file system using the Amazon FSx console. For information about associating DNS aliases with existing file systems, and details about using the CLI and API, see Managing DNS aliases (p. 90).

To associate DNS aliases when creating a new file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system as described in Step 1: Create your file system (p. 7) of the Getting Started section.
3. In the **Access - optional** section of the **Create file system** wizard, enter the DNS aliases that you want to associate with your file system.



Use the following guidelines when specifying DNS aliases:

- Must be formatted as a fully qualified domain name (FQDN) *hostname.domain*, for example, `accounting.example.com`.
- Can contain alphanumeric characters and hyphens (-).
- Cannot start or end with a hyphen.
- Can start with a numeric.

For DNS alias names, Amazon FSx stores alphabetic characters as lowercase letters (a-z), regardless of how you specify them: as uppercase letters, lowercase letters, or the corresponding letters in escape codes.

4. For **Maintenance preferences**, make any changes that you want.
5. In the **Tags - optional** section, add any tags that you need, and then choose **Next**.
6. Review the file system configuration shown on the **Create file system** page. Choose **Create file system** to create the file system.

   When your new file system becomes available, continue with step 2.

Step 2: Configure service principal names (SPNs) for Kerberos

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients that access your file system.

To enable Kerberos authentication for clients that access Amazon FSx using a DNS alias, you must add service principal names (SPNs) that correspond to the DNS alias on your Amazon FSx file system's Active Directory computer object. An SPN can only be associated with a single Active Directory computer object at a time. If you have existing SPNs for the DNS name configured for your original file system's Active Directory computer object, you must delete them first.

There are two required SPNs for Kerberos authentication:

```
HOST/alias
HOST/alias.domain
```

If the alias is `finance.domain.com`, the following are the two required SPNs:

```
HOST/finance
HOST/finance.domain.com
```

Note
You will need to delete any existing HOST SPNs that correspond to the DNS alias on the Active Directory computer object before you create new HOST SPNs for your Amazon FSx file system's

Active Directory (AD) computer object. Attempts to set SPNs for your Amazon FSx file system will fail if an SPN for the DNS alias exists in the AD.

The following procedures describes how to do the following:

- Find any existing DNS alias SPNs on the original file system's Active Directory computer object.
- Delete the existing SPNs found, if any.
- Create new DNS alias SPNs for your Amazon FSx file system's Active Directory computer object.

To install the required PowerShell Active Directory module

1. Log on to a Windows instance joined to the Active Directory to which your Amazon FSx file system is joined.
2. Open PowerShell as administrator.
3. Install the PowerShell Active Directory module using the following command.

```
Install-WindowsFeature RSAT-AD-PowerShell
```

To find and delete existing DNS alias SPNs on the original file system's Active Directory computer object

1. Find any existing SPNs by using the following commands. Replace *alias_fqdn* with the DNS alias that you associated with the file system in Step 1 (p. 166).

```
## Find SPNs for original file system's AD computer object
$ALIAS = "alias_fqdn"
SetSPN /Q ("HOST/" + $ALIAS)
SetSPN /Q ("HOST/" + $ALIAS.Split(".")[0])
```

2. Delete the existing HOST SPNs returned in the previous step by using the following example script.

   - Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Step 1 (p. 166).
   - Replace *file_system_DNS_name* with the original file system's DNS name.

```
## Delete SPNs for original file system's AD computer object
$Alias = "alias_fqdn"
$FileSystemDnsName = "file_system_dns_name"
$FileSystemHost = (Resolve-DnsName ${FileSystemDnsName} | Where Type -eq 'A')
[0].Name.Split(".")[0]
$FSxAdComputer = (Get-AdComputer -Identity ${FileSystemHost})

SetSPN /D ("HOST/" + ${Alias}) ${FSxAdComputer}.Name
SetSPN /D ("HOST/" + ${Alias}.Split(".")[0]) ${FSxAdComputer}.Name
```

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Step 1 (p. 166).

To set SPNs on your Amazon FSx file system's Active Directory computer object

1. Set new SPNs for your Amazon FSx file system by running the following commands.

   - Replace *file_system_DNS_name* with the DNS name that Amazon FSx assigned to the file system.

To find your file system's DNS name on the Amazon FSx console, choose **File systems**, choose your file system, and then choose the **Network & security** pane on the file system details page.

You can also get the DNS name in the response of the DescribeFileSystems API operation.

- Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Step 1 (p. 166).

```
## Set SPNs for FSx file system AD computer object
$FSxDnsName = "file_system_DNS_name"
$Alias = "alias_fqdn"
$FileSystemHost = (Resolve-DnsName $FSxDnsName | Where Type -eq 'A')[0].Name.Split(".")
[0]
$FSxAdComputer = (Get-AdComputer -Identity $FileSystemHost)

Set-AdComputer -Identity $FSxAdComputer -Add @{"msDS-AdditionalDnsHostname"="$Alias"}
SetSpn /S ("HOST/" + $Alias.Split('.')[0]) $FSxAdComputer.Name
SetSpn /S ("HOST/" + $Alias) $FSxAdComputer.Name
```

Note
Setting an SPN for your Amazon FSx file system will fail if an SPN for the DNS alias exists in the AD for the original file system's computer object. For information about finding and deleting existing SPNs, see To find and delete existing DNS alias SPNs on the original file system's Active Directory computer object (p. 168).

2. Verify that the new SPNs are configured for the DNS alias using the following example script. Ensure that the response includes two HOST SPNs, `HOST/alias` and `HOST/alias_fqdn`, as described previously in this procedure.

   Replace *file_system_DNS_name* with the DNS name that Amazon FSx assigned to your file system. To find your file system's DNS name on the Amazon FSx console, choose **File systems**, choose your file system, and then choose the **Network & security** pane on the file system details page.

   You can also get the DNS name in the response of the DescribeFileSystems API operation.

```
## Verify SPNs on FSx file system AD computer object
$FileSystemDnsName = "file_system_dns_name"
$FileSystemHost = (Resolve-DnsName ${FileSystemDnsName} | Where Type -eq 'A')
[0].Name.Split(".")[0]
$FSxAdComputer = (Get-AdComputer -Identity ${FileSystemHost})
SetSpn /L ${FSxAdComputer}.Name
```

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Step 1 (p. 166).

For information about how to enforce clients to use Kerberos authentication and encryption when connecting to your Amazon FSx file system, see Enforcing Kerberos authentication using GPOs (p. 171).

Step 3: Update or create a DNS CNAME record for the file system

After you properly configure SPNs for your file system, you can cut over to Amazon FSx by replacing each DNS record that resolved to the original file system with a DNS record that resolves to the default DNS name of the Amazon FSx file system.

The `dnsserver` and `activedirectory` Windows modules are required to run the commands presented in this section.

To install the required PowerShell cmdlets

1. Log on to a Windows instance joined to the Active Directory that your Amazon FSx file system is joined to as a user that is a member of a group that has DNS administration permissions (**AWSAWS Delegated Domain Name System Administrators** in AWS Managed Active Directory, and **Domain Admins** or another group to which you've delegated DNS administration permissions in your self-managed Active Directory).

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
2. Open PowerShell as administrator.
3. The PowerShell DNS Server module is required to perform the instructions in this procedure. Install it using the following command.

```
Install-WindowsFeature RSAT-DNS-Server
```

To update or create a custom DNS name to your Amazon FSx file system

1. Connect to your Amazon EC2 instance as a user that is a member of a group that has DNS administration permissions (**AWS Delegated Domain Name System Administrators** in AWS Managed Active Directory, and **Domain Admins** or another group to which you've delegated DNS administration permissions in your self-managed Active Directory).

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
2. At the command prompt, run the following script. This script migrates any existing DNS CNAME records to your Amazon FSx file system. If none are found, it creates a new DNS CNAME record for the DNS alias *`alias_fqdn`* that resolves to the default DNS name for your Amazon FSx file system.

   To run the script:

   - Replace *`alias_fqdn`* with the DNS alias that you associated with the file system.
   - Replace *`file_system_DNS_name`* with the DNS name Amazon FSx has assigned to the file system.

```
$Alias="alias_fqdn"
$FSxDnsName="file_system_dns_name"
$AliasHost=$Alias.Split('.')[0]
$ZoneName=((Get-WmiObject Win32_ComputerSystem).Domain)
$DnsServerComputerName = (Resolve-DnsName $ZoneName -Type NS | Where Type -eq 'A' |
 Select -ExpandProperty Name)[0]

Add-DnsServerResourceRecordCName -Name $AliasHost -ComputerName $DnsServerComputerName
 -HostNameAlias $FSxDnsName -ZoneName $ZoneName
```

3. Repeat the previous step for each DNS alias that you associated with the file system in Step 1 (p. 166).

You've now added a DNS CNAME value for your Amazon FSx file system with the DNS alias. You can now use the DNS alias to access your data.

Note
When updating a DNS CNAME record to point to an Amazon FSx file system previously pointed to another file system, clients might not be able to connect with file system for a brief period of time. When the client DNS cache refreshes, they should be able to connect using the DNS alias. For more information, see Can't access the file system using a DNS alias (p. 206).

Enforcing Kerberos authentication using GPOs

You can enforce Kerberos authentication when accessing the file system by setting the following Group Policy Objects (GPOs) in your Active Directory:

- **Restrict NTLM: Outgoing NTLM traffic to remote servers** - Use this policy setting to deny or audit outgoing NTLM traffic from a computer to any remote server running the Windows operating system.
- **Restrict NTLM: Add remote server exceptions for NTLM authentication** - Use this policy setting to create an exception list of remote servers to which client devices are allowed to use NTLM authentication if the *Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers* policy setting is configured.

1. Log on to a Windows instance joined to the Active Directory to which your Amazon FSx file system is joined as an administrator. If you are configuring a self-managed Active Directory, apply these steps directly to your Active Directory.
2. Choose **Start**, choose **Administrative Tools**, and then choose **Group Policy Management**.
3. Choose **Group Policy Objects**.
4. If your Group Policy Object does not already exist, create it.
5. Locate the existing **Network Security: Restrict NTLM: Outgoing NTLM traffic to remote servers** policy. (If there is no existing policy, create a new policy.) In the **Local security setting** tab, open the context (right-click) menu, and choose **Properties**.
6. Choose **Deny all**.
7. Choose **Apply** to save the security setting.
8. To set exceptions for NTLM connections to specific remote servers for the client, locate the **Network security: Restrict NTLM: Add remote server exceptions**.

   Open the context (right-click) menu, and choose **Properties** in the **Local security setting** tab.
9. Enter the names of any servers to add to the exception list.
10. Choose **Apply** to save the security setting.

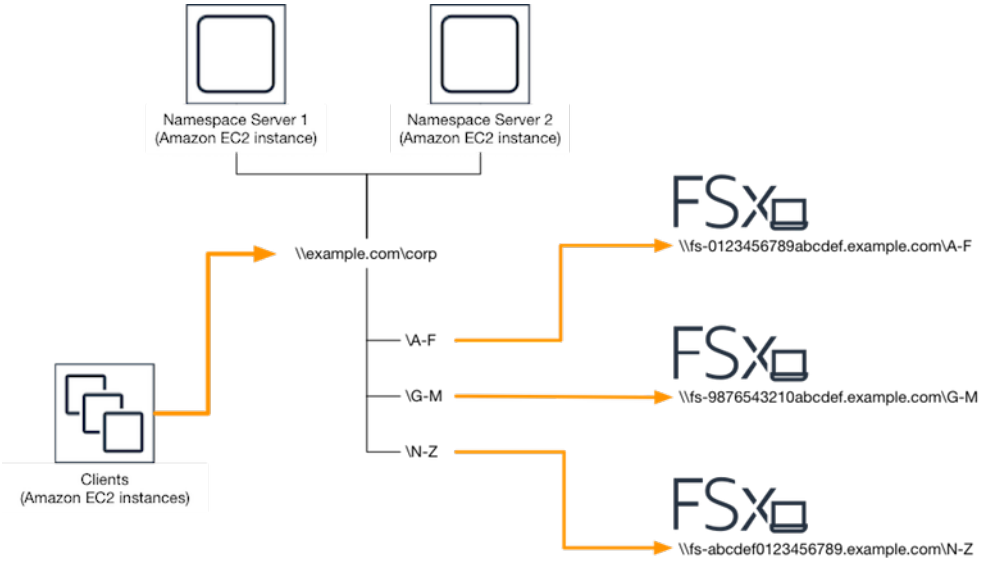
Walkthrough 6: Scaling out performance with shards

Amazon FSx for Windows File Server supports the use of the Microsoft Distributed File System (DFS). By using DFS Namespaces, you can scale out performance (both read and write) to serve I/O-intensive workloads by spreading your file data across multiple Amazon FSx file systems. At the same time, you can still present a unified view under a common namespace to your applications. This solution involves dividing your file data into smaller datasets or *shards* and storing them across different file systems. Applications accessing your data from multiple instances can achieve high levels of performance by reading and writing to these shards in parallel.

You can use this solution when your workload requires uniformly distributed read/write access to your file data (for example, if each subset of compute instances accesses a different portion of your file data).

Setting up DFS Namespaces for scale-out performance

The following procedure guides you through creating a DFS solution on Amazon FSx for scale-out performance. In this example, the data stored in the *corp* namespace is sharded alphabetically. Data files 'A-F', 'G-M' and 'N-Z' are all stored on different file shares. Based on the type of data, I/O size, and I/O access pattern, you should decide how to best shard your data across multiple file shares. Choose a sharding convention that distributes I/O evenly across all the file shares you plan on using. Keep in mind that each namespace supports up to 50,000 file shares and hundreds of petabytes of storage capacity in aggregate.



