# Table of Contents

What Is Amazon FSx for Windows File Server? ......................................................... 1  
Amazon FSx for Windows File Server Resources: File Systems, Backups, and File Shares ........................................ 1  
Accessing File Shares ......................................................................................... 2  
Security and Data Protection ............................................................................... 2  
Availability and Durability ................................................................................... 2  
Managing File Systems ......................................................................................... 2  
Price and Performance Flexibility ....................................................................... 2  
Pricing for Amazon FSx ....................................................................................... 3  
Assumptions ........................................................................................................ 3  
Prerequisites ........................................................................................................ 3  
Amazon FSx for Windows File Server Forums .................................................. 3  
Are You a First-Time User of Amazon FSx? ....................................................... 4  
Setting Up ............................................................................................................ 5  
Sign Up for AWS ................................................................................................. 5  
Create an IAM User ............................................................................................ 5  
Next Step ............................................................................................................. 6  
Getting Started ................................................................................................... 7  
Step 1: Create Your File System ....................................................................... 7  
Step 2: Map Your File Share to an EC2 Instance Running Windows Server ...... 10  
Step 3: Write Data to Your File Share ............................................................... 10  
Step 4: Back Up Your File System .................................................................. 11  
Step 5: Transfer Files Using DataSync ............................................................. 11  
Before You Begin .............................................................................................. 11  
Basic Steps for Transfer .................................................................................... 12  
Step 6: Clean Up Resources ............................................................................. 12  
Amazon FSx File System Status ....................................................................... 13  
Supported Clients, Access Methods, and Environments .................................... 14  
Supported Clients ............................................................................................ 14  
Supported Access Methods ............................................................................ 14  
Accessing Amazon FSx File Systems Using DNS Names ................................ 15  
Working with Amazon FSx for Windows File Server File Systems and DFS Namespaces ............................................. 15  
Supported Environments ................................................................................... 15  
Access File Systems from On-premises ........................................................... 16  
Accessing Amazon FSx for Windows File Server File Systems from Another VPC, Account, or AWS Region ................ 16  
Availability and Durability ................................................................................ 17  
Choosing Single-AZ or Multi-AZ File System Deployment ............................. 17  
Feature Support by Deployment Types ............................................................ 17  
Failover Process for Amazon FSx for Windows File Server ............................ 18  
Failover Experience on Windows Clients ......................................................... 18  
Failover Experience on Linux Clients ............................................................... 18  
Working with Single and Multi-AZ File System Resources ............................ 18  
Subnets ............................................................................................................. 18  
File System Elastic Network Interfaces ............................................................ 19  
Optimizing Costs with Amazon FSx ............................................................... 20  
Storage Type Options ....................................................................................... 20  
Flexibility to Choose Storage and Throughput Independently ....................... 20  
Optimizing Storage Costs Using Data Deduplication ..................................... 20  
Working with Active Directory ....................................................................... 22  
Using AWS Managed Microsoft AD ................................................................. 22  
Test Your Active Directory Configuration ...................................................... 23  
Using AWS Managed Microsoft AD in Different VPC or Account .................. 23  
Using a Self-Managed AD ................................................................................. 24  
Self-Managed AD Prerequisites ....................................................................... 25
What Is Amazon 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. Amazon 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 submillisecond 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, Amazon 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.

Amazon 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 a Windows file server and storage volumes, and is accessed with its DNS name. When you create a file system, you specify an amount of storage capacity (in GiB) and an amount of throughput capacity (in MBps).

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. You can create and manage as many other Windows file shares as you want using the Shared Folders graphical user interface (GUI) tool on Windows. For more information, see Using Microsoft Windows File Shares (p. 32).

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 Amazon 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. 14).

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. 100).

Availability and Durability

Amazon 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. 17).

Managing File Systems

You can administer your Amazon 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. 56).

Price and Performance Flexibility

Amazon 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 Amazon FSx for Windows File Server, you can provision file system storage and throughput independently to achieve the right mix of cost and performance. For more information, see Optimizing Costs with Amazon FSx (p. 20).
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, Amazon 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 Amazon WorkSpaces, we assume that you're familiar with Amazon WorkSpaces. For more information on how to use Amazon 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 Active Directory in Amazon FSx for Windows File Server (p. 22).
• 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. 100).
• Install and configure the AWS Command Line Interface (AWS CLI). Supported versions are 1.9.12 and newer. For more information, see Installing the AWS Command Line Interface 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 on performance, see Amazon FSx for Windows File Server Performance (p. 84).
3. For Amazon FSx security details, see Security in Amazon FSx (p. 100).
4. For information on 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

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. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.

   Note
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below 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 user.
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 (p. 7)
- Step 2: Map Your File Share to an EC2 Instance Running Windows Server (p. 10)
- Step 3: Write Data to Your File Share (p. 10)
- Step 4: Back Up Your File System (p. 11)
- Step 5: Transfer Files to or from Amazon FSx for Windows File Server Using AWS DataSync (p. 11)
- Step 6: Clean Up Resources (p. 12)
- Amazon FSx File System Status (p. 13)

**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. 88).

**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 **Amazon 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. 17).

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**.

Amazon 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 **Storage Type Options** (p. 20).

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.

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. You can't change this value after the file system is created. If you need more throughput capacity, choose **Specify throughput capacity**, and then choose a value.

   For more information about throughput capacity and file system performance, see **Amazon FSx for Windows File Server Performance** (p. 84).

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 you're using for this getting started exercise:

    - Inbound and outbound rules to allow the following ports:
      - TCP/UDP 445 (SMB)
Step 1: Create Your File System

- TCP 135 (RPC)
- TCP/UDP 1024-65535 (Ephemeral ports for RPC)

From and to IP addresses or security group IDs associated with the following source and destination resources:
- Client compute instances from which you want to access the file system.
- Other file servers that you expect this file system to participate with in DFS Replication groups.
- Outbound rules to allow all traffic to the security group ID associated with the AWS Managed Microsoft AD directory that you’re joining your file system to.

**Note**
In some cases, you might have modified the rules of the security group for your AWS Managed Microsoft AD 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. To learn more 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. 103).

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 Active Directory in Amazon FSx for Windows File Server (p. 22).

   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.
   - **DNS server IP addresses**—the IPv4 addresses of the DNS servers for your domain
   - **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 you 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. Keep the default settings for **Maintenance preferences**.

15. 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 and choose **Next**.

16. 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**.

17. After the file system has been created, choose the file system ID in the **File Systems** dashboard, then 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 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 connect 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. Once 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. 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 this:
     \fs-0123456789abcdef0.ad-domain.com
   • For a Single-AZ file system joined to a self-managed AD, and any Multi-AZ file system, the DNS name looks like this:
     amznfsxaal1bb22.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 Amazon 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 Amazon 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 Amazon 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 Amazon 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 Amazon 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 right-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. 88)**, 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**.

<table>
<thead>
<tr>
<th>File System Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABLE</td>
<td>The file system is in a healthy state, and is reachable and available for use.</td>
</tr>
<tr>
<td>CREATING</td>
<td>Amazon FSx is creating a new file system.</td>
</tr>
<tr>
<td>DELETING</td>
<td>Amazon FSx is deleting an existing file system.</td>
</tr>
<tr>
<td>UPDATING</td>
<td>The file system is undergoing a customer-initiated update.</td>
</tr>
<tr>
<td>MISCONFIGURED</td>
<td>The file system is in a failed but recoverable state.</td>
</tr>
</tbody>
</table>
| FAILED             | 1. The file system has failed and Amazon FSx can’t recover it.  
                        2. When creating new file system, Amazon FSx was unable to create a new file system. |
Supported Clients, Access Methods, and Environments for Amazon FSx for Windows File Server

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

Topics
- Supported Clients (p. 14)
- Supported Access Methods (p. 14)
- Supported Environments (p. 15)

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.
- Amazon WorkSpaces instances – To learn more, see the AWS blog post Using Amazon 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 Amazon FSx for Windows File Server in a VMware Cloud on AWS Environment.

The following operating systems are supported for use with Amazon FSx:
- Windows Vista, Windows 7, Windows 8, Windows 8.1, and Windows 10 (including the Windows 7 and Windows 10 desktop experiences of Amazon WorkSpaces)
- Linux, using the cifs-utils tool

Supported Access Methods

You can use the following access methods and approaches with Amazon FSx.
Accessing Amazon FSx File Systems Using DNS Names

Amazon FSx for Windows File Server provides a Domain Name System (DNS) name for every file system. You access your Amazon 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. 32).