To set up DFS Namespaces for scale-out performance

1. If you don't already have DFS Namespace servers running, you can launch a pair of highly available DFS Namespace servers using the setup-DFSN-servers.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the *AWS CloudFormation User Guide*.
2. Connect to one of the DFS Namespace servers launched in the previous step as a user in the **AWS Delegated Administrators** group. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Access the DFS Management Console. Open the **Start** menu and run **dfsmgmt.msc**. This opens the DFS Management GUI tool.
4. Choose **Action** then **New Namespace**, type in the computer name of the first DFS Namespace server you launched for **Server** and choose **Next**.
5. For **Name**, type in the namespace you're creating (for example, **corp**).
6. Choose **Edit Settings** and set the appropriate permissions based on your requirements. Choose **Next**.
7. Leave the default **Domain-based namespace** option selected, leave the **Enable Windows Server 2008 mode** option selected, and choose **Next**.

   Note
   Windows Server 2008 mode is the latest available option for Namespaces.
8. Review the namespace settings and choose **Create**.
9. With the newly created namespace selected under **Namespaces** in the navigation bar, choose **Action** then **Add Namespace Server**.

10. Type in the computer name of the second DFS Namespace server you launched for **Namespace server**.
11. Choose **Edit Settings**, set the appropriate permissions based on your requirements, and choose **OK**.
12. Open the context (right-click) menu for the namespace you just created, choose **New Folder**, enter the name of the folder for the first shard (for example, `A-F` for **Name**), and choose **Add**.
13. Type in the DNS name of the file share hosting this shard in UNC format (for example, `\\fs-0123456789abcdef0.example.com\A-F`) for **Path to folder target** and choose **OK**.
14. If the share doesn't exist:
    a. Choose **Yes** to create it.
    b. From the **Create Share** dialog, choose **Browse**.
    c. Choose an existing folder, or create a new folder under **D$**, and choose **OK**.
    d. Set the appropriate share permissions, and choose **OK**.
15. With the folder target now added for the shard, choose **OK**.
16. Repeat the last four steps for other shards you want to add to the same namespace.

Walkthrough 7: Copying a backup to another AWS Region

With Amazon FSx, you can copy an existing backup within the same AWS account to another AWS Region (a cross-Region backup copy) or to the same AWS Region (an in-Region backup copy).

The following procedure guides you through the process of creating a copy of a backup within the same AWS account. Before you can create this backup copy, you must have an existing backup. For more information, see Working with backups (p. 78).

To copy an existing backup within the same AWS account (cross-Region or in-Region)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the navigation pane, choose **Backups**.
3. In the **Backups** table, choose the backup that you want to copy.
4. Choose **Copy backup**. Doing so opens the **Copy backup** wizard.
5. In the **Destination Region** list, choose a destination AWS Region to copy the backup to. The destination can be in another AWS Region or within the same AWS Region.
6. (Optional) Select **Copy Tags** to copy tags from the source backup to the destination backup. If you select **Copy Tags** and also add tags at step 8, all the tags are merged.
7. For **Encryption**, choose the AWS KMS encryption key to encrypt the copied backup.
8. For **Tags - optional**, enter a key and value to add tags for your copied backup. If you add tags here and also selected **Copy Tags** at step 6, all the tags are merged.
9. Choose **Copy backup**.

You've now successfully copied a backup within the same AWS account to another AWS Region or within the same AWS Region.

Security in Amazon FSx

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the Amazon Web Services 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 Amazon FSx for Windows File Server, 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 Amazon FSx for Windows File Server. The following topics show you how to configure Amazon FSx to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon FSx for Windows File Server resources.

Following, you can find a description of security considerations for working with Amazon FSx.

Topics

Data Encryption in Amazon FSx

Amazon FSx for Windows File Server supports two forms of encryption for file systems, encryption of data in transit and encryption at rest. Encryption of data in transit is supported on file shares that are mapped on a compute instance that supports SMB protocol 3.0 or newer. Encryption of data at rest is automatically enabled when creating an Amazon FSx file system. Amazon FSx automatically encrypts data in transit using SMB encryption as you access your file system without the need for you to modify your applications.

When to Use Encryption

If your organization is subject to corporate or regulatory policies that require encryption of data and metadata at rest, we recommend creating an encrypted file system mounting your file system using encryption of data in transit.

For more information on encryption with Amazon FSx for Windows File Server, see these related topics:

- Create Your Amazon FSx for Windows File Server File System (p. 7)
- Amazon FSx API permissions: actions, resources, and conditions reference (p. 183)

Topics

Encryption at Rest

All Amazon FSx file systems are encrypted at rest with keys managed using AWS Key Management Service (AWS KMS). Data is automatically encrypted before being written to the file system, and automatically decrypted as it is read. These processes are handled transparently by Amazon FSx, so you don't have to modify your applications.

Amazon FSx uses an industry-standard AES-256 encryption algorithm to encrypt Amazon FSx data and metadata at rest. For more information, see Cryptography Basics in the *AWS Key Management Service Developer Guide*.

> **Note**
> The AWS key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms. The infrastructure is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

How Amazon FSx uses AWS KMS

Amazon FSx integrates with AWS KMS for key management. Amazon FSx uses an AWS KMS key to encrypt your file system. You choose the KMS key used to encrypt and decrypt file systems (both data and metadata). You can enable, disable, or revoke grants on this KMS key. This KMS key can be one of the two following types:

- **AWS managed key** – This is the default KMS key, and it's free to use.
- **Customer managed key** – This is the most flexible KMS key to use, because you can configure its key policies and grants for multiple users or services. For more information on creating customer managed keys, see Creating keys in the *AWS Key Management Service Developer Guide*.

If you use a customer managed key as your KMS key for file data encryption and decryption, you can enable key rotation. When you enable key rotation, AWS KMS automatically rotates your key once per year. Additionally, with a customer managed key, you can choose when to disable, re-enable, delete, or revoke access to your KMS key at any time. For more information, see Rotating AWS KMS keys in the *AWS Key Management Service Developer Guide.*

File system encryption and decryption at rest are handled transparently. However, AWS account IDs specific to Amazon FSx appear in your AWS CloudTrail logs related to AWS KMS actions.

Amazon FSx Key Policies for AWS KMS

Key policies are the primary way to control access to KMS keys. For more information on key policies, see Using key policies in AWS KMS in the *AWS Key Management Service Developer Guide.* The following list describes all the AWS KMS-related permissions supported by Amazon FSx for encrypted at rest file systems:

- **kms:Encrypt** – (Optional) Encrypts plaintext into ciphertext. This permission is included in the default key policy.

- **kms:Decrypt** – (Required) Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted. This permission is included in the default key policy.
- **kms:ReEncrypt** – (Optional) Encrypts data on the server side with a new KMS key, without exposing the plaintext of the data on the client side. The data is first decrypted and then re-encrypted. This permission is included in the default key policy.
- **kms:GenerateDataKeyWithoutPlaintext** – (Required) Returns a data encryption key encrypted under a KMS key. This permission is included in the default key policy under **kms:GenerateDataKey***.
- **kms:CreateGrant** – (Required) Adds a grant to a key to specify who can use the key and under what conditions. Grants are alternate permission mechanisms to key policies. For more information on grants, see Using grants in the *AWS Key Management Service Developer Guide.* This permission is included in the default key policy.
- **kms:DescribeKey** – (Required) Provides detailed information about the specified KMS key. This permission is included in the default key policy.
- **kms:ListAliases** – (Optional) Lists all of the key aliases in the account. When you use the console to create an encrypted file system, this permission populates the list of KMS keys. We recommend using this permission to provide the best user experience. This permission is included in the default key policy.

Encryption in Transit

Encryption of data in transit is supported on file shares that are mapped on a compute instance that supports SMB protocol 3.0 or newer. This includes all Windows versions starting from Windows Server 2012 and Windows 8, and all Linux clients with Samba client version 4.2 or newer. Amazon FSx for Windows File Server automatically encrypts data in transit using SMB encryption as you access your file system without the need for you to modify your applications.

SMB encryption uses AES-128-GCM or AES-128-CCM (with the GCM variant being chosen if the client supports SMB 3.1.1) as its encryption algorithm, and also provides data integrity with signing using SMB Kerberos session keys. The use of AES-128-GCM leads to better performance, for example, up to a 2x performance improvement when copying large files over encrypted SMB connections.

To meet compliance requirements for always encrypting data-in-transit, you can limit file system access to only allow access to clients that support SMB encryption. You can also enable or disable in-transit encryption per file share or to the entire file system. This allows you to have a mix of encrypted and unencrypted file shares on the same file system. To learn more about managing encryption-in-transit on your file system, see Managing encryption in transit (p. 123).

File- and Folder-Level Access Control Using Windows ACLs

Amazon FSx for Windows File Server supports identity-based authentication over the Server Message Block (SMB) protocol through Microsoft Active Directory. Active Directory is the Microsoft directory service to store information about objects on the network and make this information easy for administrators and users to find and use. These objects typically include shared resources such as file servers, and the network user and computer accounts. To learn more about Active Directory support in Amazon FSx, see Working with Microsoft Active Directory in FSx for Windows File Server (p. 25).

Your domain-joined compute instances can access Amazon FSx file shares using Active Directory credentials. You use standard Windows access control lists (ACLs) for fine-grained file- and folder-level access control. Amazon FSx file systems automatically verify the credentials of users accessing file system data to enforce these Windows ACLs.

Every Amazon FSx file system comes with a default Windows file share called `share`. The Windows ACLs for this shared folder are configured to allow read/write access to domain users. They also allow full control to the delegated administrators group in your Active Directory that is delegated to perform administrative actions on your file systems. If you're integrating your file system with AWS Managed Microsoft AD, this group is AWS Delegated FSx Administrators. If you're integrating your file system with your self-managed Microsoft AD setup, this group can be Domain Admins. Or it can be a custom delegated administrators group that you specified when creating the file system. To change the ACLs, you can map the share as a user that is a member of the delegated administrators group.

> **Warning**
> Amazon FSx requires that the SYSTEM user have **Full control** NTFS ACL permissions on all folders within your file system. Do not change the NTFS ACL permissions for this user on your folders. Doing so can make your file share inaccessible and prevent file system backups from being usable.

Related Links

- What Is AWS Directory Service? in the AWS Directory Service Administration Guide.
- Create Your AWS Managed Microsoft AD directory in the *AWS Directory Service Administration Guide*.
- When to Create a Trust Relationship in the *AWS Directory Service Administration Guide*.
- Walkthrough 1: Prerequisites for getting started (p. 157).

File System Access Control with Amazon VPC

You access your Amazon FSx file system through an elastic network interface. This network interface resides in the virtual private cloud (VPC) based on the Amazon Virtual Private Cloud (Amazon VPC) service that you associate with your file system. You connect to your Amazon FSx file system through its Domain Name Service (DNS) name. The DNS name maps to the private IP address of the file system's elastic network interface in your VPC. Only resources within the associated VPC, resources connected with the associated VPC by AWS Direct Connect or VPN, or resources within peered VPCs can access your file system's network interface. For more information, see What is Amazon VPC? in the *Amazon VPC User Guide.*

> **Warning**
> You must not modify or delete the elastic network interface(s) associated with your file system. Modifying or deleting the network interface can cause a permanent loss of connection between your VPC and your file system.

FSx for Windows File Server supports VPC sharing, which enables you to view, create, modify, and delete resources in a shared subnet in a VPC owned by another AWS account. For more information, see Working with Shared VPCs in the *Amazon VPC User Guide*.

Amazon VPC Security Groups

To further control network traffic going through your file system's elastic network interface within your VPC, you use security groups to limit access to your file systems. A *security group* is a stateful firewall that controls the traffic to and from its associated network interfaces. In this case, the associated resource is your file system's network interface.

To use a security group to control access to your Amazon FSx file system, add inbound and outbound rules. Inbound rules control incoming traffic, and outbound rules control outgoing traffic from your

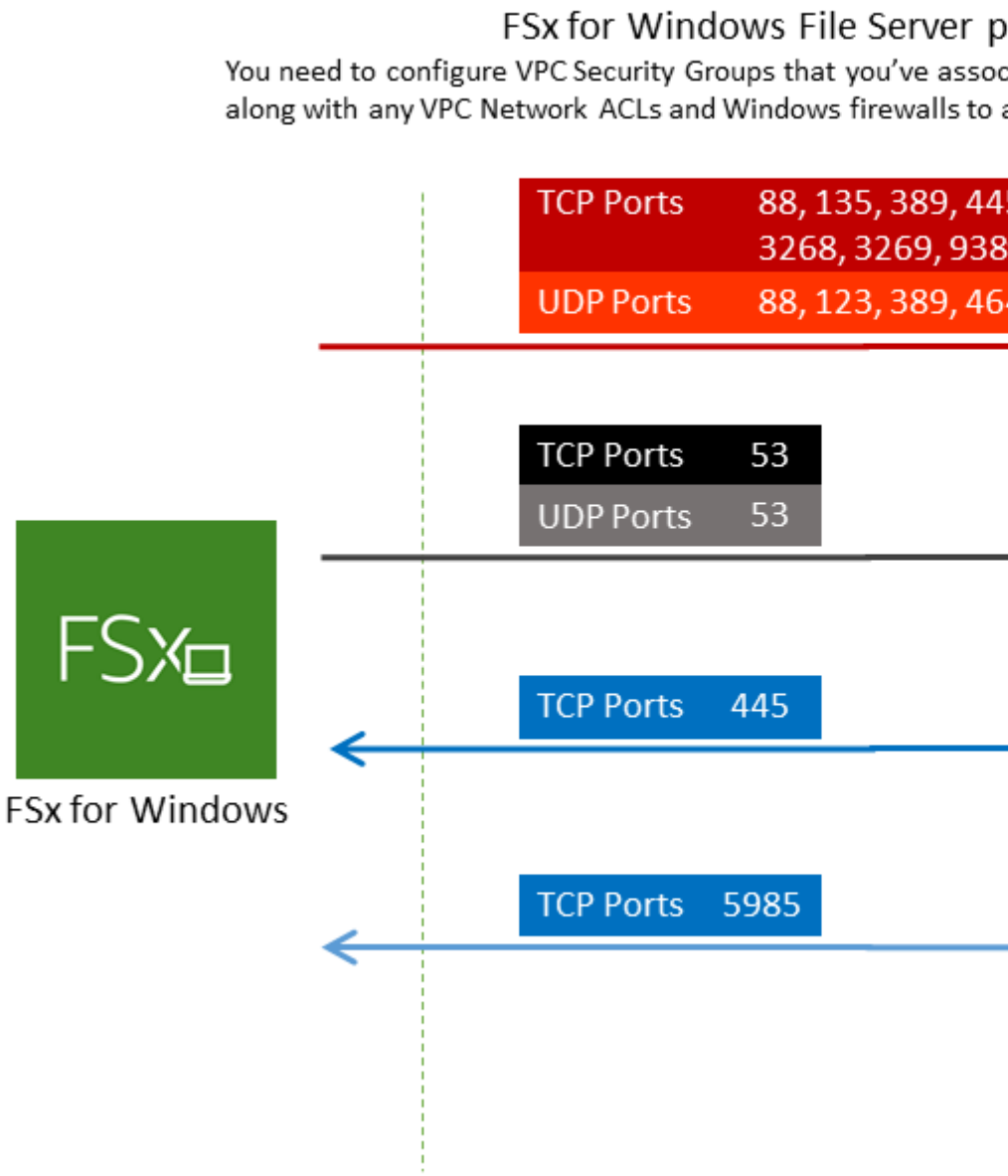
file system. Make sure that you have the right network traffic rules in your security group to map your Amazon FSx file system's file share to a folder on your supported compute instance.

For more information on security group rules, see Security Group Rules in the *Amazon EC2 User Guide for Linux Instances.*

To create a security group for Amazon FSx

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. Specify a name and description for the security group.
5. For **VPC**, choose the Amazon VPC associated with your file system to create the security group within that VPC.
6. Add the following rules to allow outbound network traffic on the following ports:

    a. For **VPC security groups**, the default security group for your default Amazon VPC is already added to your file system in the console. Please ensure that the security group and the VPC Network ACLs for the subnet(s) where you're creating your FSx file system allow traffic on the ports and in the directions shown in the following diagram.



The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Domain Name System (DNS)

Protocol	Ports	Role
TCP/UDP	88	Kerberos authentication
TCP/UDP	464	Change/Set password
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
UDP	123	Network Time Protocol (NTP)
TCP	135	Distributed Computing Environment / End Point Mapper (DCE / EPMAP)
TCP	445	Directory Services SMB file sharing
TCP	636	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
TCP	3268	Microsoft Global Catalog
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management)
TCP	9389	Microsoft AD DS Web Services, PowerShell
TCP	49152 - 65535	Ephemeral ports for RPC

Important
Allowing outbound traffic on TCP port 9389 is required for Single-AZ 2 and all Multi-AZ file system deployments.

b. Ensure that these traffic rules are also mirrored on the firewalls that apply to each of the AD domain controllers, DNS servers, FSx clients and FSx administrators.

Important
While Amazon VPC security groups require ports to be opened only in the direction that network traffic is initiated, most Windows firewalls and VPC network ACLs require ports to be open in both directions.

Note
If you have Active Directory sites defined, you must be sure that the subnet(s) in the VPC associated with your Amazon FSx file system are defined in an Active Directory site, and that no conflicts exist between the subnet(s) in your VPC and the subnets in your other sites. You can view and change these settings using the Active Directory Sites and Services MMC snap-in.

Note
In some cases, you might have modified the rules of your AWS Managed Microsoft AD security group from the default settings. If so, make sure that this security group has the required inbound rules to allow traffic from your Amazon FSx file system. For more information about the required inbound rules, see AWS Managed Microsoft AD Prerequisites in the *AWS Directory Service Administration Guide*.

Now that you've created your security group, you can associate it with your Amazon FSx file system's elastic network interface.

To associate a security group with your Amazon FSx file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the dashboard, choose your file system to view its details.
3. Choose the **Network & Security** tab, and choose your file system's network interface ID (for example, **ENI-01234567890123456**).
4. For **Actions**, choose **Change Security Groups**.
5. In the **Change Security Groups** dialog box, choose the security groups to use, and choose **Save**.

Disallow Access to a File System

To temporarily disallow network access to your file system from all clients, you can remove all the security groups associated with your file system's elastic network interface(s) and replace them with a group that has no inbound/outbound rules.

Amazon VPC Network ACLs

Another option for securing access to the file system within your VPC is to establish network access control lists (network ACLs). Network ACLs are separate from security groups, but have similar functionality to add an additional layer of security to the resources in your VPC. For more information on network ACLs, see Network ACLs in the *Amazon VPC User Guide.*

Resource administration access control with IAM for Amazon FSx

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to AWS Identity and Access Management (IAM) identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

Amazon FSx for Windows File Server resources and operations

In Amazon FSx for Windows File Server, the primary resource is a *file system*. Amazon FSx for Windows File Server also supports the additional subresource type *backup*. You can create backups only in the context of an existing file system, or by copying an existing backup.

These resources and subresources have unique Amazon Resource Names (ARNs) associated with them as shown in the following table.

Amazon FSx provides a set of operations to work with Amazon FSx resources. For a list of available operations, see the Amazon FSx API Reference.

Understanding resource ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the root account, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a file system, your AWS account is the owner of the resource (in Amazon FSx, the resource is the file system).
- If you create an IAM user in your AWS account and grant permissions to create a file system to that user, the user can create a file system. However, your AWS account, to which the user belongs, owns the file system resource.
- If you create an IAM role in your AWS account with permissions to create a file system, anyone who can assume the role can create a file system. Your AWS account, to which the role belongs, owns the file system resource.

Grant permission to tag resources during creation

Some resource-creating Amazon FSx API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based access control (ABAC). For more information, see What is ABAC for AWS in the *IAM User Guide*.

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `fsx:CreateFileSystem` or `fsx:CreateBackup`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `fsx:TagResource` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `fsx:TagResource` action.

The following example demonstrates a policy that allows users to create file systems and apply tags to file systems during creation in a specific AWS account.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateFileSystem",
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/*"
    }
  ]
}
```

Similarly, the following policy allows users to create backups on a specific file system and apply any tags to the backup during backup creation.

```
{
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateBackup"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/file-system-id*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:backup/*"
    }
  ]
}
```

The `fsx:TagResource` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `fsx:TagResource` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `fsx:TagResource` action.

For more information about tagging Amazon FSx resources, see Tag your Amazon FSx resources (p. 136). For more information about using tags to control access to FSx resources, see Using tags to control access to your Amazon FSx resources (p. 186).

Managing access to Amazon FSx resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of Amazon FSx for Windows File Server. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM policies) and policies attached to a resource are referred to as *resource-based* policies. Amazon FSx for Windows File Server supports only identity-based policies (IAM policies).

Amazon FSx API permissions: actions, resources, and conditions reference

When you are setting up access control and writing a permissions policy that you can attach to an IAM identity (identity-based policies), you can use the following as a reference. The each Amazon FSx API operation, the corresponding actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

You can use AWS-wide condition keys in your Amazon FSx policies to express conditions. For a complete list of AWS-wide keys, see Available Keys in the *IAM User Guide*.

To specify an action, use the `fsx:` prefix followed by the API operation name (for example, `fsx:CreateFileSystem`). Each action applies to either a single Amazon FSx file system, to all Amazon FSx file systems owned by an AWS account, to a single backup, or to all backups owned by an AWS account.

This section only includes the Amazon FSx permissions required for these actions. Additional permissions from othe AWS services are required for some of these actions.

Amazon FSx API and required permissions for actions

Amazon FSx API operation	Required permissions (API actions)	Resource
AssociateFileSystemAliases	fsx:AssociateFileSystemAliases	arn:aws:fsx:*region*:*account-id*:file-system/*file-system-id*
CancelDataRepositoryTask	fsx:CancelDataRepositoryTask	arn:aws:fsx:*region*:*account-id*:file-system/*file-system-id*
CopyBackup	fsx:CopyBackup fsx:CopyBackup fsx:TagResource	arn:aws:fsx:*region*:*account-id*:backup/*source-backup-id* – the source backup arn:aws:fsx:*region*:*account-id*:backup/* – the destination region arn:aws:fsx:*region*:*account-id*:backup/* – required to copy or create tags on the backup copy
CreateBackup	fsx:CreateBackup fsx:CreateBackup fsx:TagResource	arn:aws:fsx:*region*:*account-id*:backup/* arn:aws:fsx:*region*:*account-id*:file-system/*file-system-id* arn:aws:fsx:*region*:*account-id*:backup/* – required to create tags on the new backup
CreateFileSystem	fsx:CreateFileSystem fsx:TagResource	arn:aws:fsx:*region*:*account-id*:file-system/* arn:aws:fsx:*region*:*account-id*:file-system/* – to create tags on the file system
CreateFileSystemFromBackup	fsx:CreateFileSystemFromBackup fsx:CreateFileSystemFromBackup fsx:TagResource	arn:aws:fsx:*region*:*account-id*:file-system/* arn:aws:fsx:*region*:*account-id*:backup/* arn:aws:fsx:*region*:*account-id*:file-system/* – to create tags on the file system
DeleteBackup	fsx:DeleteBackup	arn:aws:fsx:*region*:*account-id*:backup/*backup-id*

Amazon FSx API operation	Required permissions (API actions)	Resource

DeleteFileSystem	`fsx:DeleteFileSystem`
	
	`fsx:TagResource`
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		
		`arn:aws:fsx:region:account-id:backup/*` – required to create tags on a final backup if created

DescribeBackups	`fsx:DescribeBackups`	`arn:aws:fsx:region:account-id:backup/*`
DescribeFileSystemAliases	`fsx:DescribeFileSystemAliases`	`arn:aws:fsx:region:account-id:file-system/file-system-id`
DescribeFileSystems	`fsx:DescribeFileSystems`	`arn:aws:fsx:region:account-id:file-system/*`
DisassociateFileSystemAliases	`fsx:DisassociateFileSystemAliases`	`arn:aws:fsx:region:account-id:file-system/*`

ListTagsForResource	`fsx:ListTagsForResource`	`arn:aws:fsx:region:account-id:backup/backup-id`
		
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		
		`arn:aws:fsx:region:account-id:task/task-id`

TagResource	`fsx:TagResource`	`arn:aws:fsx:region:account-id:backup/backup-id`
		
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		
		`arn:aws:fsx:region:account-id:task/task-id`

UntagResource	`fsx:UntagResource`	`arn:aws:fsx:region:account-id:backup/backup-id`
		
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		
		`arn:aws:fsx:region:account-id:task/task-id`

UpdateFileSystem	`fsx:UpdateFileSystem`	`arn:aws:fsx:region:account-id:file-system/filesystem-id`

Using tags to control access to your Amazon FSx resources

To control access to Amazon FSx resources and actions, you can use AWS Identity and Access Management (IAM) policies based on tags. You can provide the control in two ways:

1. Control access to Amazon FSx resources based on the tags on those resources.
2. Control what tags can be passed in an IAM request condition.

For information about how to use tags to control access to AWS resources, see Controlling access using tags in the *IAM User Guide*. For more information about tagging Amazon FSx resources at creation, see Grant permission to tag resources during creation (p. 182). For more information about using tags, see Tag your Amazon FSx resources (p. 136).

Controlling access based on tags on a resource

To control what actions a user or role can perform on an Amazon FSx resource, you can use tags on the resource. For example, you might want to allow or deny specific API operations on a file system resource based on the key-value pair of the tag on the resource.

Example Example policy – Create a file system on when providing a specific tag

This policy allows the user to create a file system only when they tag it with a specific tag key value pair, in this example, `key=Department, value=Finance`.

```
{
    "Effect": "Allow",
    "Action": [
        "fsx:CreateFileSystem",
        "fsx:TagResource"
    ],
    "Resource": "arn:aws:fsx:region:account-id:file-system/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Department": "Finance"
        }
    }
}
```

Example Example policy – Create backups only on file systems with a specific tag

This policy allows users to create backups only on file systems that are tagged with the key value pair `key=Department, value=Finance`, and the backup will be created with the tag `Deparment=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource",
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }

    ]
}
```

Example Example policy – Create a file system with a specific tag from backups with a specific tag

This policy allows users to create file systems that are tagged with `Department=Finance` only from backups that are tagged with `Department=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Example Example policy – Delete file systems with specific tags

This policy allows a user to delete only file systems that are tagged with `Department=Finance`. If they create a final backup, then it must be tagged with `Department=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "fsx:DeleteFileSystem"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Using service-linked roles for Amazon FSx

Amazon FSx for Windows File Server uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon FSx. Service-linked roles are predefined by Amazon FSx and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon FSx easier because you don't have to manually add the necessary permissions. Amazon FSx defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon FSx can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Amazon FSx resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Amazon FSx

Amazon FSx uses the service-linked role named **AWSServiceRoleForAmazonFSx** – Which performs certain actions in your account, like creating Elastic Network Interfaces for your file systems in your VPC.

The role permissions policy allows Amazon FSx to complete the following actions on the all applicable AWS resources:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ds:AuthorizeApplication",
```

```
            "ds:GetAuthorizedApplicationDetails",
            "ds:UnauthorizeApplication",
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DescribeAddresses",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVPCs",
            "ec2:DisassociateAddress",
            "route53:AssociateVPCWithHostedZone"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:network-interface/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction": "CreateNetworkInterface"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "AmazonFSx.FileSystemId"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:AssignPrivateIpAddresses",
            "ec2:ModifyNetworkInterfaceAttribute",
            "ec2:UnassignPrivateIpAddresses"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:network-interface/*"
        ],
        "Condition": {
            "Null": {
                "aws:ResourceTag/AmazonFSx.FileSystemId": "false"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateRoute",
            "ec2:ReplaceRoute",
            "ec2:DeleteRoute"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:route-table/*"
        ],
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/AmazonFSx": "ManagedByAmazonFSx"
            }
        }
    },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/fsx/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:DescribeDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/aws-fsx-*"
        }
    ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for Amazon FSx

You don't need to manually create a service-linked role. When you create a file system in the AWS Management Console, the IAM CLI, or the IAM API, Amazon FSx creates the service-linked role for you.