You can find the DNS name in the Amazon FSx Management Console, the File systems > Details > Network & Security section, or in the response of the CreateFileSystem or DescribeFileSystems API command.

- For a Single AZ file system joined to an AWS Managed Microsoft Active Directory, the DNS name looks as follows.
  
  fs-0123456789abcdef0.ad-domain.com

- For a Single AZ file system joined to a self-managed AD, and any Multi AZ file system, the DNS name looks as follows.

  amznfsxaal1bb22.ad-domain.com

Important
To get 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.

Working with Amazon FSx for Windows File Server File Systems and DFS Namespaces

Amazon 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.

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. 88).

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

Important
In some cases, you might want to access a file system created before February 22, 2019, from on-premises resources or from resources in a different VPC, AWS account, or AWS Region. To do this, create a new file system from a backup of your existing file system. To learn more about creating and restoring backups, see Working with Backups (p. 50).

Note
Amazon FSx can support access from resources outside the VPC associated with your file system. It can do this if those resources have an IP address in the following private IP version 4 (IPv4) address ranges, as specified in RFC 1918:
Access File Systems from On-premises

- 10.0.0.0–10.255.255.255 (10/8 prefix)
- 172.16.0.0–172.31.255.255 (172.16/12 prefix)
- 192.168.0.0–192.168.255.255 (192.168/16 prefix)

If those resources have a non-private IP address, you can only access file systems from within the same VPC that the resource resides in.

Accessing Amazon FSx for Windows File Server File Systems from On-Premises

Amazon 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, Amazon 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, Amazon 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. 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 a VPN connection, see VPN Connections in the Amazon VPC User Guide.

Accessing Amazon FSx for Windows File Server File Systems from Another VPC, Account, or AWS Region

You can access your Amazon 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 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 Amazon FSx for Windows File Server file system deployment types:

<table>
<thead>
<tr>
<th>Deployment type</th>
<th>SSD storage</th>
<th>HDD storage</th>
<th>DFS namespaces</th>
<th>DFS replication</th>
<th>Custom DNS name</th>
<th>CA shares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-AZ 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Single-AZ 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓*</td>
</tr>
<tr>
<td>Multi-AZ</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓*</td>
</tr>
</tbody>
</table>
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 Amazon 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.

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.

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 reduce cross-AZ data transfer costs and 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 Amazon 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 Amazon FSx for Windows File Server file system deployment types:

<table>
<thead>
<tr>
<th>File system deployment type</th>
<th>Number of subnets</th>
<th>Number of elastic network interfaces</th>
<th>Number of IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-AZ 2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Single-AZ 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Multi-AZ</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Optimizing Costs with Amazon FSx

Amazon 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
- Storage Type Options (p. 20)
- Flexibility to Choose Storage and Throughput Independently (p. 20)
- Optimizing Storage Costs Using Data Deduplication (p. 20)

Storage Type Options

Amazon 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. 84) and Amazon FSx for Windows File Server Pricing.

Flexibility to Choose Storage and Throughput Independently

With Amazon FSx for Windows File Server, you can configure your file system's storage and throughput capacity 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, and you can take advantage of fast caching on the file server to optimize performance for actively accessed data. For more information, see Amazon FSx for Windows File Server Performance (p. 84).

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. 63).
Working with Active Directory in Amazon 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 Amazon FSx for Windows File Server file system with Active Directory: Using Amazon FSx with AWS Directory Service for Microsoft Active Directory (p. 22) and Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24).

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't change that configuration. However, you can create a new file system from a backup and change the Active Directory integration configuration for that file system. This way, you can also change other settings like the maintenance window. For more information, see Walkthrough 2: Create a File System from a Backup (p. 92).

Topics
- Using Amazon FSx with AWS Directory Service for Microsoft Active Directory (p. 22)
- Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24)
- Validating Your Active Directory Configuration (p. 30)

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 Amazon 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).
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 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. 24).

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. Or you can choose to 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.

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 your Active Directory configuration using the Amazon FSx Network Validation tool. For more information, see Validating Your Active Directory Configuration (p. 30).

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

- 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. 88)

**Using Amazon FSx with AWS Managed Microsoft AD in a Different VPC or Account**

You can join your Amazon FSx 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.

To join your file system to an AWS Managed Microsoft AD directory that's in a different VPC, make sure that the two VPCs have network connectivity. To do this, you can use AWS Transit Gateway or Amazon VPC peering. In addition, make sure that network traffic is allowed between them.

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 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.

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 Amazon 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.

- IP addresses of the DNS servers for your domain

- 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. 28).

When you join your file system directly to your self-managed AD, your Amazon 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 (p. 25)
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 that are either in the same VPC CIDR range as the one in which your Amazon FSx file system is being created or in the following private IP address ranges, as specified in RFC 1918:
    - 10.0.0.0–10.255.255.255 (10/8 prefix)
    - 172.16.0.0–172.31.255.255 (172.16/12 prefix)
    - 192.168.0.0–192.168.255.255 (192.168/16 prefix)
  - Domain name that is not in the Single Label Domain (SLD) format. Amazon FSx does not support SLD domains.
  - 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 VPC based on 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.
  - VPC Security Groups that you’ve associated with your Amazon FSx file system, along with any VPC Network ACLs, configured to allow outbound network traffic on the following ports:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>53, 88, 123, 389, 445, 464</td>
</tr>
<tr>
<td>TCP</td>
<td>53, 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535</td>
</tr>
</tbody>
</table>

**Note**
Allowing outbound traffic on TCP port 9839 is required for Multi-AZ file system deployments.

- 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.

Windows Firewall on your Active Directory domain controllers configured to allow inbound traffic on the above mentioned ports from the subnet(s) where you'd like to have your Amazon FSx file system.

In the case where the domain controllers are in AWS, the VPC Security Groups that you’ve associated with them, along with any VPC Network ACLs, configured to allow inbound traffic on the above mentioned ports.

Use the Amazon FSx Network Validation tool (p. 30) 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

To learn more about creating a service account with the correct permissions, see Delegating Privileges to Your Amazon FSx Service Account (p. 26).

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. 27).

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

Best Practices for Joining Amazon 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 Restrictions**
   - **Validated write to DNS host name**
   - **Validated write to service principal name**

11. Choose **Next**, and then choose **Finish**.


**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.

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 action beside the file system description in the console, API, and CLI. If an issue happens, take the recommended corrective action to provide the correct configuration properties. If the issue is resolved, verify that your file system's state changes to **Updating** and eventually to **Available**. To learn more about troubleshooting possible self-managed AD misconfigurations, see File System Is In a Misconfigured State (p. 119).

**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. 103).
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. 103).

Joining an Amazon FSx File System to a Self-Managed Microsoft Active Directory Domain

When you create a new Amazon 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.

Before You Begin

Make sure that you have completed the Prerequisites for Using a Self-Managed Microsoft AD (p. 25) detailed in Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24).

To Create an Amazon 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. Provide a name for your file system. You can use a maximum of 256 Unicode letters, white space, and numbers, plus the special characters + - = . _ : /
4. Enter the Storage capacity for your file system, in GiB. This value can be any whole number in the range of 300 to 65,536.
5. Choose your throughput capacity. The Recommended throughput capacity value is based on your chosen storage capacity. You can change this level to more megabytes per second as needed at creation. To do this, choose a throughput capacity from the box.
6. 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.

7. Choose any value for **Availability Zones** and **Subnet**.

8. 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:

   - Inbound and outbound rules to allow the following ports:
     - TCP/UDP 445 (SMB)
     - TCP 135 (RPC)
     - TCP/UDP 1024-65535 (Ephemeral ports for RPC)

   From and to IP addresses or security group IDs associated with the following source and destination resources:
   - Client compute instances from which you want to access the file system.
   - Other file servers that you expect this file system to participate with in DFS Replication groups.
   - 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.

9. For **Windows authentication**, choose **Self-managed Microsoft Active Directory**.

10. 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.

11. (Optional) Enter a value for **Organizational Unit** for the self-managed Microsoft AD directory.

12. Enter at least one, and no more that two, values for **DNS Server IP Addresses** for the self-managed Microsoft AD directory.

13. 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`).

14. 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.

15. Re-enter the password to confirm it in **Confirm password**.

16. (Optional) Specify a value for **Delegated file system administrators group** in your domain. This group has the delegated authority to perform administrative tasks on your file system. If none is provided, Amazon FSx attempts to use 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 Amazon FSx for Windows File Server File System Joined to a Self-managed AD (AWS CLI)

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

```bash
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",FileSystemAdministratorsGroup="FSxAdmins", 
  UserName="FSxService",Password="password", 
  DnsIps=["10.0.1.18"]}',ThroughputCapacity=8
```

Validating Your Active Directory Configuration

Before you create an Amazon FSx for Windows File Server file system joined to your Active Directory, we recommend that you validate the network connectivity to your DNS servers using the Amazon FSx Network Validation tool. To run the tool, you’ll need the IPv4 addresses for your DNS servers.

To validate network connectivity to your DNS servers

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 Amazon FSx for Windows File Server file system.
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 on the EC2 instance.
5. Download the network validation tool using the following command.
   ```bash
   PS C:\> Invoke-WebRequest "https://docs.aws.amazon.com/fsx/latest/WindowsGuide/samples/AmazonFSxNetworkValidation.zip" -OutFile "AmazonFSxNetworkValidation.zip"
   ```
6. Expand the zip file by using the following command.
   ```bash
   PS C:\> Expand-Archive -Path "AmazonFSxNetworkValidation.zip"
   ```
7. Add the AmazonFSxNetworkValidation module to the current session.
   ```bash
   PS C:\> Import-Module .\AmazonFSxNetworkValidation
   ```
8. Validate the network connectivity to your AD domain controller by running the following command. Replace `IP_ADDRESS_1` and `IP_ADDRESS_2` with the IPv4 addresses of the DNS servers for your AD domain that you will use when creating your Amazon FSx file system.
The following is an example of a successful test result.

```powershell
PS C:\> @("IP_ADDRESS_1", "IP_ADDRESS_2") | Test-FSxADControllerConnection

The following is an example of a successful test result.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcpDetails</td>
<td>@{Port=53; Result=Listening; Description=Domain Name System (DNS)}, @{Port=88; Result=Listening}</td>
</tr>
<tr>
<td>ADControllerIp</td>
<td>10.0.75.243</td>
</tr>
<tr>
<td>UdpDetails</td>
<td>@{Port=53; Result=Timed Out; Description=Domain Name System (DNS)}, @{Port=88; Result=Timed Out}</td>
</tr>
<tr>
<td>Success</td>
<td>True</td>
</tr>
</tbody>
</table>

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

```powershell
PS C:\> $Result

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcpDetails</td>
<td>@{Port=53; Result=Listening; Description=Domain Name System (DNS)}, @{Port=88; Result=Listening}</td>
</tr>
<tr>
<td>ADControllerIp</td>
<td>10.0.75.243</td>
</tr>
<tr>
<td>UdpDetails</td>
<td>@{Port=53; Result=Timed Out; Description=Domain Name System (DNS)}, @{Port=88; Result=Timed Out}</td>
</tr>
<tr>
<td>Success</td>
<td>False</td>
</tr>
<tr>
<td>FailedTcpPorts</td>
<td>[9389]</td>
</tr>
</tbody>
</table>
Using Microsoft Windows File Shares

A Microsoft Windows file share is a specific folder in your file system, including 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 called 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, you can find procedures for mapping a file share on the different supported compute instances.

Topics
- Mapping a File Share on an Amazon EC2 Windows Instance (p. 32)
- Mounting a File Share on an Amazon EC2 Linux Instance (p. 34)
- Automatically Mount File Shares on an Amazon Linux EC2 Instance Not Joined to Your Active Directory (p. 37)

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 the file system DNS name and the share name. 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 this:
     
     ```
     fs-0123456789abcdef0.ad-domain.com
     ```

   - For a Single AZ file system joined to a self-managed AD, and any Multi AZ file system, the DNS name looks like this:
     
     ```
     amznfsxaa11bb22.ad-domain.com
     ```

   For example, enter:

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

   for **Folder**.

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

   - For a Single AZ file system joined to a self-managed AD, and any Multi AZ file system, the DNS name looks like this:
     
     ```
     amznfsxaa11bb22.ad-domain.com
     ```

   Here's an example command to mount the file share.
Mounting a File Share on an Amazon EC2 Linux Instance

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

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=3.0,sec=krb5,cruid=ad_user,rsize=CIFSMaxBufSize,wsize=CIFSMax BufSize,cache=none,ip=preferred-file-server-Ip
   ```

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 this:

  ```
  fs-0123456789abcdef0.ad-domain.com
  ```
- For a Single AZ file system joined to a self-managed AD, and any Multi AZ file system, the DNS name looks like this:

  ```
  amznfsxaalbb22.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 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=3.0,sec=krb5,cache=strict,username=user1@CORP.NETWORK.COM,uid=0,noforceuid,gid=0,noforcegid,addr=192.0.2.0,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,wsize=1048576,echo_interval=60,actimeo=1)
```

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=3.0` – Specifies the type of file system as CIFS and the protocol version as 3.0.
- `sec=krb5` – Specifies to use Kerberos version 5 for authentication.
- `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 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 the maximum buffer size is 130048.

- `cache=none` – Sets the CIFS cache mode to none, that is to not cache file data at all.
- `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 management console, in 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 Amazon FSx for Windows File Server file share by using it's private
IP address. You can get the file system's private IP address in the Amazon FSx console (https://console.aws.amazon.com/fsx/), in the Network & security tab, the 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 Amazon 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 https://console.aws.amazon.com/ec2/. 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 shared above.
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 (https://console.aws.amazon.com/fsx/), in the Network & security tab, 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=3.0,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 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
```

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=3.0` – Specifies the type of file system as CIFS and the protocol version as 3.0.
- `sec=ntlmsspi` – Specifies to use NT LAN Manager Security Support Provider Interface (NTLMSSPI) for authentication.
- `cred=/home/ec2-user/creds.txt` – Specifies where to get the user credentials.
- `/mnt/fsx` – Specifies the mount point for the 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 the maximum buffer size is 130048.
- `cache=none` – Sets the CIFS cache mode to none, that is not to cache file data at all.

Automatically Mount File Shares on an Amazon Linux EC2 Instance Not Joined to Your Active Directory

To automatically mount your Amazon FSx for Windows File Server file share whenever the Amazon EC2 Linux instance to which it's mounted reboots, you 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 Amazon FSx for Windows File Server file share by using its private IP address. You can get the file system's private IP address in the Amazon FSx console (https://console.aws.amazon.com/fsx/), in the Network & security tab, the Preferred File Server IP Address.

The following procedure use Microsoft NTLM authentication. You mount the file system as a user that is a member of the Microsoft Active Directory domain to which the Amazon 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 https://console.aws.amazon.com/ec2/. 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.

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

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

```bash
$ 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.

```bash
$ 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 (https://console.aws.amazon.com/fsx/), 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.

```bash
//file-system-IP-address/file_share /mnt/fsx cifs vers=3.0,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.

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

The output shows 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.

```bash
$ 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.
Automatically Mount File Shares on an Amazon Linux EC2 Instance Not Joined to Your Active Directory

```
$ sudo mount -l -t cifs
//file-system-IP-address/file_share on /mnt/fsx type cifs
(rw,relatime,vers=3.0,sec=ntlmsspi,cache=strict,username=user1,domain=CORP.EXAMPLE.COM,uid=0,noforceuid,gid=0,noforcegid,a ...
```

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 FSx file system on your EC2 instance.
- `cifs vers=3.0` – Specifies the type of file system as CIFS and the protocol version as 3.0.
- `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

You can migrate files and file share configurations from your on-premises file systems to your Amazon FSx for Windows File Server file systems, as described following.

Topics
- Migrating Existing File Storage to Amazon FSx for Windows File Server (p. 40)
- Migrating File Share Configurations to Amazon FSx (p. 44)

Migrating Existing File Storage to Amazon FSx for Windows File Server

If you’d like to migrate your existing files to Amazon 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 Amazon FSx for Windows File Server Using AWS DataSync (p. 41).

Note
While DataSync does support copying NTFS access control lists (ACLs), it does not currently support copying file audit control information, also known as NTFS System Access Control Lists (SACLs). SACLs are used by administrators to control audit logging of user attempts to access files. For more information, see How DataSync Handles Metadata and Special Files in the AWS DataSync User Guide. If you need to copy SACLs into your Amazon FSx file system, we recommend that you use Robocopy. For more information, see Migrating Existing Files to Amazon FSx for Windows File Server Using Robocopy (p. 42).

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 Amazon FSx for Windows File Server, see Migrating Existing Files to Amazon FSx for Windows File Server Using Robocopy (p. 42).

Best Practices for Migrating Existing File Storage to Amazon FSx for Windows File Server

To migrate large amounts of data to Amazon 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:

- Take a backup of your SSD file system. For more information, see Creating Manual Backups (p. 51).
• Restore the backup to a file system using HDD storage. For more information, see Restoring Backups (p. 51).