> **Important**
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. To learn more, see A New Role Appeared in My IAM Account.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a file system, Amazon FSx creates the service-linked role for you again.

Editing a service-linked role for Amazon FSx

Amazon FSx does not allow you to edit the AWSServiceRoleForAmazonFSx service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for Amazon FSx

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must delete all of your file systems and backups before you can manually delete the service-linked role.

> **Note**
> If the Amazon FSx service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the AWSServiceRoleForAmazonFSx service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported regions for Amazon FSx service-linked roles

Amazon FSx supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

AWS managed policies for Amazon FSx

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the `ViewOnlyAccess` AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AmazonFSxDeleteServiceLinkedRoleAccess

You can't attach `AmazonFSxDeleteServiceLinkedRoleAccess` to your IAM entities. This policy is linked to a service service and used only with the service-linked role for that service. You cannot attach, detach, modify, or delete this policy. For more information, see Using service-linked roles for Amazon FSx (p. 188).

This policy grants administrative permissions that allow Amazon FSx to delete its Service Linked Role for Amazon S3 access, used only by Amazon FSx for Lustre.

Permissions details

This policy includes permissions in `iam` to allow Amazon FSx to view, delete, and view the deletion status for the FSx Service Linked Roles for Amazon S3 access.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:DeleteServiceLinkedRole",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:GetRole"
```

```
            ],
            "Resource": "arn:*:iam::*:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/AWSServiceRoleForFSxS3Access_*"
        }
    ]
}
```

AWS managed policy: AmazonFSxFullAccess

You can attach AmazonFSxFullAccess to your IAM entities. Amazon FSx also attaches this policy to a service role that allows Amazon FSx to perform actions on your behalf.

Provides full access to Amazon FSx and access to related AWS services.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals full access to perform all Amazon FSx actions.
- `ds` – Allows principals to view information about the AWS Directory Service directories.
- `iam` – Allows principles to create an Amazon FSx service linked role on the user's behalf. This is required so that Amazon FSx can manage AWS resources on the user's behalf.
- `logs` – Allows principals to create log groups, log streams, and write events to log streams. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to CloudWatch Logs.
- `firehose` – Allows principals to write records to a Amazon Kinesis Data Firehose. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to Kinesis Data Firehose.
- `ec2` – Allows principals to create tags under the specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ds:DescribeDirectories",
                "fsx:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "s3.data-source.lustre.fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/fsx/*:log-group:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord"
            ],
            "Resource": [
                "arn:aws:firehose:*:*:deliverystream/aws-fsx-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonFSx": "ManagedByAmazonFSx"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["fsx.amazonaws.com"]
                }
            }
        }
    ]
}
```

AWS managed policy: AmazonFSxConsoleFullAccess

You can attach the `AmazonFSxConsoleFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow full access to Amazon FSx and access to related AWS services via the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to perform all actions in the Amazon FSx management console.
- `cloudwatch` – Allows principals to view CloudWatch Alarms in the Amazon FSx management console.

- `ds` – Allows principals to list information about an AWS Directory Service directory.
- `ec2` – Allows principals to create tags on route tables, list network interfaces, route tables, security groups, subnets and the VPC associated with an Amazon FSx file system.
- `kms` – Allows principals to list aliases for AWS Key Management Service keys.
- `s3` – Allows principals to list some or all of the objects in an Amazon S3 bucket (up to 1000).
- `iam` – Grants permission to create a service linked role that allows Amazon FSx to perform actions on the user's behalf.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "ds:DescribeDirectories",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "firehose:ListDeliveryStreams",
                "fsx:*",
                "kms:ListAliases",
                "logs:DescribeLogGroups",
                "s3:ListBucket"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "s3.data-source.lustre.fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "aws:RequestTag/AmazonFSx": "ManagedByAmazonFSx"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["fsx.amazonaws.com"]
                }
            }
        }
    ]
}
```

AWS managed policy: AmazonFSxConsoleReadOnlyAccess

You can attach the `AmazonFSxConsoleReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions to Amazon FSx and related AWS services so that users can view information about these services in the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to view information about Amazon FSx file systems, including all tags, in the Amazon FSx Management Console.
- `cloudwatch` – Allows principals to view CloudWatch Alarms in the Amazon FSx Management Console.
- `ds` – Allows principals to view information about an AWS Directory Service directory in the Amazon FSx Management Console.
- `ec2` – Allows principals to view network interfaces, security groups, subnets and the VPC associated with an Amazon FSx file system in the Amazon FSx Management Console.
- `kms` – Allows principals to view aliases for AWS Key Management Service keys in the Amazon FSx Management Console.
- `log` – Allows principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.
- `firehose` – Allows principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "ds:DescribeDirectories",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "firehose:ListDeliveryStreams",
                "fsx:Describe*",
```

```
                "fsx:ListTagsForResource",
                "kms:DescribeKey",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: AmazonFSxReadOnlyAccess

You can attach the `AmazonFSxReadOnlyAccess` policy to your IAM identities.

This policy grants administrative permissions that allow read-only access to Amazon FSx.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:Describe*",
                "fsx:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon FSx updates to AWS managed policies

View details about updates to AWS managed policies for Amazon FSx since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon FSx Document history (p. 232) page.

Change	Description	Date
AmazonFSxReadOnlyAccess (p. 196) – Started tracking policy	This policy grants read-only access to all Amazon FSx resources and any tags associated with them.	February 4, 2022
AmazonFSxDeleteServiceLinkedRoleAccess (p. 191) – Started tracking policy	This policy grants administrative permissions that allow Amazon FSx to delete its Service Linked Role for Amazon S3 access.	January 7, 2022
AmazonFSxServiceRolePolicy (p. 188) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to manage network configurations for Amazon FSx for NetApp ONTAP file systems.	September 2, 2021
AmazonFSxFullAccess (p. 192) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021

Change	Description	Date
AmazonFSxConsoleFullAccess (p. 191) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to create Amazon FSx for NetApp ONTAP Multi-AZ file systems.	September 2, 2021
AmazonFSxConsoleFullAccess (p. 191) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021
AmazonFSxServiceRolePolicy (p. 188) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to CloudWatch Logs log streams. This is required so that users can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021
AmazonFSxServiceRolePolicy (p. 188) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to Amazon Kinesis Data Firehose delivery streams. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Kinesis Data Firehose.	June 8, 2021
AmazonFSxFullAccess (p. 192) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe and create CloudWatch Logs log groups, log streams, and write events to log streams. This is required so that principals can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021
AmazonFSxFullAccess (p. 192) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe and write records to a Amazon Kinesis Data Firehose. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Kinesis Data Firehose.	June 8, 2021

Change	Description	Date
AmazonFSxConsoleFullAccess (p. 194) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can choose an existing CloudWatch Logs log group when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleFullAccess (p. 194) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can choose an existing Kinesis Data Firehose delivery stream when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleReadOnlyAccess (p. 195) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleReadOnlyAccess (p. 195) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021

Change	Description	Date
Amazon FSx started tracking changes	Amazon FSx started tracking changes for its AWS managed policies.	June 8, 2021

Compliance Validation for Amazon FSx for Windows File Server

Third-party auditors assess the security and compliance of Amazon FSx for Windows File Server as part of multiple AWS compliance programs. These include SOC, PCI, ISO, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using Amazon FSx 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 and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- 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 that helps you check your compliance with security industry standards and best practices.

Amazon FSx for Windows File Server and interface VPC endpoints

You can improve the security posture of your VPC by configuring Amazon FSx to use an interface VPC endpoint. Interface VPC endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Amazon FSx APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon FSx APIs. Traffic between your VPC and Amazon FSx does not leave the AWS network.

Each interface VPC endpoint is represented by one or more elastic network interfaces in your subnets. A network interface provides a private IP address that serves as an entry point for traffic to the Amazon FSx API.

Considerations for Amazon FSx interface VPC endpoints

Before you set up an interface VPC endpoint for Amazon FSx, be sure to review Interface VPC endpoint properties and limitations in the *Amazon VPC User Guide*.

You can call any of the Amazon FSx API operations from your VPC. For example, you can create an FSx for Windows File Server file system by calling the CreateFileSystem API from within your VPC. For the full list of Amazon FSx APIs, see Actions in the Amazon FSx API Reference.

VPC peering considerations

You can connect other VPCs to the VPC with interface VPC endpoints using VPC peering. VPC peering is a networking connection between two VPCs. You can establish a VPC peering connection between your own two VPCs, or with a VPC in another AWS account. The VPCs can also be in two different AWS Regions.

Traffic between peered VPCs stays on the AWS network and does not traverse the public internet. Once VPCs are peered, resources like Amazon Elastic Compute Cloud (Amazon EC2) instances in both VPCs can access the Amazon FSx API through interface VPC endpoints created in the one of the VPCs.

Creating an interface VPC endpoint for Amazon FSx API

You can create a VPC endpoint for the Amazon FSx API using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface VPC endpoint in the *Amazon VPC User Guide*.

To create an interface VPC endpoint for Amazon FSx, use one of the following:

- **`com.amazonaws.`*`region`*`.fsx`** – Creates an endpoint for Amazon FSx API operations.
- **`com.amazonaws.`*`region`*`.fsx-fips`** – Creates an endpoint for the Amazon FSx API that complies with Federal Information Processing Standard (FIPS) 140-2.

To use the private DNS option, you must set the `enableDnsHostnames` and `enableDnsSupport` attributes of your VPC. For more information, see Viewing and updating DNS support for your VPC in the *Amazon VPC User Guide*.

Excluding AWS Regions in China, if you enable private DNS for the endpoint, you can make API requests to Amazon FSx with the VPC endpoint using its default DNS name for the AWS Region, for example `fsx.us-east-1.amazonaws.com`. For the China (Beijing) and China (Ningxia) AWS Regions, you can make API requests with the VPC endpoint using `fsx-api.cn-north-1.amazonaws.com.cn` and `fsx-api.cn-northwest-1.amazonaws.com.cn`, respectively.

For more information, see Accessing a service through an interface VPC endpoint in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Amazon FSx

To further control access to the Amazon FSx API, you can optionally attach an AWS Identity and Access Management (IAM) policy to your VPC endpoint. The policy specifies the following:

- The principal that can perform actions.

- The actions that can be performed.
- The resources upon which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Quotas

Following, you can find out about quotas when working with Amazon FSx for Windows File Server.

Topics

Quotas that you can increase

Following are the quotas for Amazon FSx for Windows File Server for each AWS account, per AWS Region, that you can increase.

Resource	Default	Description
Windows file systems	100	The maximum number of Amazon FSx for Windows Server file systems that you can create in this account.
Windows throughput capacity	10240	The total amount of throughput capacity (in MBps) allowed for all Amazon FSx for Windows file systems in this account.
Windows HDD storage capacity	524288	The maximum amount of HDD storage capacity (in GiB) allowed for all Amazon FSx for Windows File Server file systems in this account.
Windows SSD storage capacity	524288	The maximum amount of SSD storage capacity (in GiB) allowed for all Amazon FSx for Windows File Server file systems in this account.
Windows backups	500	The maximum number of user-initiated backups for all Amazon FSx for Windows File Server file systems that you can have in this account.

To request a quota increase

1. Open the Service Quotas console.
2. In the navigation pane, choose **AWS services**.
3. Choose **Amazon FSx**.

4. Choose a quota.
5. Choose **Request quota increase**, and follow the directions to request a quota increase.
6. To view the status of the quota request, choose **Quota request history** in the console navigation pane.

For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Resource quotas for each file system

Following are the quotas on Amazon FSx for Windows File Server resources for each file system in an AWS Region.

Resource	Limit per file system
Maximum number of tags	50
Maximum retention period for automated backups	90 days
Maximum number of backup copy requests in progress to a single destination Region per account.	5
Minimum storage capacity, SSD file systems	32 GiB
Minimum storage capacity, HDD file systems	2,000 GiB
Maximum storage capacity, SSD and HDD	64 TiB
Minimum throughput capacity	8 MBps
Maximum throughput capacity	2,048 MBps
Maximum number of file shares	100,000

Additional considerations

In addition, note the following:

- You can use each AWS Key Management Service (AWS KMS) key on up to 125 Amazon FSx file systems.
- For a list of AWS Regions where you can create file systems, see Amazon FSx Endpoints and Quotas in the *AWS General Reference*.
- You map your file shares from Amazon EC2 instances in your virtual private cloud (VPC) with their Domain Name Service (DNS) names.

Quotas specific to Microsoft Windows

For more information, see NTFS limits on the Microsoft Windows Dev Center.

Troubleshooting Amazon FSx

Use the following sections to help troubleshoot problems you have with Amazon FSx.

If you encounter problems not listed following while using Amazon FSx, try asking a question in the Amazon FSx forum.

Topics

You can't access your file system

There are a number of potential causes for being unable to access your file system, each with their own resolution, as follows.

Topics

The file system elastic network interface was modified or deleted

You must not modify or delete the file system's elastic network interface. Modifying or deleting the network interface can cause a permanent loss of connection between your VPC and your file system.

Create a new file system, and do not modify or delete the Amazon FSx elastic network interface. For more information, see File System Access Control with Amazon VPC (p. 177).

The Elastic IP address attached to the file system elastic network interface was deleted

Amazon FSx doesn't support accessing file systems from the public internet. Amazon FSx automatically detaches any Elastic IP address, which is a public IP address reachable from the internet, that gets attached to a file system's elastic network interface. For more information, see Supported clients, access methods, and environments for Amazon FSx for Windows File Server (p. 15).

The file system security group lacks the required inbound or outbound rules.

Review the inbound rules specified in Amazon VPC Security Groups (p. 177), and make sure that the security group associated with your file system has the corresponding inbound rules.

The compute instance's security group lacks the required outbound rules

Review the outbound rules specified in Amazon VPC Security Groups (p. 177), and make sure that the security group associated with your compute instance has the corresponding outbound rules.

Compute instance not joined to an Active Directory

Your compute instances might not be correctly joined to one of two types of Active Directory:

- The AWS Managed Microsoft AD directory to which your file system is joined.
- A Microsoft Active Directory directory that has a one-way forest trust relationship established with the AWS Managed Microsoft AD directory.

Make sure that your compute instances are joined to one of two types of directory. One type is the AWS Managed Microsoft AD directory to which your file system is joined. The other type is a Microsoft AD directory that has a one-way forest trust relationship established with the AWS Managed Microsoft AD directory. For more information, see Using Amazon FSx with AWS Directory Service for Microsoft Active Directory (p. 26).

The file share doesn't exist

The Microsoft Windows file share that you're attempting to access doesn't exist.

If you're using an existing file share, make sure that the file system DNS name and the share name are correctly specified. To manage your file shares, see File shares (p. 95).

Active Directory user lacks required permissions

The Active Directory user that you're accessing the file share as lacks the necessary access permissions.

Make sure that the access permissions for the file share and Windows access control lists (ACLs) for the shared folder allow access to the Active Directory users that need to access it.

Allow Full control NTFS ACL permissions removed

If you remove **Allow Full control** NTFS ACL permissions for the SYSTEM user on a folder that you shared, that share can become inaccessible and any file system backups taken from that point onwards may not be usable.

You will need to re-create the affected file share. For more information, see File shares (p. 95). After you recreate the folder or share, you can map and use the Windows file shares from your compute instances.

Can't access a file system using an on-premises client

You're using your Amazon FSx file system from on-premises using AWS Direct Connect or VPN, and you're using a non-private IP address range for the on-premises client.

Amazon FSx only supports access from on-premises clients with non-private IP addresses on file systems created after December 17, 2020.

If you need to access your FSx for Windows File Server file system that was created before December 17, 2020 using a non-private IP address range, you can create a new file system by restoring a backup of the file system. For more information, see Working with backups (p. 78).

New file system is not registered in DNS

For file systems joined to a self-managed Active Directory, Amazon FSx did not register the file system DNS when it was created because the customer network does not use Microsoft DNS.

Amazon FSx does not register file systems in DNS if your network uses a third-party DNS service instead of Microsoft DNS. You must manually set up DNS A entries for your Amazon FSx file systems. For Single-AZ 1 file systems, you will need to add one DNS A entry; for Single-AZ 2 and Multi-AZ file systems, you will need to add two DNS A entries. Use the following procedure to obtain the file system IP address or addresses to use when manually adding the DNS A entries.

1. In the https://console.aws.amazon.com/fsx/, choose the file system that you want to obtain the IP address of to display the file system details page.
2. In the **Network & security** tab do one of the following:

   - For a Single-AZ 1 file system:
     - In the **Subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 .
     - The IP address for the Single-AZ 1 file system to use is shown in the **Primary private IPv4 IP** column.
   - For a Single-AZ 2 or Multi-AZ file system:
     - In the **Preferred subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 .
     - The IP address for the preferred subnet to use is shown in the **Secondary private IPv4 IP** column.
     - In the Amazon FSx **Standby subnet** panel, choose the elastic network interface shown under **Network interface** to open the **Network Interfaces** page in the Amazon EC2 console.
     - The IP address for the standby subnet to use is shown in the **Secondary private IPv4 IP** column.

Can't access the file system using a DNS alias

If you're unable to access a file system using a DNS alias, use the following procedure to troubleshoot the issue.

1. Verify that the alias is associated with the file system by doing either of the following steps:

   a. **Using the Amazon FSx console** – Choose the file system that you're trying to access. On the **File system details** page, the **DNS aliases** are shown on the **Network & security** tab.

   b. **Using the CLI or API** – Use the `describe-file-system-aliases` CLI command, or the DescribeFileSystemAliases API operation to retrieve the aliases currently associated with the file system.

2. If the DNS alias is not listed, you must associate it with the file system. For more information, see Managing DNS aliases on existing file systems (p. 93).

3. If the DNS alias is associated with the file system, verify that you've also configured the following required items:

   - Created service principal names (SPNs) corresponding to the DNS alias on your Amazon FSx file system's Active Directory computer object.

     For more information, see Step 2: Configure service principal names (SPNs) for Kerberos (p. 167).

   - Created a DNS CNAME record for the DNS alias that resolves to the default DNS name of the Amazon FSx file system.

     For more information, see Step 3: Update or create a DNS CNAME record for the file system (p. 169).

4. If you created valid SPNs and a DNS CNAME record, verify that the client's DNS has the DNS CNAME record that resolves to the correct file system.

   a. Run `nslookup` to confirm that the record exists and that it resolves to the file system's default DNS name.

   b. If the DNS CNAME resolves to another file system, wait for the client's DNS cache to refresh, and then check the CNAME record again. You can accelerate the process by flushing the client's DNS cache using the following command.

   ```
   ipconfig /flushdns
   ```

5. If the DNS CNAME record resolves to the Amazon FSx file system's default DNS, and the client is still unable to access the file system, see You can't access your file system (p. 204) for additional troubleshooting steps.

Trying to create an Amazon FSx file system fails

There are a number of potential causes when a file system creation request fails, as described in the following section.

Topics

Troubleshooting file systems joined to an AWS Managed Microsoft Active Directory

Use the following sections to help troubleshoot problems trying to create an FSx for Windows File Server file system joined to your self-managed Active Directory.

VPC security groups and network ACLs aren't using recommended security group configuration

Make sure that the VPC security groups and network ACLs are configured using the recommended security group configuration. For more information, see Creating FSx Security Groups, Step 6 (p. 178).

Troubleshooting file systems joined to a self-managed Active Directory

Topics

Amazon FSx can't reach self-managed AD DNS server or domain controllers. File system creation failed.

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't reach the DNS servers provided or the domain controllers for
your self-managed directory in Microsoft Active Directory. File system creation failed.
 Amazon FSx is
unable to communicate with your Microsoft Active Directory domain controllers.
This is because Amazon FSx can't reach the DNS servers provided or domain controllers
for your domain. To fix this problem, delete your file system and create a new
one with valid DNS servers and networking configuration that allows traffic from
the file system to the domain controller.
```

Use the following steps to troubleshoot and resolve the issue.

1. Verify that you followed the prerequisites for having network connectivity and routing established between the subnet where you're creating an Amazon FSx file system, and your self-managed Active Directory. For more information, see Prerequisites for using a self-managed Microsoft AD (p. 34).

   Use the Amazon FSx Active Directory Validation tool (p. 40) to test and verify these network settings.

Note
If you have multiple Active Directory sites defined, ensure that the subnets in the VPC associated with your Amazon FSx file system are defined in an Active Directory site and that no IP conflicts exist between the subnets in your VPC and the subnets in your other sites. You can view and change these settings using the Active Directory Sites and Services MMC snap-in.

2. Verify that you configured the VPC security groups that you associated with your Amazon FSx file system, along with any VPC network ACLs, to allow outbound network traffic on all ports.

   Note
   If you want to implement least privilege, you can allow outbound traffic only to the specific ports required for communication with the Active Directory domain controllers. For more information, see the Microsoft Active Directory documentation.

3. Verify that the values for Microsoft Windows file server or network administrative properties do not contain non-Latin-1 characters. For example, the file system creation fails if you use `Domänen-Admins` as the name of the file system administrators group.
4. Verify that your Active Directory domain's DNS servers and domain controllers are active and able to respond to requests for the domain provided.
5. Ensure that the functional level of your Active Directory domain is Windows Server 2008 R2 or higher.
6. Make sure that the firewall rules on your Active Directory domain's domain controllers allow traffic from your Amazon FSx file system. For more information, see the Microsoft Active Directory documentation.

Can't connect to Microsoft AD domain controllers due to invalid service account credentials

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to establish a connection with your Microsoft
Active Directory domain controllers because the service account credentials provided are
invalid. To fix this problem, delete your file system and create a new one using a valid
 service
account.
```

Use the following steps to troubleshoot and resolve the issue.

1. Verify that you're entering only the user name as input for the **Service account username**, such as `ServiceAcct`, in the self-managed Active Directory configuration.

   Important
   DO NOT include a domain prefix (`corp.com\ServiceAcct`) or domain suffix (`ServiceAcct@corp.com`) when entering the service account user name.
   DO NOT use the distinguished name (DN) when entering the service account user name (CN=ServiceAcct,OU=example,DC=corp,DC=com).

2. Verify that the service account that you provided exists in your Active Directory domain.
3. Make sure that you delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain to which you're joining the file system. The service account also needs, at a minimum, to have permissions to do the following:

   - Reset passwords
   - Restrict accounts from reading and writing data

- Validated ability to write to the DNS hostname
- Validated ability to write to the service principal name

For more information about creating a service account with correct permissions, see Delegating privileges to your Amazon FSx service account (p. 38).

Amazon FSx can't connect to Microsoft AD domain controllers due to insufficient service account permissions

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to establish a connection with your
Microsoft Active Directory domain controllers. This is because the service account provided
 does not
have permission to join the file system to the domain with the specified organizational
 unit.
To fix this problem, delete your file system and create a new one using a service account
 with
permission to join the file system to the domain with the specified organizational unit.
```

Use the following procedure to troubleshoot and resolve the issue.

- Make sure that you delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain to which you're joining the file system. The service account also needs, at a minimum, to have permissions to do the following:

  - Reset passwords
  - Restrict accounts from reading and writing data
  - Validated ability to write to the DNS hostname
  - Validated ability to write to the service principal name

  For more information about creating a service account with correct permissions, see Delegating privileges to your Amazon FSx service account (p. 38).

Amazon FSx can't connect to the Microsoft AD domain controllers because the service account provided can't join any more computers to the domain

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't establish a connection with your Microsoft Active Directory
domain controllers. This is because the service account provided has reached the
maximum number of computers that it can join to the domain. To fix this problem,
delete your file system and create a new one, supplying a service account that
is able to join new computers to the domain.
```

To resolve the issue, verify that the service account you provided has reached the maximum number of computers it can join to the domain. If it has reached the maximum limit, create a new service account

with the correct permissions. Use the new service account and create a new file system. For more information, see Delegating privileges to your Amazon FSx service account (p. 38).

Amazon FSx can't connect to the Microsoft AD domain controllers because the organizational unit specified doesn't exist or isn't accessible

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't establish a connection with your Microsoft Active Directory domain
 controller(s).
This is because the organizational unit you specified either doesn't exist or isn't
 accessible
to the service account provided. To fix this problem, delete your file system and create a
 new one specifying an
organizational unit to which the service account can join the file system.
```

Use the following steps to troubleshoot and resolve the issue.

1. Verify that the OU you provided is in your Active Directory domain.
2. Make sure that you have delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain that you're joining the file system to. The service account also needs to have, at a minimum, permissions to do the following:

   - Reset passwords
   - Restrict accounts from reading and writing data
   - Validated ability to write to the DNS hostname
   - Validated ability to write to the service principal name
   - Be delegated control to create and delete computer objects
   - Validated ability to read and write Account Restrictions

   For more information about creating a service account with the correct permissions, see Delegating privileges to your Amazon FSx service account (p. 38).

Amazon FSx can't apply the Microsoft AD configuration because the file system administrators group doesn't exist or isn't accessible to the service account

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to apply your Microsoft Active Directory configuration. This is
 because the file system
administrators group you provided either doesn't exist or isn't accessible to the service
 account you
provided. To fix this problem, delete your file system and create a new one specifying a
 file
system administrators group in the domain that is accessible to the service account
provided.
```

Use the following steps to troubleshoot and resolve the issue.

1. Ensure that you're providing just the name of the group as a string for the administrators group parameter.

   Important
   DO NOT include a domain prefix (`corp.com\FSxAdmins`) or domain suffix (`FSxAdmins@corp.com`) when providing the group name parameter.
   DO NOT use the distinguished name (DN) for the group. An example of a distinguished name is CN=FSxAdmins,OU=example,DC=corp,DC=com.
2. Ensure that the administrators group provided exists in the same Active Directory domain as the one that you want to join the file system to.
3. If you did not provide an administrator group parameter, Amazon FSx attempts to use the `Builtin Domain Admins` group in your Active Directory domain. If the name of this group has been changed, or if you're using a different group for domain administration, you need to provide that name for the group.

Amazon FSx can't apply your Microsoft Active Directory configuration.

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to apply your Microsoft Active
Directory configuration. To fix this problem, delete your file system and create a new one
meeting the pre-requisites described in the Amazon FSx user guide.
```

When creating your file system, Amazon FSx was able to reach your Active Directory domain's DNS servers and domain controllers, and join the file system successfully to your Active Directory domain. However, while completing file system creation, Amazon FSx lost connectivity to or membership in your domain. Use the following steps to troubleshoot and resolve the issue.

1. Ensure that network connectivity continues to exist between your Amazon FSx file system and your Active Directory. And, ensure that network traffic continues to be allowed between them by using routing rules, VPC security group rules, VPC network ACLs, and domain controller firewall rules.
2. Ensure that the computer objects created by Amazon FSx for your file systems in your Active Directory domain are still active, and were not deleted or otherwise manipulated.