Migrating Existing Files to Amazon FSx for Windows File Server Using AWS DataSync

We recommend using AWS DataSync to transfer data between Amazon 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, time stamps, and access permissions.

Note
While DataSync does support copying NTFS access control lists (ACLs), it does not currently support 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. If you need to copy SACLs into your Amazon FSx file system, we recommend that you use Robocopy. For more information, see Migrating Existing Files to Amazon FSx for Windows File Server Using Robocopy (p. 42).

You can also use DataSync to transfer files between two Amazon 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 Amazon 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 automate replication for data protection and recovery.

In AWS DataSync, a location for Amazon FSx for Windows is an endpoint for an Amazon 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 Amazon 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 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 Amazon 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 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 Amazon 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 Amazon FSx for Windows File Server, see Deploying the DataSync Agent as an Amazon EC2 Instance in the AWS DataSync User Guide.

**Migrating Existing Files to Amazon 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, time stamps, 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 AWS 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.

<table>
<thead>
<tr>
<th>Migration user access model</th>
<th>Source file system accessibility requirements</th>
<th>Destination FSx file server accessibility requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct read/write permissions model</td>
<td>The user needs to have at least read permissions (NTFS ACLs) on the files and folders being migrated.</td>
<td>The user needs to have at least write permissions (NTFS ACLs) on the files and folders being migrated.</td>
</tr>
<tr>
<td>Backup/restore privilege model to override access permissions</td>
<td>The user needs to be a member of the on-premises AD’s Backup</td>
<td>The user needs to be a member of the FSx file system’s</td>
</tr>
</tbody>
</table>
Migrating Files Using Robocopy

<table>
<thead>
<tr>
<th>Migration user access model</th>
<th>Source file system accessibility requirements</th>
<th>Destination FSx file server accessibility requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operators group, and use the /b flag with RoboCopy</td>
<td>Administrators group*, and use the /b flag with RoboCopy</td>
<td></td>
</tr>
<tr>
<td>Domain administrator (full) privilege model to override access permissions</td>
<td>The user needs to be a member of the on-premises AD's Domain Admins group.</td>
<td>The user needs to be a member of the FSx file system's administrators group*, and use the /b flag with RoboCopy</td>
</tr>
</tbody>
</table>

Note

* For file systems joined to an AWS Managed Microsoft AD, the FSx file system administrators group is **AWS Delegated FSx Administrators**. In your self-managed Microsoft AD, the 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 AD's Domain Administrators group.
   ```
   C:\>net use Y: \fileserver1.mydata.com\localdata /user:mydata.com\Administrator
   Enter the password for ‘fileserver1.mydata.com\Administrator’: _
   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 AD'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. 42) in the Prerequisites (p. 42).

```
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.

```
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 – time stamps
  - 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 wish to migrate to Amazon FSx.
To migrate existing file shares to Amazon 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.

   ```powershell
   $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 Amazon 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.

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

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

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

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

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

   ```powershell
   $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.

   ```powershell
   $params = @{}
   ForEach ($item in $shares) {
       $i = $shares.IndexOf($item);
       $params[$i] = @();
       foreach ($property in $item.psObject.properties) {
           if ($property.Name -in $FSxAcceptedParameters) {
               $params[$i][$property.Name] = $property.Value
           }
       }
   }
   $param=$params[$i];
   Invoke-Command -ConfigurationName FSxRemoteAdmin -ComputerName amznfsxxxxxxx.corp.com -ErrorVariable errmsg -ScriptBlock { New-FSxSmbShare -Credential $Using:credential @Using:param }
Using Amazon 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 Amazon 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.

For information about configuring and using Amazon FSx to reduce the complexity and costs for your SQL Server high availability deployments, see Simplify your Microsoft SQL Server high availability deployments using Amazon FSx for Windows File Server on the AWS Storage Blog.

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, you should use CA shares on Multi-AZ file systems for SQL Server HA deployments.

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. 59).

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.
Protecting Your Data with Shadow Copies and Backups

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

- Windows shadow copies enable your users to easily undo file changes and compare file versions by restoring files to previous versions.
- Backups support your backup retention and compliance needs.

Topics
- Working with Shadow Copies (p. 47)
- Working with Backups (p. 50)

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 its 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. Thus, when you restore a backup, users continue to have previous versions available as of the time that the file system was backed up.

Note
Shadow copies are not enabled on Amazon FSx for Windows File Server by default. To have shadow copies running on your file system, you need to 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. 48).

Topics
- Shadow Copies Configuration Overview (p. 47)
- Setting Up Shadow Copies Using Default Settings (p. 48)
- Restoring Individual Files and Folders (p. 49)

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 512 shadow copies per file system at any point in time. When you reach this limit, or when the maximum shadow copy storage amount configured is reached, the next shadow copy that you take replaces the oldest 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. 48). For information about how to customize your shadow copy configuration, see Shadow Copies (p. 66).

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. 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.
To learn about additional options and creating a custom shadow copy schedule, see Creating a Custom Shadow Copy Schedule (p. 68).

### 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 Amazon FSx for Windows File Server shares, see Shadow Copies (p. 66).

Working with Backups

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, restoring a file system from a backup, or deleting a backup.

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, which means that only the changes after your most recent backup are saved. Thus, you can save on backup storage costs by not duplicating data. Using Amazon FSx, you can create a new file system from a backup, effectively restoring a point-in-time snapshot of the file system.

Topics

- Working with Automatic Daily Backups (p. 50)
- Working with Manual Backups (p. 51)
- Restoring Backups (p. 51)
- Deleting Backups (p. 52)
- Setting Up a Custom Backup Schedule (p. 52)

Working with Automatic Daily Backups

By default, Amazon FSx automatically takes backups of your file systems once a day. These daily backups are taken 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 under 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. By default, backups are retained for 7 days. However, you can change the retention period to anywhere in a range of 0–35 days. Automatic daily backups are deleted when the file system is deleted.

You can turn off automatic daily backup by setting the retention period to 0 (zero) days.

**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 with the `UpdateFileSystem` operation. For more information, see Walkthrough 3: Update an Existing File System (p. 93).

### Working with Manual Backups

Amazon FSx enables you to take additional backups of your file systems at any time. You can do so using the Amazon FSx Management Console, API, or the AWS Command Line Interface (CLI). Any backups that you create manually do not expire automatically, and are kept until you manually delete them. Manual backups are retained after you have deleted the file system. For more information, see Deleting Backups (p. 52).

### Creating Manual 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 manual 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.

### 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.
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 right 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 Storage capacity is 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.

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.

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

Setting Up a Custom Backup Schedule

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.

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 DescribeBackup. 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon FSx file system ID</td>
<td>No default value</td>
<td>The file system ID for the file system that you want to back up.</td>
</tr>
<tr>
<td>CRON schedule pattern for backups.</td>
<td>0 0/4 * * ? *</td>
<td>The schedule to run the CloudWatch event, triggering a new backup and deleting old backups outside of the retention period.</td>
</tr>
<tr>
<td>Backup retention (days)</td>
<td>7</td>
<td>The number of days to keep user-initiated backups. The Lambda function deletes user-initiated backups older than this number of days.</td>
</tr>
<tr>
<td>Name for backups</td>
<td>user-scheduled backup</td>
<td>The name for these backups, which appears in the Backup Name column of the Amazon FSx Management Console.</td>
</tr>
<tr>
<td>Backup notifications</td>
<td>Yes</td>
<td>Choose whether to be notified when backups are successfully initiated. A notification is always sent if there's an error.</td>
</tr>
<tr>
<td>Email address</td>
<td>No default value</td>
<td>The email address to subscribe to the SNS notifications.</td>
</tr>
</tbody>
</table>

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.

```json
{
  "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.
Administering File Systems

You can administer your Amazon 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 (p. 56)
- File Shares (p. 58)
- User Sessions and Open Files (p. 60)
- Data Deduplication (p. 63)
- Storage Quotas (p. 65)
- Shadow Copies (p. 66)
- Encryption in Transit (p. 71)
- Working with Amazon FSx Maintenance Windows (p. 71)
- Best Practices for Administering Amazon FSx File Systems (p. 72)

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 Amazon 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 Connecting to Your Windows Instance in the Amazon EC2 User Guide for Windows Instances.
- 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 execute 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'll need to use `Invoke-Command`.

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 will be 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 Amazon FSx for Windows File Server file system.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName amznfsxzzzzzzz.corp.example.com -ConfigurationName FSxRemoteAdmin -scriptblock { fsx-command }
```
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 Amazon FSx for Windows File Server file system.