File system creation failed. The service account provided does not have permission to join the file system to the domain with the specified organizational unit (OU)

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
File system creation failed. Amazon FSx is unable to establish a connection with your
 Microsoft Active Directory domain controller(s).
This is because the service account provided does not have permission to join the file
 system to the domain with the specified
organizational unit (OU). To fix this problem, delete your file system and create a new one
 using a service account with permission
to create computer objects and reset passwords within the specified organizational unit.
```

Make sure that you have delegated the required permissions to the service account that you provided. Use the following steps to troubleshoot and resolve the issue.

The service account needs to have, at a minimum, the following permissions:

- Be delegated control to create and delete computer objects in the OU that you're joining the file system to
- Have the following permissions in the OU that you're joining the file system to:
  - Ability to reset passwords
  - Ability to restrict accounts from reading and writing data
  - Validated ability to write to the DNS hostname
  - Validated ability to write to the service principal name

  For more information about creating a service account with the correct permissions, see Delegating privileges to your Amazon FSx service account (p. 38).

Amazon FSx is unable to create a file system within the specified Microsoft Active Directory.

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
File system creation failed. Amazon FSx is unable to create a file system within the
 specified
Microsoft Active Directory. To fix this problem, please delete your file system and create
 a new one
meeting the pre-requisites described in the Amazon FSx user guide.
```

Amazon FSx does not support Unicode characters. Verify that none of the creation parameters have Unicode characters, such as accent marks. This includes parameters that can be left blank where a default value is filled in automatically. Ensure the corresponding default values in your Active Directory also do not contain Unicode characters.

If you encounter problems not listed here while using Amazon FSx, ask a question in the Amazon FSx Forum or contact Amazon Web Services Support.

File system is in a misconfigured state

An FSx for Windows File Server file system can get into a **Misconfigured** state due to a change in your Active Directory environment. In this state, your file system is either currently unavailable or at risk of losing availability, and backups may not succeed.

The **Misconfigured** state includes an error message and recommended corrective action that you can access using the Amazon FSx console, API, or AWS CLI. After taking the corrective action, verify that your file system's state eventually changes to `Available` – note that this change can take several minutes to complete.

Your file system can get into a **Misconfigured** state for several reasons, such as the following:

- The DNS Server IP addresses are no longer valid.
- The service account credentials are no longer valid, or lack required permissions.
- The Active Directory domain controller is not reachable due to network connectivity issues, such as invalid VPC Security Groups, VPC Network ACL or routing table configuration, or domain controller firewall settings.

(For the full list of Active Directory requirements, see Prerequisites for using a self-managed Microsoft AD (p. 34). You can also validate that your Active Directory environment is properly configured to meet these requirements by using the Amazon FSx Active Directory Validation tool (p. 40).)

Resolving some of these issues requires directly updating one or more parameters in your file system's Active Directory configuration, such as changing DNS Server IP addresses, or changing the service account username or password. In these cases, your corrective action will necessarily involve using the Amazon FSx console, API, or AWS CLI to update the required configuration parameters.

Other issues may not require changing any Active Directory configuration parameters, such as changing your domain controller firewall settings or VPC Security Groups. In these cases, however, you will need to take further action before the file system can become `Available`. After ensuring your Active Directory environment is configured properly, simply update your file system's Active Directory configuration parameters to their current (unchanged) values by using the he Amazon FSx console, API, or AWS CLI.

Topics

Misconfigured file system: Amazon FSx can't reach either the DNS servers or domain controllers for your domain.

A file system will go into a `Misconfigured` state when Amazon FSx can't communicate with your Microsoft Active Directory domain controller or controllers.

To resolve this situation, do the following:

1. Make sure that your networking configuration allows traffic from the file system to the domain controller.
2. Use the Amazon FSx Active Directory Validation tool (p. 40) to test and verify the network settings for your self-managed Active Directory. For more information, see Using Amazon FSx with your self-managed Microsoft Active Directory (p. 33).
3. Review the file system's self-managed Active Directory configuration in the Amazon FSx console.
4. To update the file system's self-managed Active Directory configuration, you can use the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx CLI `update-file-system` command or the API operation UpdateFileSystem.

Misconfigured file system: The service account credentials are invalid

Amazon FSx can't establish a connection with your Microsoft Active Directory domain controller or controllers. This is because the service account credentials provided are invalid. For more information, see Using Amazon FSx with your self-managed Microsoft Active Directory (p. 33).

To resolve the misconfiguration, do the following:

1. Verify that you are using the correct service account, and you are using the correct credentials for that account.
2. Then update the file system's configuration with the correct service account or account credentials using the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the misconfigured file system to update.

    b. On the **File system details** page, choose **Update** in the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the *Amazon FSx API Reference*.

Misconfigured file system: The service account provided doesn't have permission to join the file system to the domain

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers. This is because the service account provided doesn't have permission to join the file system to the domain with the specified OU.

To resolve the misconfiguration, do the following:

1. Add the required permissions to the Amazon FSx service account, or create a new service account with the required permissions. For more information about doing this, see Delegating privileges to your Amazon FSx service account (p. 38).
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials. To update the configuration, you can use the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the *Amazon FSx API Reference*.

Misconfigured file system: The service account can't join any more computers to domain

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers. In this case, this is because the service account provided has reached the maximum number of computers that it can join to the domain.

To resolve the misconfiguration, do the following:

1. Identify another service account or create a new service account that can join new computers to the domain.
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials using the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the *Amazon FSx API Reference*.

Misconfigured file system: The service account doesn't have access to the OU

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers because the service account provided doesn't have access to the OU specified.

To resolve the misconfiguration, do the following:

1. Identify another service account or create a new service account that has access to the OU.
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the *Amazon FSx API Reference*.

Troubleshooting using Remote Power Shell on FSx for Windows File Server

You can administer your FSx for Windows File Server file systems using custom remote-management PowerShell commands.

Topics

- You can't access your file system using Remote PowerShell (p. 217)

New-FSxSmbShare command fails with one-way trust

Amazon FSx does not support executing the `New-FSxSmbShare` PowerShell command in cases where you have a one-way trust and the domain in which the user resides is not configured to trust the domain associated with Amazon FSx file system.

You can resolve this situation using one of following solutions:

- The user executing the `New-FSxSmbShare` command needs to be in the same domain as the FSx file system.
- You can use the fsmgmt.msc GUI to create shares on your file system. For more information, see Using the GUI to manage file shares (p. 96).

You can't access your file system using Remote PowerShell

There are a number of potential causes for being unable to connect to your file system using Remote PowerShell, each with their own resolution, as follows.

To first ensure that you can connect successfully to the Windows Remote PowerShell Endpoint, you can also run a basic connectivity test. For example, you can run the `test-netconnection endpoint -port 5985` command.

The file system's security group lacks the required inbound rules to allow a remote PowerShell connection

The file system's security group must have an inbound rule that allows traffic on port 5985 in order to establish a Remote PowerShell session. For more information, see Amazon VPC Security Groups (p. 177).

You have an external trust configured between the AWS managed Microsoft Active Directory and your on-premises Active Directory

In order to use the Amazon FSx Remote PowerShell with Kerberos authentication, you need to configure a local group policy on the client for forest search order. For more information, see the Microsoft documentation Configure Kerberos Forest Search Order (KFSO).

A language localization error occurs when trying to initiate a remote PowerShell session

You need to add the following `-SessionOption` to your command: `-SessionOption (New-PSSessionOption -uiCulture "en-US")`

Following are two examples using `-SessionOption` when initiating a remote PowerShell session on your file system.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName Windows Remote PowerShell Endpoint
 -ConfigurationName FSxRemoteAdmin -scriptblock {fsx-command} -SessionOption (New-
PSSessionOption -uiCulture "en-US")
```

```
PS C:\Users\delegateadmin> Enter-Pssession -ComputerName Windows Remote PowerShell Endpoint
 -ConfigurationName FsxRemoteAdmin -SessionOption (New-PSSessionOption -uiCulture "en-US")
```

You can't configure DFS-R on a Multi-AZ or Single-AZ 2 file system

Microsoft Distributed File System Replication (DFS-R) is not supported on Multi-AZ and Single-AZ 2 file systems.

Multi-AZ file systems are configured for redundancy across multiple access zones natively. Use the Multi-AZ deployment type for high availability across multiple Availability Zones. For more information, see Availability and durability: Single-AZ and Multi-AZ file systems (p. 20).

Storage or throughput capacity updates fail

There are a number of potential causes for file system storage and throughput capacity update requests to fail, each with their own resolution.

Storage capacity increase fails because Amazon FSx can't access the file system's KMS encryption key

A storage capacity increase request failed because Amazon FSx was unable to access the file system's AWS Key Management Service (AWS KMS) encryption key.

You need to ensure that Amazon FSx has access to the AWS KMS key in order to run the administrative action. Use the following information to resolve the key access issue.

- If the KMS key has been deleted, you must create a new file system from a backup using a new KMS key. For more information, see Walkthrough 2: Create a file system from a backup (p. 161). You can retry the request after the new file system is available.
- If the KMS key is disabled, re-enable it, and then retry the storage capacity increase request. For more information, see Enabling and disabling keys in the *AWS Key Management Service Developer Guide*.
- If the key is invalid because of its pending deletion, you must create a new file system from a backup using a new KMS key. You can retry the request after the new file system is available. For more information, see Walkthrough 2: Create a file system from a backup (p. 161).
- If the key is invalid because of its pending import, you must wait until the import has completed, and then retry the storage increase request.
- If the key's grant limit has been exceeded, you must request an increase in the number of grants for the key. For more information, see Resource quotas in the *AWS Key Management Service Developer Guide*. When the quota increase is granted, retry the storage increase request.

Storage or throughput capacity update fails because the self-managed Active Directory is misconfigured

The storage capacity or throughput capacity update request failed because your file system's self-managed Active Directory is in a misconfigured state.

To resolve the specific misconfigured state, see File system is in a misconfigured state (p. 213).

Storage capacity increase fails because of insufficient throughput capacity

The storage capacity increase request failed because the file system's throughput capacity is set to 8 MB/s.

Increase the file system's throughput capacity to a minimum of 16 MB/s, then retry the request. For more information, see Managing throughput capacity (p. 133).

Throughput capacity update to 8 MB/s fails

A request to modify a file system's throughput capacity to 8 MB/s failed.

This can occur when a storage capacity increase request is pending or in progress. Storage capacity increases require a minimum throughput of 16 MB/s. Wait until the storage capacity increase request has completed, and then retry the throughput capacity modification request.

Switching storage type to HDD while restoring a backup fails

Creating a file system from a backup fails with the following error message:

```
Switching storage type to HDD while creating a file system from backup
backup_id is not supported because a storage scaling activity was still under
way on the source file system to increase storage capacity from less than 2000
GiB when the backup backup_id was taken, and the minimum storage capacity for
HDD storage is 2000 GiB.
```

This issue occurs when restoring a backup and you have changed the storage type from SSD to HDD. The restore from backup fails because the backup that you are restoring was taken while a storage capacity increase was still in progress on the original file system. The file system's SSD storage capacity before the increase request was less than 2000 GiB, which is the minimum storage capacity required to create an HDD file system.

Use the following procedure to resolve this issue.

1. Wait for the storage capacity increase request to complete and the file system has at least 2000 GiB of SSD storage capacity. For more information, see Monitoring storage capacity increases (p. 127).
2. Take a user-initiated backup of the file system. For more information, see Working with user-initiated backups (p. 79).
3. Restore the user-initiated backup to a new file system using HDD storage. For more information, see Restoring backups (p. 82).

Troubleshooting shadow copies

There are a number of potential causes when shadow copies are missing or inaccessible, as described in the following section.

Topics

Oldest shadow copies are missing

The oldest shadow copies are deleted in either of these situations:

- If you have 500 shadow copies, the next shadow copy replaces the oldest shadow copy, regardless of the remaining allocated storage volume space for shadow copies.
- If the maximum shadow copy storage amount configured is reached, the next shadow copy replaces one or more of the oldest shadow copies, even if you have fewer than 500 shadow copies.

Both results are expected behavior. If you have insufficient storage allocated for shadow copies, consider increasing the storage you have allocated.

All of my shadow copies are missing

During the creation of shadow copies, having insufficient I/O performance capacity on your file system (for example, because you're using HDD storage, because the HDD storage has run out of burst capacity, or because the throughput capacity is insufficient) can cause all shadow copies to be deleted by Windows Server because it is unable to maintain the shadow copies with the available I/O performance capacity. Consider the following recommendations to help prevent this problem:

- If you're using HDD storage, switch to using SSD storage. You can do so by taking a backup of your file system and restoring it with the storage type being switched to SSD.
- Increase the file system's throughput capacity to a value three times your expected workload.
- Make sure that your file system has at least 320 MB of free space, in addition to the maximum shadow copy storage amount configured.
- Schedule shadow copies when you expect your file system to be idle.

For more information, see File system recommendations for shadow copies (p. 85).

Cannot create Amazon FSx backups or access shadow copies on a recently restored or updated file system

This is expected behavior. Amazon FSx rebuilds shadow-copy state on a recently restored file system and does not allow access to shadow copies or backups while rebuilding the shadow copy state.

Troubleshooting data deduplication

There are a number of potential causes for data deduplication issues, as described in the following section.

Topics

- Space is not freed up on file system after deleting files (p. 221)

Data deduplication is not working

Using the instructions in our data deduplication documentation (p. 113), run the `Get-FSxDedupStatus` command to view the completion status for the most recent deduplication jobs. If one or more jobs is failing, you may not see an increase in free storage capacity on your file system.

The most common reason for deduplication jobs failing is insufficient memory.

- Microsoft recommends optimally having 1 GB of memory per 1 TB of logical data (or at a minimum 300 MB + 50 MB per 1 TB of logical data). Use the Amazon FSx performance table (p. 155) to determine the memory associated with your file system's throughput capacity and ensure the memory resources are sufficient for the size of your data.
- Deduplication jobs are configured with the Windows recommended default of 25% memory allocation, which means that for a file system with 32 GB of memory, 8 GB will be available for deduplication. The memory allocation is configurable (using the `Set-FSxDedupSchedule` command with parameter `-Memory`), but consuming additional memory may impact file system performance.
- You can modify the configuration of deduplication jobs to further reduce memory requirements. For example, you can constrain the optimization to run on specific file types or folders, or set a minimum file size and age for optimization. We also recommend configuring deduplication jobs to run during idle periods when there is minimal load on your file system.

You may also see errors if deduplication jobs have insufficient time to complete. You may need to change the maximum duration of jobs, as described in Modifying a data deduplication schedule (p. 115).

If deduplication jobs have been failing for a long period of time, and there have been changes to the data on the file system during this period, subsequent deduplication jobs may require more resources to complete successfully for the first time.

Deduplication values are unexpectedly set to 0

The values for `SavedSpace` and `OptimizedFilesSavingsRate` are unexpectedly 0 for a file system on which you have configured data deduplication.

This can occur during the storage optimization process when you increase the file system's storage capacity. When you increase a file system's storage capacity, Amazon FSx cancels existing data deduplication jobs during the storage optimization process, which migrates data from the old disks to the new, larger disks. Amazon FSx resumes data deduplication on the file system once the storage optimization job completes. For more information about increasing storage capacity and storage optimization, see Managing storage capacity (p. 123).

Space is not freed up on file system after deleting files

The expected behavior of data deduplication is that if the data that was deleted was something that dedup had saved space on, then the space is not actually freed up on your file system until the garbage collection job runs.

A practice you may find helpful is to set the schedule to run the garbage collection job right after you delete a large number of files. After the garbage collection job finishes, you can set the garbage collection schedule back to its original settings. This ensures you can quickly see the space from your deletions immediately.

Use the following procedure to set the garbage collection job to run in 5 minutes.

1. To verify that data deduplication is enabled, use the `Get-FSxDedupStatus` command. For more information on the command and its expected output, see Viewing the amount of saved space (p. 115).
2. Use the following to set the schedule to run the garbage collection job 5 minutes from now.