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 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. 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.
- 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.

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. 56).

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 Amazon 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\Marketing" -Description "CA share" -ContinuouslyAvailable $True
```

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

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

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 -?.
User Sessions and Open Files

You can monitor connected user sessions and open files on your Amazon 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 Amazon 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 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. 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-012345678901234567.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 Amazon 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.
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. 56).

Following are commands that you can use for user session and open file management.

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.
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. Because data deduplication runs as a background process, it doesn't significantly affect your file system's performance. It's also transparent to your users and connected clients. After data deduplication is enabled, it continually and automatically scans and optimizes your file system in the background.

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.

### 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 amznfsxzzzzzzzzz.corp.example.com -ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup}
```

When you enable data deduplication, a default data dedup schedule is in place. Also, the minimum file age before optimizing is set to 3 days.

### Setting a Data Deduplication Schedule

Even though the default schedule works well in most cases, you can modify an existing deduplication schedule by using the `Set-FsxDedupSchedule` command, shown as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -ConfigurationName FSxRemoteAdmin -ScriptBlock {New-FsxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days Mon,Tues,Wed,Sat -Start 08:00 -DurationHours 9}
```
This command modifies the default BackgroundOptimization schedule to run on days Monday to Wednesday and Saturday, starting the job at 8:00 am each day, with a maximum duration of 9 hours, after which the job will stop if it still running.

To modify the minimum file age before optimizing setting, use the Set-FSxDedupConfiguration command.

## Retrieving the Dedup Configuration

You can retrieve the data dedup configuration for a file system using the Get-FSxDedupConfiguration command, as follows.

```powershell
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -ConfigurationName FSxRemoteAdmin -ScriptBlock { Get-FsxDeDupConfiguration }
```

The output lists the dedup configuration parameters and their current values.

**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.

To view the amount of disk space you are saving from running data deduplication, use the following command.

```powershell
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -ConfigurationName FsxRemoteAdmin -ScriptBlock { Get-FSxDedupStatus } | select OptimizedFilesCount,OptimizedFilesSize,SavedSpace,OptimizedFilesSavingsRate
```

<table>
<thead>
<tr>
<th>OptimizedFilesCount</th>
<th>OptimizedFilesSize</th>
<th>SavedSpace</th>
<th>OptimizedFilesSavingsRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>12587</td>
<td>31163594</td>
<td>25944826</td>
<td>83</td>
</tr>
</tbody>
</table>

## 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. 56).

Following are commands that you can use for data deduplication.

### Data Deduplication Command
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable-FSxDedup</td>
<td>Enables data deduplication on the file share.</td>
</tr>
<tr>
<td>Disable-FSxDedup</td>
<td>Disables data deduplication on the file share.</td>
</tr>
<tr>
<td>Get-FSxDedupConfiguration</td>
<td>Retrieves deduplication configuration information, including Minimum file size and age for optimization, Compression settings, and Excluded file types and folders.</td>
</tr>
<tr>
<td>Set-FSxDedupConfiguration</td>
<td>Changes the deduplication configuration settings, including Minimum file size and age for optimization, compression settings, and excluded file types and folders.</td>
</tr>
<tr>
<td>Get-FSxDedupStatus</td>
<td>Retrieves the deduplication status, and includes read-only properties that describe optimization savings and status on the file</td>
</tr>
</tbody>
</table>
### Data Deduplication Command | Description
--- | ---
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.

**Note**

The values shown for following output fields do not represent the actual values, and you should not rely on them: Capacity, FreeSpace, UsedSpace, UnoptimizedSize, and SavingsRate.

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.
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. 56).

Following are commands that you can use to manage user storage quotas.

<table>
<thead>
<tr>
<th>User Storage Quotas Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable-FSxUserQuotas</td>
<td>Starts tracking or enforcing user storage quotas, or both.</td>
</tr>
<tr>
<td>Disable-FSxUserQuotas</td>
<td>Stops tracking and enforcement for user storage quotas.</td>
</tr>
<tr>
<td>Get-FSxUserQuotaSettings</td>
<td>Retrieves the current user-storage quota settings for the file system.</td>
</tr>
<tr>
<td>Get-FSxUserQuotaEntries</td>
<td>Retrieves the current user-storage quota entries for individual users and groups on the file system.</td>
</tr>
<tr>
<td>Set-FSxUserQuotas</td>
<td>Set the user storage quota for an individual user or group. Quota values are specified in bytes.</td>
</tr>
</tbody>
</table>

The online help for each command provides a reference of all command options. To access this help, run the command with 

| 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 Amazon FSx for Windows File Server file systems.

Topics
- Setting Shadow Copy Storage (p. 66)
- Deleting Shadow Copy Storage, Schedule, and All Shadow Copies (p. 68)
- Creating a Custom Shadow Copy Schedule (p. 68)
- Viewing Your Shadow Copy Schedule (p. 69)
- Deleting a Shadow Copy Schedule (p. 69)
- Creating a Shadow Copy (p. 70)
- Viewing Existing Shadow Copies (p. 70)
- Deleting Shadow Copies (p. 70)

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 undefined: `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. 48)](#).

**To set the amount of shadow copy storage on an Amazon 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.

   ```powershell
   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.

   ```powershell
   [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.

   ```powershell
   [fs-1234567890abcef12]: PS>Set-FsxShadowStorage -Default
   FSx Shadow Storage Configuration
   AllocatedSpace UsedSpace MaxSpace
   -------------- --------- --------
   0 0 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 currently used by shadow copies. Initially, this value is 0.
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. 56).

```
[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.

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. 48).

**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.

   ```powershell
   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 `file-system-Id` with the ID of the file system on which you're setting the shadow copy schedule.

```
PS C:\Users\delegateadmin> invoke-command -ComputerName file-system-Id -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.

<table>
<thead>
<tr>
<th>FSx Shadow Copy Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time: 2019-07-16T06:00:00+00:00</td>
</tr>
<tr>
<td>Days of Week: Monday,Tuesday,Wednesday,Thursday,Friday</td>
</tr>
<tr>
<td>WeeksInterval: 1</td>
</tr>
<tr>
<td>PSComputerName: fs-0123456789abcdef1</td>
</tr>
<tr>
<td>RunspaceId: 12345678-90ab-cdef-1234-567890abcdef</td>
</tr>
</tbody>
</table>

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. 56).

```
[fs-0123456789abcdef1]PS> Get-FsxShadowCopySchedule
FSx Shadow Copy Schedule

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Days of week</th>
<th>WeeksInterval</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-07-16T07:00:00+00:00</td>
<td>Monday,Tuesday,Wednesday,Thursday,Friday</td>
<td>1</td>
</tr>
<tr>
<td>2019-07-16T12:00:00+00:00</td>
<td>Monday,Tuesday,Wednesday,Thursday,Friday</td>
<td>1</td>
</tr>
</tbody>
</table>
```

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. 56).

```
[fs-0123456789abcdef1]PS> Remove-FsxShadowCopySchedule
```
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. 56).

```
[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. 56).

```
[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. 56).

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?
```
Encryption in Transit

You can use a set of custom PowerShell commands to control the encryption of your data in transit between your Amazon 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.

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. 56).

Following are commands that you can use to manage user in-transit encryption on your file system.

<table>
<thead>
<tr>
<th>Encryption in Transit Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-FSxSmbServerConfiguration</td>
<td>Retrieves the Server Message Block (SMB) server configuration.</td>
</tr>
<tr>
<td>Set-FSxSmbServerConfiguration</td>
<td>Sets in-transit data encryption on or off, and allows or disallows access to clients that do not support encryption.</td>
</tr>
</tbody>
</table>

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 -?.

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, your Single-AZ file system is temporarily unavailable, and Multi-AZ file systems 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 AWS CLI or one of the AWS SDKs (but not the console) to change the maintenance window for your file systems with the UpdateFileSync operation. For more information, see Walkthrough 3: Update an Existing File System (p. 93).

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. 56) and Getting Started with the Amazon FSx CLI for Remote Management on PowerShell (p. 56).
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.

```powershell
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.

```powershell
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName FSxRemoteAdmin -ScriptBlock { Set-FSxDedupConfiguration -MinimumFileAgeDays 0 }
```

For more information, see Data Deduplication (p. 63).

• Use the following command to turn on user storage quotas in “Track” mode, which is for reporting purposes only and not for enforcement.

```powershell
$QuotaLimit = Quota limit in bytes
$QuotaWarningLimit = Quota warning threshold in bytes
```

For more information, see Storage Quotas (p. 65).

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.