```
$date=get-date
$DayOfWeek = $date.DayOfWeek
$Hour = $date.Hour
$Minute = $date.Minute + 5
$Time = "${Hour}:${Minute}"
Invoke-Command -ComputerName ${RPS_ENDPOINT} -ConfigurationName FSxRemoteAdmin -
ScriptBlock {
    Set-FSxDedupSchedule -Name "WeeklyGarbageCollection" -Days $Using:DayOfWeek -Start
 $Using:Time -DurationHours 9
}
```

3. After the garbage collection job has run and the space has been freed up, set the schedule back to its original settings.

Additional information

This section provides a reference of supported, but deprecated Amazon FSx features.

Topics

Setting up a custom backup schedule

We recommend using AWS Backup to set up a custom backup schedule for your file system. The information provided here is for reference purposes if you need to schedule backups more frequently than you can when using AWS Backup.

When enabled, Amazon FSx for Windows File Server automatically takes a backup of your file system once a day during a daily backup window. Amazon FSx enforces a retention period that you specify for these automatic backups. It also supports user-initiated backups, so you can make backups at any point.

Following, you can find the resources and configuration to deploy custom backup scheduling. Custom backup scheduling performs user-initiated backups on an Amazon FSx file system on a custom schedule that you define. Examples might be once every six hours, once every week, and so on. This script also configures deleting backups older than your specified retention period.

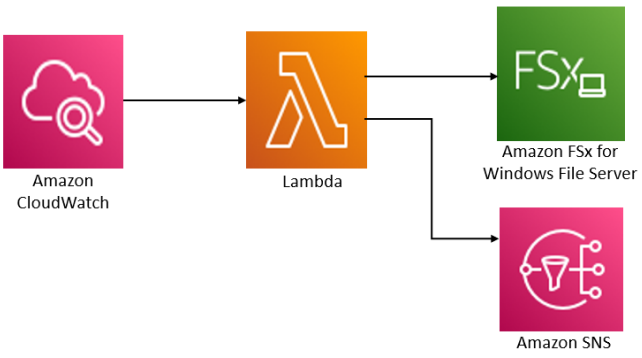
The solution automatically deploys all the components needed, and takes in the following parameters:

- The file system
- A CRON schedule pattern for performing backups
- The backup retention period (in days)
- The backup name tags

For more information on CRON schedule patterns, see Schedule Expressions for Rules in the Amazon CloudWatch User Guide.

Architecture overview

Deploying this solution builds the following resources in the AWS Cloud.



This solution does the following:

1. The AWS CloudFormation template deploys an CloudWatch Event, a Lambda function, an Amazon SNS queue, and an IAM role. The IAM role gives the Lambda function permission to invoke the Amazon FSx API operations.
2. The CloudWatch event runs on a schedule you define as a CRON pattern, during the initial deployment. This event invokes the solution's backup manager Lambda function that invokes the Amazon FSx `CreateBackup` API operation to initiate a backup.
3. The backup manager retrieves a list of existing user-initiated backups for the specified file system using `DescribeBackups`. It then deletes backups older than the retention period, which you specify during the initial deployment.
4. The backup manager sends a notification message to the Amazon SNS queue on a successful backup if you choose the option to be notified during the initial deployment. A notification is always sent in the event of a failure.

AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Amazon FSx custom backup scheduling solution. To use this solution, download the fsx-scheduled-backup.template AWS CloudFormation template.

Automated deployment

The following procedure configures and deploys this custom backup scheduling solution. It takes about five minutes to deploy. Before you start, you must have the ID of an Amazon FSx file system running in an Amazon Virtual Private Cloud (Amazon VPC) in your AWS account. For more information on creating these resources, see Getting started with Amazon FSx (p. 7).

> **Note**
> Implementing this solution incurs billing for the associated AWS services. For more information, see the pricing details pages for those services.

To launch the custom backup solution stack

1. Download the fsx-scheduled-backup.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the *AWS CloudFormation User Guide*.

   > **Note**
   > By default, this template launches in the US East (N. Virginia) AWS Region. Amazon FSx is currently only available in specific AWS Regions. You must launch this solution in an AWS Region where Amazon FSx is available. For more information, see the Amazon FSx section of AWS Regions and Endpoints in the *AWS General Reference*.

2. For **Parameters**, review the parameters for the template and modify them for the needs of your file system. This solution uses the following default values.

Parameter	Default	Description
Amazon FSx file system ID	No default value	The file system ID for the file system that you want to back up.
CRON schedule pattern for backups.	0 0/4 * * ? *	The schedule to run the CloudWatch event, triggering a new backup and deleting

Parameter	Default	Description
		old backups outside of the retention period.
Backup retention (days)	7	The number of days to keep user-initiated backups. The Lambda function deletes user-initiated backups older than this number of days.
Name for backups	user-scheduled backup	The name for these backups, which appears in the **Backup Name** column of the Amazon FSx Management Console.
Backup notifications	Yes	Choose whether to be notified when backups are successfully initiated. A notification is always sent if there's an error.
Email address	No default value	The email address to subscribe to the SNS notifications.

3. Choose **Next**.
4. For **Options**, choose **Next**.
5. For **Review**, review and confirm the settings. You must select the check box acknowledging that the template create IAM resources.
6. Choose **Create** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in about five minutes.

Additional options

You can use the Lambda function created by this solution to perform custom scheduled backups of more than one Amazon FSx file system. The file system ID is passed to the Amazon FSx function in the input JSON for the CloudWatch event. The default JSON passed to the Lambda function is as follows, where the values for `FileSystemId` and `SuccessNotification` are passed from the parameters specified when launching the AWS CloudFormation stack.

```
{
 "start-backup": "true",
 "purge-backups": "true",
 "filesystem-id": "${FileSystemId}",
 "notify_on_success": "${SuccessNotification}"
}
```

To schedule backups for an additional Amazon FSx file system, create another CloudWatch event rule. You do so using the Schedule event source, with the Lambda function created by this solution as the target. Choose **Constant (JSON text)** under **Configure Input**. For the JSON input, simply substitute the file system ID of the Amazon FSx file system to back up in place of `${FileSystemId}`. Also, substitute either `Yes` or `No` in place of `${SuccessNotification}` in the JSON above.

Any additional CloudWatch Event rules you create manually aren't part of the Amazon FSx custom scheduled backup solution AWS CloudFormation stack. Thus, they aren't removed if you delete the stack.

Using Microsoft Distributed File System Replication

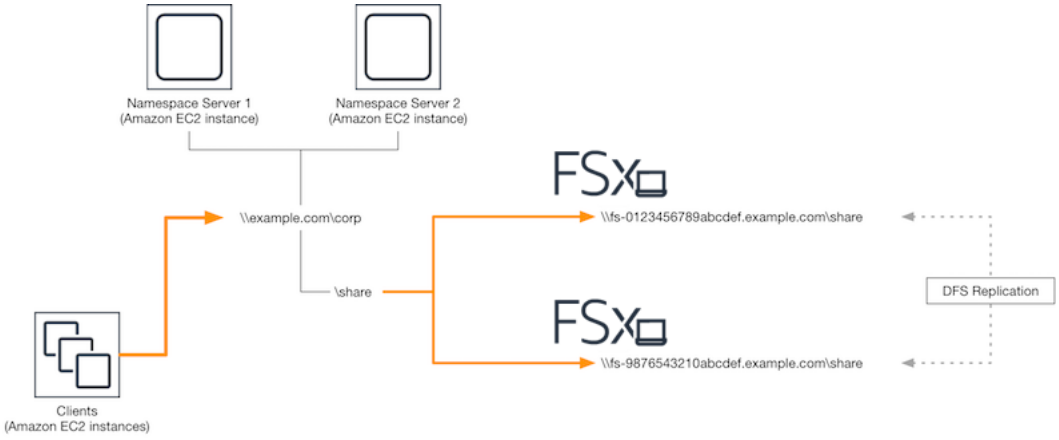
Note
To implement high availability for an FSx for Windows File Server, we recommend using Amazon FSx Multi-AZ. For more information about Amazon FSx Multi-AZ, see Availability and durability: Single-AZ and Multi-AZ file systems (p. 20)

Amazon FSx supports the use of the Microsoft Distributed File System (DFS) for file system deployments across multiple Availability Zones (AZs) to get Multi-AZ availability and durability. Using DFS Replication, you can automatically replicate data between two file systems. Using DFS Namespaces, you can configure one file system as your primary and the other as your standby, with automatic failover to the standby if the primary becomes unresponsive.

Before using DFS Replication, take the following steps:

- Set up your security groups as described in Step 8 (p. 8) of Getting Started with Amazon FSx.
- Create two Amazon FSx file systems in different AZs within an AWS Region. For more information on creating your file systems, see Step 3: Write data to your file share (p. 11).
- Ensure that both file systems are in the same AWS Directory Service for Microsoft Active Directory.
- After the file systems are created, note their file system IDs for later on.

In the following topics, you can find a description of how to set up and use DFS Replication and DFS Namespaces failover across AZs with Amazon FSx.



Setting Up DFS Replication

You can use DFS Replication to automatically replicate data between two Amazon FSx file systems. This replication is bidirectional, meaning that you can write to either file system and the changes are replicated to the other.

Important
You can't use the DFS Management UI in the Microsoft Windows Administrative Tools (dfsmgmt.msc) to configure DFS Replication on your FSx for Windows File Server file system.

To Set Up DFS Replication (Scripted)

1. Begin the process of managing DFS by launching your instance and connecting it to the Microsoft Active Directory where you joined your Amazon FSx file systems. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:

- Seamlessly Join a Windows EC2 Instance
- Manually Join a Windows Instance

2. Connect to your instance as an Active Directory user that is a member of the file system administrators group. In AWS Managed AD, this group is called AWS Delegated FSx Administrators. In your self-managed Microsoft AD, this group is called Domain Admins or the custom name for the administrators group that you provided during creation.

   This user must also be a member of a group that has DFS administration permissions delegated to it. In AWS Managed AD, this group is called AWS Delegated Distributed File System Administrators. In your self-managed AD, this user must be a member of Domain Admins or another group to which you delegated DFS administration permissions.

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Download the FSx-DFSr-Setup.ps1 PowerShell script.
4. Open the **Start** menu and enter **PowerShell**. From the list, choose **Windows PowerShell**.
5. Run the PowerShell script with the following specified parameters to establish DFS Replication between your two file systems:

   - The names of the DFS Replication group and folder
   - The local path to the folder that you want to replicate on your file systems (for example, `D:\share` for the default share that comes included with your Amazon FSx file system)
   - The DNS names of the primary and standby Amazon FSx file systems you created in the prerequisite steps

   Example

```
FSx-DFSr-Setup.ps1 -group Group -folder Folder -path ContentPath -
primary FSxFileSystem1-DNS-Name -standby FSxFileSystem2-DNS-Name
```

To Set Up DFS Replication (Step by Step)

1. Begin the process of managing DFS by launching your instance and connecting it to the Microsoft Active Directory where you joined your Amazon FSx file systems. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:

   - Seamlessly Join a Windows EC2 Instance
   - Manually Join a Windows Instance
2. Connect to your instance as an Active Directory user that is a member of the file system administrators group. In AWS Managed AD, this group is called AWS Delegated FSx Administrators. In your self-managed Microsoft AD, this group is called Domain Admins or the custom name for the administrators group that you provided during creation.

   This user must also be a member of a group that has DFS administration permissions delegated to it. In AWS Managed AD, this group is called AWS Delegated Distributed File System Administrators. In your self-managed AD, this user must be a member of Domain Admins or another group to which you delegated DFS administration permissions.

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Open the **Start** menu and enter **PowerShell**. From the list, choose **Windows PowerShell**.

4. If you don't have DFS Management Tools installed already, install them on your instance with the following command.