```powershell
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. 66).

Enforcing Encryption in Transit

The following command enforces encryption for clients connecting to your file system.
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}
```
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.

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 Amazon FSx for Windows File Server

With Amazon FSx for Windows File Server, you can monitor activity for your file systems using Amazon CloudWatch metrics.

Topics
- Monitoring with Amazon CloudWatch (p. 77)
- Amazon FSx for Windows File Server Dimensions (p. 78)
- How to Use Amazon FSx for Windows File Server Metrics (p. 78)
- Accessing CloudWatch Metrics (p. 79)
- Creating CloudWatch Alarms to Monitor Amazon FSx (p. 80)
- Logging Amazon FSx for Windows File Server API Calls with AWS CloudTrail (p. 81)

Monitoring with Amazon CloudWatch

You can monitor file systems using Amazon CloudWatch, which collects and processes raw data from Amazon 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 Are Amazon CloudWatch, Amazon CloudWatch Events, and Amazon CloudWatch Logs? in the Amazon CloudWatch User Guide.

As with Amazon EFS, Amazon S3, and Amazon EBS, 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, Amazon FSx for Windows File Server emits a data point per file system per minute.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataReadBytes</code></td>
<td>The number of bytes for file system read operations.</td>
</tr>
<tr>
<td></td>
<td>The <code>Sum</code> statistic is the total number of bytes associated with read operations during the period. To calculate the average throughput (Bytes per second) for a period, divide the <code>Sum</code> statistic by the number of seconds in the period.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes</td>
</tr>
<tr>
<td></td>
<td>Valid statistics: <code>Sum</code></td>
</tr>
<tr>
<td><code>DataWriteBytes</code></td>
<td>The number of bytes for file system write operations.</td>
</tr>
<tr>
<td></td>
<td>The <code>Sum</code> 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 <code>Sum</code> statistic by the number of seconds in the period.</td>
</tr>
</tbody>
</table>
### Amazon FSx for Windows File Server Dimensions

Amazon FSx for Windows File Server metrics use the `FSx` namespace and provide metrics for a single dimension, `FileSystemId`. A file system's ID can be found using the `aws fsx describe-file-systems` AWS CLI command, and it takes the form of `fs-0123456789abcdef0`.

### How to Use Amazon 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.
Accessing CloudWatch Metrics

You can see Amazon FSx metrics for CloudWatch in the following ways.

- The Amazon FSx console.
- The the CloudWatch console.
- The CloudWatch CLI (Command Line Interface).
- The the CloudWatch API.

The following procedures show you how to access the metrics using these various tools.

**Using the Amazon FSx console**

**To view metrics using the Amazon FSx console**

1. Open the Amazon FSx console at [https://console.aws.amazon.com/fsx/](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.

**Using the CloudWatch console.**

**To view metrics using the CloudWatch console**

2. In the navigation pane, choose *Metrics*.
3. Select the *FSx* namespace.
4. (Optional) To view a metric, type its name in the search field.
5. (Optional) To filter by dimension, select *FileSystemId*.

**Using the CloudWatch CLI**

**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](https://docs.aws.amazon.com/cli/latest/userguide/complete-reference.html).

**Using the CloudWatch API**

**To access metrics from the CloudWatch API**

- Call `GetMetricStatistics`. For more information, see [Amazon CloudWatch API Reference](https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/).
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 Amazon FSx console or the CloudWatch console.

The following procedures outline how to create alarms for Amazon FSx.

Using the Amazon FSx console

To set alarms using the CloudWatch 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 you want to create the alarm for.
3. Choose the Actions menu and choose View details.
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 and 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.
10. If you want CloudWatch to send you an email or SNS notification when the alarm state triggers the action, choose an alarm state from the Whenever this alarm stat is...: field.

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 and 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, Whenever values for the metric and choose Next.
12. On the Preview and create page, review the alarm you’re about to create and choose Create Alarm.

Using the CloudWatch console

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. This launches 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**, **Whenever** values for the metric.

5. If you want CloudWatch to send you an email when the alarm state is reached, in the **Whenever this alarm** field, choose **State is ALARM**. In the **Send notification to** field, 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**.

### Using the AWS CLI

**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 Amazon 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.

```json
{
    "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": {
```
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"
}
```
Amazon FSx for Windows File Server Performance

Amazon 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 (p. 84)
- Performance Details (p. 84)
- Measuring Performance Using CloudWatch Metrics (p. 87)

Overview

File system performance is measured by its latency, throughput, and I/O operations per second (IOPS).

Latency

Amazon 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 Amazon 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:

<table>
<thead>
<tr>
<th>Storage type</th>
<th>Disk throughput (MB/s per TiB of storage)</th>
<th>Disk IOPS (IOPs per TiB of storage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD</td>
<td>750</td>
<td>3,000</td>
</tr>
<tr>
<td>HDD</td>
<td>12 baseline; 80 burst (up to a max. of 1 GB/s per file system)</td>
<td>12 baseline; 80 burst</td>
</tr>
</tbody>
</table>

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 AWS 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.

The following table shows the full set of specifications for throughput capacity, along with baseline and burst levels, and amount of memory for caching on the file server.
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 Amazon FSx for Windows File Server Metrics (p. 78).
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.

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.
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**

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
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.
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).

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.

```bash
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.

```bash
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-configuration WeeklyMaintenanceStartTime=1:01:30
```

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.

```bash
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-configuration AutomaticBackupRetentionDays=30
```
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. 58).

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.
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 in this script, which you can get from the file system details view in the Amazon FSx console.

```bash
@echo off
net use S: /delete
net use S: \file-system-DNS-name\users\%username%
```
2. Open a PowerShell prompt and run gedit.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 can be accessed using the address `\file-system-DNS-name\share`. You can use the default share or create a different shared folder. For more information, see File Shares (p. 58).

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 systems DNS name (which you can obtain from the file system details view in the Amazon FSx Console), and credentials for accessing the shared folder.

   ```batch
   @echo off
   net use S: /delete
   net use S: \file-system-DNS-name\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 a Custom DNS Name for Your File System

Amazon FSx for Windows File Server provides a Domain Name Service (DNS) name for every file system of the form `filesystem.your-domain` using DNS integrated with Microsoft Active Directory. This DNS name is created on the AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) directory to which you join your file system.
You can add a custom DNS name for your Amazon FSx file system within your Microsoft AD directory. Doing this can make it easier to manage mapping the file shares within your file systems for large organizations or for discrete use cases. To add a custom DNS name for your file system from a Microsoft Windows Server client that is joined to the same AWS Managed Microsoft AD directory, use the DNS Manager application (\dnsmgmt.msc).

Note
Amazon FSx for Windows File Server supports custom DNS names only on Single-AZ 1 file systems. Support for custom DNS names on Multi-AZ and Single-AZ 2 file systems coming soon.

To add a custom DNS name to your Amazon FSx file system

1. Launch your Windows EC2 instance and connect it to the Microsoft Active Directory to which you’ve joined your Amazon FSx file systems. 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 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 AD, and Domain Admins or another group to which you've delegated DNS administration permissions in your self-managed AD) 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 \dnsmgmt.msc to open the DNS Manager application.
   Note
   If DNS Manager isn't already installed, you can install it from the Add Roles and Features wizard of the Server Manager application.
4. In DNS Manager, choose Forward Lookup Zones from the left navigation.
5. Open the context (right-click) menu for your AWS Managed Microsoft AD directory, and choose New Alias (CNAME) to open the New Resource Record dialog box.
6. For Alias name, enter the name to use for your Amazon FSx file system.
7. For Fully qualified domain name (FQDN) for target host, choose Browse.
8. In the Browse dialog box that opens, double-click on the record name for your target host.
9. Double-click on Forward Lookup Zone, and then double-click on the original name of your Amazon FSx file server.
10. Choose the name of a server, and then choose OK.

You’ve now added a CNAME value for your Amazon FSx file system. You can use this CNAME value to access your data within this Microsoft Active Directory.

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-DFSNServers.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.
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 AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to 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 (p. 100)
- File- and Folder-Level Access Control Using Windows ACLs (p. 102)
- File System Access Control with Amazon VPC (p. 103)
- Resource Administration Access Control with IAM for Amazon FSx (p. 105)
- Compliance Validation for Amazon FSx for Windows File Server (p. 109)

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. 107)
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 customer master keys (CMKs) to encrypt your file system. You choose the CMK used to encrypt and decrypt file systems (both data and metadata). You can enable, disable, or revoke grants on this CMK. This CMK can be one of the two following types:

- **AWS-managed CMK** – This is the default CMK, and it’s free to use.
- **Customer-managed CMK** – This is the most flexible master key to use, because you can configure its key policies and grants for multiple users or services. For more information on creating CMKs, see Creating Keys in the AWS Key Management Service Developer Guide.

If you use a customer-managed CMK as your master 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 CMK, you can choose when to disable, re-enable, delete, or revoke access to your CMK at any time. For more information, see Rotating Customer Master 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 CMKs. 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 customer master key (CMK), 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.
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 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-CCM [RFC5084] as its encryption algorithm, and also provides data integrity with signing using SMB Kerberos session keys.

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 Encryption in Transit (p. 71).

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 Active Directory in Amazon FSx for Windows File Server (p. 22).

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.

Related Links

- 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. 88).

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.

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

<table>
<thead>
<tr>
<th>Rules</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>53, 88, 123, 389, 445, 464</td>
</tr>
<tr>
<td>TCP</td>
<td>53, 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535</td>
</tr>
</tbody>
</table>

Add from and to IP addresses or security group IDs associated with the following source and destination resources:

- Client compute instances that you want to access your file system from.
- Other file servers that you expect your file system to communicate with in DFS Replication groups.

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.

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 (p. 105)
• Using Service-Linked Roles for Amazon FSx (p. 105)
• Understanding Resource Ownership (p. 107)
• Managing Access to Resources (p. 107)
• Amazon FSx API Permissions: Actions, Resources, and Conditions Reference (p. 107)

Amazon FSx for Windows File Server Resources and Operations

In Amazon FSx for Windows File Server, the primary resources are a file system and a backup. Amazon FSx for Windows File Server also supports additional resource types being file share and tags. However, for Amazon FSx, you can create file shares and tags only in the context of an existing file system. File shares and tags are referred to as subresources.

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.

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:

- `ec2:CreateNetworkInterface`

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.

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.

Managing Access to Resources

A permissions policy describes who has access to what. The following section explains the available options for creating permissions policies.

Note

This section discusses using IAM in the context of 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.

### Amazon FSx API and Required Permissions for Actions

<table>
<thead>
<tr>
<th>Amazon FSx API Operations</th>
<th>Required Permissions (API Actions)</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ds:DescribeDirectories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>kms:CreateGrant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>kms:DescribeKey</td>
<td></td>
</tr>
<tr>
<td>CreateBackup</td>
<td>elasticfilesystem:CreateMountTarget</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ec2:DescribeSubnets</td>
<td>arn:aws:fsx:region:account-id:backup/*</td>
</tr>
<tr>
<td>CreateFileSystemFromBackup</td>
<td>fsx:CreateFileSystemFromBackup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/backup-id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:fsx:region:account-id:file-system/backup-id</td>
<td></td>
</tr>
<tr>
<td>DeleteBackup</td>
<td>fsx:DeleteBackup</td>
<td>arn:aws:fsx:region:account-id:backup/*</td>
</tr>
<tr>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/backup-id</td>
<td></td>
</tr>
<tr>
<td>DescribeFileSystems</td>
<td>fsx:DescribeFileSystems</td>
<td>N/A</td>
</tr>
<tr>
<td>DescribeBackups</td>
<td>fsx:DescribeBackups</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>arn:aws:fsx:region:account-id:file-system/backup-id</td>
<td></td>
</tr>
</tbody>
</table>
## 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:

<table>
<thead>
<tr>
<th>Amazon FSx API Operations</th>
<th>Required Permissions (API Actions)</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:file-system/filesystem-id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/backup-id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:file-system/filesystem-id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:fsx:region:account-id:backup/backup-id</td>
</tr>
</tbody>
</table>
• **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.
Quotas

Following, you can find out about quotas when working with Amazon FSx for Windows File Server.

Topics

• Quotas That You Can Increase (p. 111)
• Resource Quotas for Each File System (p. 111)
• Additional Considerations (p. 112)
• Quotas Specific to Microsoft Windows (p. 112)

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 by contacting AWS Support.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
<th>Can Be Increased Up To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of file systems</td>
<td>100</td>
<td>Thousands</td>
</tr>
<tr>
<td>Total SSD storage for all file systems</td>
<td>512 TiB</td>
<td>Multiple PiBs</td>
</tr>
<tr>
<td>Total HDD storage for all file systems</td>
<td>512 TiB</td>
<td>Multiple PiBs</td>
</tr>
<tr>
<td>Total throughput capacity for all file systems</td>
<td>10 GBps</td>
<td>Hundreds of GBps</td>
</tr>
<tr>
<td>Total number of user-initiated backups for all file system</td>
<td>500</td>
<td>Thousands</td>
</tr>
</tbody>
</table>

To request a quota increase

1. Open the AWS Support Center page, sign in if necessary, and then choose Create Case.
2. Under Regarding, choose Service Limit Increase.
3. Under Limit Type, choose the type of limit to increase, fill in the necessary fields in the form, and then choose your preferred method of contact.

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.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Limit per file system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tags</td>
<td>50</td>
</tr>
</tbody>
</table>
## 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. You can also map your file share on your EC2-Classic instances (which are not in a VPC), but you must link them to your VPC by using ClassicLink. For more information about using ClassicLink, see ClassicLink in the Amazon EC2 User Guide for Linux Instances.

## Quotas Specific to Microsoft Windows

For more information, see NTFS limits on the Microsoft Windows Dev Center.
Troubleshooting Problems for 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 (p. 113)
• Trying to Create an Amazon FSx File System Fails (p. 114)
• File System Is In a Misconfigured State (p. 119)
• You can't Access Your File System Using Remote PowerShell (p. 120)
• You can't configure Microsoft Distributed File System Replication (DFS-R) on a Multi-AZ or Single-AZ 2 File System (p. 121)

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.

Potential Cause

The FSx elastic network interface for your file system was modified or deleted.

Resolution

You must not modify or delete the FSx 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 FSx elastic network interface. To learn more, see File System Access Control with Amazon VPC (p. 103).

Potential Cause

The security group you've specified for your Amazon FSx file system lacks the required inbound rules.

Resolution

Make sure that your security group has the inbound rules specified in Amazon VPC Security Groups (p. 103).

Potential Cause

The security group you've associated with your compute instances lacks the required outbound rules.

Resolution

Review the outbound rules specified in Amazon VPC Security Groups (p. 103), and make sure that the security group associated with your compute instances has the corresponding outbound rules.

Potential Cause
Your compute instances might not be correctly joined to one of two types of Active Directory. In a working setup, they can be joined to the AWS Managed Microsoft AD directory to which your file system is joined. Or they can be joined to a Microsoft Active Directory directory that has a one-way forest trust relationship established with the AWS Managed Microsoft AD directory.

Resolution

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.

Potential Cause

The Microsoft Windows file share that you're attempting to access doesn't exist.

Resolution

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. 58).

Potential Cause

The Active Directory user that you're accessing the file share as lacks the necessary access permissions.

Resolution

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.

Potential Cause

If you remove Allow Full control NTFS ACL permissions for the SYSTEM user on a folder that you shared, that share can become inaccessible.

Resolution

Recreate your file share or shares. For more information, see File Shares (p. 58). After they're recreated, you can map and use the Windows file shares from your compute instance or instances.

Potential Cause

You're using your Amazon FSx file system from on-premises using AWS Direct Connect or VPN, and you're using a public IP address range for the on-premises client.

Resolution

Use a private IP address range for the on-premises client. For more information, see Accessing Amazon FSx for Windows File Server File Systems from On-Premises (p. 16).

Trying to Create an Amazon FSx File System Fails

If a file system that you tried to create is in the Failed state, there can be a number of potential causes, as described following.

Topics

- Troubleshooting File Systems Joined to an AWS-Managed Microsoft Active Directory (p. 115)
- Troubleshooting File Systems Joined to a Self-Managed Active Directory (p. 115)
Troubleshooting File Systems Joined to an AWS-Managed Microsoft Active Directory

Use the following sections to help troubleshoot problems you experience while creating an Amazon FSx file system that is joined to your self-managed Active Directory.

Potential Cause

VPC security groups and network ACLs aren't using the recommended security group configuration for AWS Managed Microsoft AD.

Resolution

Make sure that the VPC security groups and network ACLs are configured using the recommended security group configuration. To learn more, see Creating FSx Security Groups, Step 6 (p. 104).

Troubleshooting File Systems Joined to a Self-Managed Active Directory

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.

Resolution

Use the following to troubleshoot and resolve the issue.

1. Verify that you've 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. 25)

2. Verify that you've configured the VPC Security Groups that you've 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 Microsoft's Active Directory documentation.