```
Install-WindowsFeature RSAT-DFS-Mgmt-Con
```

5. From the PowerShell prompt, create a DFS Replication group and folder with the following commands.

```
$Group = "Name of the DFS Replication group"
$Folder = "Name of the DFS Replication folder"

New-DfsReplicationGroup -GroupName $Group
New-DfsReplicatedFolder -GroupName $Group -FolderName $Folder
```

6. Determine the Active Directory computer name associated with each file system with the following commands.

```
$Primary = "DNS name of the primary FSx file system"
$Standby = "DNS name of the standby FSx file system"

$C1 = (Get-ADObject -Filter "objectClass -eq 'Computer' -and ServicePrincipalName -eq
 'HOST/$Primary'").Name
$C2 = (Get-ADObject -Filter "objectClass -eq 'Computer' -and ServicePrincipalName -eq
 'HOST/$Standby'").Name
```

7. Add your file systems as members of the DFS Replication group that you created with the following commands.

```
Add-DfsrMember -GroupName $Group -ComputerName $C1
Add-DfsrMember -GroupName $Group -ComputerName $C2
```

8. Use the following commands to add the local path (for example, `D:\share`) for each file system to the DFS Replication group. In this procedure, *`file system 1`* serves as the primary member, meaning that its contents initially are synced to the other file system.

```
$ContentPath1 = "Local path to the folder you want to replicate on file system 1"
$ContentPath2 = "Local path to the folder you want to replicate on file system 2"

Set-DfsrMembership -GroupName $Group -FolderName $Folder -ContentPath $ContentPath1 -
ComputerName $C1 -PrimaryMember $True
Set-DfsrMembership -GroupName $Group -FolderName $Folder -ContentPath $ContentPath2 -
ComputerName $C2 -PrimaryMember $False
```

9. Add a connection between the file systems with the following command.

```
Add-DfsrConnection -GroupName $Group -SourceComputerName $C1 -DestinationComputerName
 $C2
```

Within minutes, both file systems should begin synchronizing the contents of the `ContentPath` specified preceding.

Setting Up DFS Namespaces For Failover

You can use DFS Namespaces to treat one file system as your primary, and the other as your standby. By doing this, you can configure automatic failover to the standby if the primary becomes unresponsive. DFS Namespaces enables you to group shared folders on different servers into a single Namespace, where a single folder path can lead to files stored on multiple servers. DFS Namespaces are managed

by DFS Namespace servers, which direct compute instances mapping a DFS Namespace folder to the appropriate file servers.

To Set Up DFS Namespaces for Failover (UI)

1. If you don't already have DFS Namespace servers running, launch a pair of highly available DFS Namespace servers using the setup-DFSN-servers.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the *AWS CloudFormation User Guide*.
2. Connect to one of the DFS Namespace servers launched in the previous step as a user in the AWS Delegated Administrators group. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Open the DFS Management console. Open the **Start** menu and run `dfsmgmt.msc`. Doing this opens the DFS Management GUI tool.
4. For **Action**, choose **New Namespace**, and enter the computer name of the first DFS Namespace server that you launched for **Server** and choose **Next**.
5. For **Name**, enter the namespace you're creating (for example, **`corp`**).
6. Choose **Edit Settings** and set the appropriate permissions based on your requirements. Choose **Next**.
7. Keep the default **Domain-based namespace** option selected, keep the **Enable Windows Server 2008 mode** option selected, and choose **Next**.

   Note
   Windows Server 2008 mode is the latest available option for Namespaces.
8. Review the namespace settings and choose **Create**.
9. With the newly created namespace selected under **Namespaces** in the navigation bar, choose **Action**, then **Add Namespace Server**.
10. For **Namespace server**, enter the computer name of the second DFS Namespace server that you launched.
11. Choose **Edit Settings**, set the appropriate permissions based on your requirements, and choose **OK**.
12. Choose **Add**, enter the UNC name of the file share on the primary Amazon FSx file system (for example *fs-0123456789abcdef0*.example.com*share*) for Path to folder target, and choose **OK**.
13. Choose **Add**, enter the UNC name of the file share on the standby Amazon FSx file system (for example, *fs-fedbca9876543210f*.example.com*share*) for Path to folder target, and choose **OK.**
14. From the **New Folder** window, choose **OK**. The new folder is created with the two folder targets under your namespace.
15. Repeat the last three steps for each file share that you want to add to your namespace.

To Set Up DFS Namespaces for Failover (PowerShell)

1. If you don't already have DFS Namespace servers running, launch a pair of highly available DFS Namespace servers using the setup-DFSN-servers.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the *AWS CloudFormation User Guide*.
2. Connect to one of the DFS Namespace servers launched in the previous step as a user in the **AWS Delegated Administrators** group. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide for Windows Instances*.
3. Open the **Start** menu and enter **PowerShell**. **Windows PowerShell** appears in the list of matches.
4. Open the context (right-click) menu for **Windows PowerShell** and choose **Run as Administrator**.
5. If you don't have DFS Management Tools installed already, install it on your instance with the following command.

```
Install-WindowsFeature RSAT-DFS-Mgmt-Con
```

6. If you don't already have an existing DFS Namespace, you can create one using the following PowerShell commands.

```
$NSS1 = computer name of the 1st DFS Namespace server
$NSS2 = computer name of the 2nd DFS Namespace server

$DNSRoot = fully qualified Active Directory domain name (e.g. mydomain.com)
$Namespace = Namespace name you want to use
$Folder = Folder path you want to use within the Namespace
$FS1FolderTarget = Share path to Folder Target on File System 1
$FS2FolderTarget = Share path to Folder Target on File System 2

$NSS1,$NSS2 | ForEach-Object { Invoke-Command -ComputerName $_ -ScriptBlock { mkdir “C:
\DFS\${using:Namespace}”;
New-SmbShare -Name ${using:Namespace} -Path “C:\DFS\${using:Namespace}” } }

New-DfsnRoot -Path "\\${DNSRoot}\${Namespace}" -TargetPath "\\${NSS1}.${DNSRoot}\
${Namespace}" -Type DomainV2
New-DfsnRootTarget -Path "\\${DNSRoot}\${Namespace}" -TargetPath "\\${NSS2}.${DNSRoot}\
${Namespace}"
```

7. To create a folder within your DFS Namespace, you can use the following PowerShell command. Doing this creates a folder that directs compute instances accessing the folder to your primary Amazon FSx file system by default.

```
$FS1 = DNS name of primary FSx file system
New-DfsnFolder -Path “\\${DNSRoot}\${Namespace}\${Folder}" -TargetPath “\\${FS1}\
${FS1FolderTarget}” -EnableTargetFailback $True -ReferralPriorityClass GlobalHigh
```

8. Add your standby Amazon FSx file system to the same DFS Namespace folder. Compute instances accessing the folder fall back to this file system if they can't connect to the primary Amazon FSx file system.

```
$FS2 = DNS name of secondary FSx file system
New-DfsnFolderTarget -Path “\\${DNSRoot}\${Namespace}\${Folder}" -TargetPath “\\${FS2}\
${FS2FolderTarget}”
```

You can now access your data from compute instances using the DFS Namespace folder's remote path specified preceding. Doing this directs the compute instances to the primary Amazon FSx file system (and to the standby file system, if the primary is unresponsive).

For example, open the **Start** menu and enter `PowerShell`. From the list, choose `Windows PowerShell` and run the following command.

```
net use Z: \\${DNSRoot}\${Namespace}\${Folder} /persistent:yes
```

Working with Maintenance Windows and FSx Multi-AZ

To help ensure high availability of your Multi-AZ file system deployment, we recommend that you pick nonoverlapping maintenance windows for the two Amazon FSx file systems in your Multi-AZ deployment. Doing this helps ensure that your file data continues to be available to your applications and users during system maintenance windows.

Note
To allow DFS Replication traffic to and from the file systems, make sure that you add VPC security group inbound and outbound rules as described in Amazon VPC Security Groups (p. 177).

Document history

- **API version:** 2018-03-01
- **Latest documentation update:** April 5, 2022

The following table describes important changes to the *Amazon FSx Windows User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Support added for AWS PrivateLink interface VPC endpoints. (p. 232)	You can now use interface VPC endpoints to access the Amazon FSx API from your VPC without sending traffic over the internet. For more information, see Amazon FSx and interface VPC endpoints.	April 5, 2022
Support added for Amazon Kendra (p. 232)	You can now use your FSx for Windows File Server file system as a data source for Amazon Kendra, allowing you to index and search for information contained in documents stored on your file system. For more information, see Using FSx for Windows File Server with Amazon Kendra.	March 26, 2022
Support added for file access auditing (p. 232)	You can now enable auditing of end-user accesses on files, folders,and file shares. You can choose to send audit event logs to the Amazon CloudWatch Logs or Amazon Kinesis Data Firehose services. For more information, see File access auditing.	June 8, 2021
Support added for copying backups (p. 232)	You can now use Amazon FSx to copy backups within the same AWS account to another AWS Region (cross-Region copies) or within the same AWS Region (in-Region copies). For more information, see Copying backups.	April 12, 2021
Automatically increase a file system's storage capacity (p. 232)	Use an AWS-developed customizable AWS CloudFormation template to automatically increase your file system's stoage capacity when it's capacity reaches a threshold that you specify. For	February 17, 2021

	more information, see Increasing storage capacity dynamically.	
Support added for client access using non-private IP addresses (p. 232)	You can access FSx for Windows File Server file systems with on-premises clients using non-private IP addresses. For more information, see Supported environments. You can join FSx for Windows File Server file system to a self-managed Microsoft Active Directory with DNS servers and AD domain controllers that use non-private IP addresses. For more information, see Using Amazon FSx with Your Self-Managed Microsoft Active Directory.	December 17, 2020
Support added for using DNS aliases (p. 232)	You can now associate DNS aliases with your FSx for Windows File Server file systems that you can use to access the data on your file system. For more information, see Managing DNS aliases and Walkthrough 5: Using DNS aliases to access your file system.	November 9, 2020
Support added for Amazon Elastic Container Service (p. 232)	You can now use FSx for Windows File Server with Amazon ECS. For more information, see Supported Clients.	November 9, 2020
Amazon FSx is now integrated with AWS Backup (p. 232)	You can now use AWS Backup to back up and restore your FSx file systems in addition to using native Amazon FSx backups. For more information, see Using AWS Backup with Amazon FSx.	November 9, 2020
Support added for throughput capacity scaling (p. 232)	You can now modify the throughput capacity for existing FSx for Windows File Server file systems as your throughput requirements evolve. For more information, see Managing Throughput Capacity.	June 1, 2020
Support added for storage capacity scaling (p. 232)	You can now increase the storage capacity for existing FSx for Windows File Server file systems as your storage requirements evolve. For more information, see Managing Storage Capacity.	June 1, 2020

Support added for hard disk drive (HDD) storage (p. 232)	HDD storage gives you price and performance flexibility when using FSx for Windows File Server. For more information, see Optimizing Costs with Amazon FSx.	March 26, 2020
Support added for file transfer using AWS DataSync (p. 232)	You can now use AWS DataSync to transfer files to and from your FSx for Windows File Server. For more information, see Migrate Files to Amazon FSx for Windows File Server Using AWS DataSync.	February 4, 2020
FSx for Windows File Server releases support for additional Windows file system administration tasks (p. 232)	You can now manage and administer file shares, data deduplication, storage quotas, and encryption in transit for your file shares using the Amazon FSx CLI for remote management on PowerShell. For more information, see Administering File Systems.	November 20, 2019
FSx for Windows File Server releases native Multi-AZ support (p. 232)	You can use Multi-AZ deployment for FSx for Windows File Server to more easily create file systems with high availability that span multiple Availability Zones (AZs). For more information, see Availability and Durability: Single-AZ and Multi-AZ File Systems.	November 20, 2019
FSx for Windows File Server releases support for managing user sessions and open files (p. 232)	You can now use the Shared Folders tool native to Microsoft Windows to manage user sessions and open files on your FSx for Windows File Server file systems. For more information, see Managing User Sessions and Open Files.	October 17, 2019
Amazon FSx releases support for Microsoft Windows shadow copies (p. 232)	You can now configure Windows shadow copies on your FSx for Windows File Server file systems. Shadow copies enable your users to easily undo file changes and compare file versions by restoring files to previous versions. For more information, see Working with Shadow Copies.	July 31, 2019

Amazon FSx releases shared Microsoft Active Directory support (p. 232)	You can now join FSx for Windows File Server file systems to AWS Managed Microsoft AD directories that are in a different VPC or in a different AWS account than the file system. For more information, see Active Directory Support.	June 25, 2019
Amazon FSx releases enhanced Microsoft Active Directory support (p. 232)	You can now join FSx for Windows File Server file systems to your self-managed Microsoft Active Directory domains, either on-premises or in the cloud. For more information, see Active Directory Support.	June 24, 2019
Amazon FSx complies with SOC certification (p. 232)	Amazon FSx has been assessed to comply with SOC certification. For more information, see Security and Data Protection.	May 16, 2019
Added clarifying note regarding AWS Direct Connect, VPN, and inter-region VPC peering connection support (p. 232)	Amazon FSx file systems created after February 22, 2019 are accessible using AWS Direct Connect, VPN, and inter-region VPC peering. For more information, see Supported Access Methods.	February 25, 2019
AWS Direct Connect, VPN, and inter-region VPC peering connection support added (p. 232)	You can now access Amazon FSx for Windows File Server file systems from on-premises resources and from resources in a different Amazon VPC or AWS account. For more information, see Supported Access Methods.	February 22, 2019
Amazon FSx is now generally available (p. 232)	Amazon FSx for Windows File Server provides Microsoft Windows file servers that are fully managed, backed by a fully native Windows file system. Amazon FSx for Windows File Server provides the features, performance, and compatibility to easily lift and shift enterprise applications to AWS.	November 28, 2018