3. Verify that Your AD domain's DNS servers and domain controllers are active and able to respond to requests for the domain provided.

4. Ensure that the functional level of your AD domain is Windows Server 2008 R2 or higher.

5. Make sure that the firewall rules on your AD domain's domain controllers allow traffic from your Amazon FSx file system. For more information, see Microsoft's Active Directory documentation.

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.
Resolution

Use the following to troubleshoot and resolve the issue.

1. Verify that you’re entering only the username 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 username.
   DO NOT use the Distinguished Name (DN) when entering the service account username (CN=ServiceAcct,OU=example,DC=corp,DC=com).

2. Verify that the service account that you have provided exists in your AD domain.

3. Make sure you have delegated the required permissions to the service account that you’ve 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 host name
   - Validated ability to write to the service principal name

   To learn more on creating a service account with correct permissions, see Delegating Privileges to Your Amazon FSx Service Account (p. 26).

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.

Resolution

Use the following procedure to troubleshoot and resolve the issue.

- Make sure you have delegated the required permissions to the service account that you’ve 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 host name
  - Validated ability to write to the service principal name

  To learn more on creating a service account with correct permissions, see Delegating Privileges to Your Amazon FSx Service Account (p. 26).

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.

**Resolution**

Use the following procedure to troubleshoot and resolve the issue.

- Make sure that the service account you've provided has not 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. For more information, see Delegating Privileges to Your Amazon FSx Service Account (p. 26)

**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.

**Resolution**

Use the following procedure 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've 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 host name
   - Validated ability to write to the service principal name

To learn more on creating a service account with correct permissions, see Delegating Privileges to Your Amazon FSx Service Account (p. 26).

**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.

**Resolution**

Use the following procedure 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 AD domain as the one to which you were trying to join the file system.

3. If you did not provide an administrator group parameter, 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 need to provide that name for the group.

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.

Resolution

When creating your file system, Amazon FSx was able to reach your AD domain’s DNS servers and domain controllers, and join the file system successfully to your AD domain. However, while completing file system creation, Amazon FSx lost connectivity to or membership in your domain. Use the following 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 AD domain are still active, and were not deleted or otherwise manipulated.

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.

Resolution

Make sure you have delegated the required permissions to the service account that you've provided. Use the following 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 host name
  - Validated ability to write to the service principal name

To learn more about creating a service account with the correct permissions, see Delegating Privileges to Your Amazon FSx Service Account (p. 26).

If you encounter problems not listed here while using Amazon FSx, please ask a question in the Amazon FSx Forum or reach out to AWS Support.
File System Is In a Misconfigured State

When you update your file system's self-managed Active Directory configuration, a number of causes can put your file system into the **Misconfigured** state. Each cause has its own resolution, as follows.

**Potential Cause**

Amazon FSx can't communicate with your Microsoft Active Directory domain controller or controllers. This is because Amazon FSx can't reach either the DNS servers provided or domain controllers for your domain. To learn more, see Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24).

**Resolution**

Make sure that your networking configuration allows traffic from the file system to the domain controller. To update the configuration, you can use the console. On the navigation pane, choose **File systems**, and choose the file system to change. On its **File system details** page, you can find **Update** on the **Networking and security** tab. You can also use the Amazon FSx API operation `update-file-system`. To learn more, see Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24).

**Potential Cause**

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. To learn more, see Using Amazon FSx with Your Self-Managed Microsoft Active Directory (p. 24).

**Resolution**

Verify the service account credentials, then update the configuration with new service account credentials. To update the configuration, you can use the console. On the navigation pane, choose **File systems**, and choose the file system to change. On its **File system details** page, you can find **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.

**Potential Cause**

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.

**Resolution**

Add the required permissions to the Amazon FSx service account, or create a new service account with the required permissions. To learn more about doing this, see Delegating Privileges to Your Amazon FSx Service Account (p. 26). Then update the file system's self-managed AD configuration with the new service account credentials. To update the configuration, you can use the console. On the navigation pane, choose **File systems**, and choose the file system to change. On its **File system details** page, you can find **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.

**Potential Cause**

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.

**Resolution**

Identify another service account or create a new service account that can join new computers to the domain. Then update the file system's self-managed AD configuration with the new service account credentials.
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.

Potential Cause

The file system's security group lacks the required inbound rules to allow a remote PowerShell connection.

Resolution

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. To learn more, see Amazon VPC Security Groups (p. 103)

Potential Cause

A language localization error occurs when trying to initiate a remote PowerShell session.

Resolution

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 FSxFileSystem-DNS-Name -ConfigurationName FSxRemoteAdmin -scriptblock {fsx-command} -SessionOption (New-PSSessionOption -uiCulture "en-US")
```

```
PS C:\Users\delegateadmin> Enter-Pssession -ComputerName FSxFileSystem-DNS-Name -ConfigurationName FSxRemoteAdmin -SessionOption (New-PSSessionOption -uiCulture "en-US")
```
You can't configure Microsoft Distributed File System Replication (DFS-R) on a Multi-AZ or Single-AZ 2 File System

DFS-R is not supported on Multi-AZ and Single-AZ 2 file systems.

Potential Cause

Multi-AZ file systems are configured for redundancy across multiple access zones natively, and do not support DFS-R.

Resolution

Use the Multi-AZ deployment type for high availability across multiple AZs. To learn more, see Availability and Durability: Single-AZ and Multi-AZ File Systems (p. 17).
Document History

- **API version:**
- **Latest documentation update:** March 26, 2020

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.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support added for hard disk drive (HDD) storage (p. 122)</td>
<td>HDD storage gives you price and performance flexibility when using Amazon FSx for Windows File Server. For more information, see Optimizing Costs with Amazon FSx.</td>
<td>March 26, 2020</td>
</tr>
<tr>
<td>Support added for file transfer using AWS DataSync (p. 122)</td>
<td>You can now use AWS DataSync to transfer files to and from your Amazon FSx for Windows File Server. For more information, see Migrate Files to Amazon FSx for Windows File Server Using AWS DataSync.</td>
<td>February 4, 2020</td>
</tr>
<tr>
<td>Amazon FSx for Windows File Server releases support for additional Windows file system administration tasks (p. 122)</td>
<td>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.</td>
<td>November 20, 2019</td>
</tr>
<tr>
<td>Amazon FSx for Windows File Server releases native Multi-AZ support (p. 122)</td>
<td>You can use Multi-AZ deployment for Amazon 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.</td>
<td>November 20, 2019</td>
</tr>
<tr>
<td>Amazon FSx for Windows File Server releases support for managing user sessions and open files (p. 122)</td>
<td>You can now use the Shared Folders tool native to Microsoft Windows to manage user sessions and open files on your Amazon FSx for Windows File Server file systems. For more information, see Managing User Sessions and Open Files.</td>
<td>October 17, 2019</td>
</tr>
<tr>
<td>Feature Description</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>Amazon FSx releases support for Microsoft Windows shadow copies (p. 122)</strong></td>
<td>You can now configure Windows shadow copies on your Amazon 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 <a href="#">Working with Shadow Copies</a>.</td>
<td>July 31, 2019</td>
</tr>
<tr>
<td><strong>Amazon FSx releases shared Microsoft Active Directory support (p. 122)</strong></td>
<td>You can now join Amazon 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 <a href="#">Active Directory Support</a>.</td>
<td>June 25, 2019</td>
</tr>
<tr>
<td><strong>Amazon FSx releases enhanced Microsoft Active Directory support (p. 122)</strong></td>
<td>You can now join Amazon 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 <a href="#">Active Directory Support</a>.</td>
<td>June 24, 2019</td>
</tr>
<tr>
<td><strong>Amazon FSx complies with SOC certification (p. 122)</strong></td>
<td>Amazon FSx has been assessed to comply with SOC certification. For more information, see <a href="#">Security and Data Protection</a>.</td>
<td>May 16, 2019</td>
</tr>
<tr>
<td><strong>Added clarifying note regarding AWS Direct Connect, VPN, and inter-region VPC peering connection support (p. 122)</strong></td>
<td>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 <a href="#">Supported Access Methods</a>.</td>
<td>February 25, 2019</td>
</tr>
<tr>
<td><strong>AWS Direct Connect, VPN, and inter-region VPC peering connection support added (p. 122)</strong></td>
<td>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 <a href="#">Supported Access Methods</a>.</td>
<td>February 22, 2019</td>
</tr>
<tr>
<td>Amazon FSx is now generally available (p. 122)</td>
<td>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.</td>
<td>November 28, 2018</td>
</tr>
</tbody>
</table>