Windows User Guide

Amazon FSx for Windows File Server

Amazon FSx for Windows File Server: Windows User Guide

Table of Contents

What is FSx for Windows File Server? 1
Amazon FSx resources 1
Accessing file shares 2
Security and data protection 2
Availability and durability 3
Managing file systems 3
Price and performance flexibility 3
Pricing for Amazon FSx 4
Assumptions 4
Prerequisites 4
Amazon FSx for Windows File Server forums 5
Are you a first-time user of Amazon FSx? 5
FSx for Windows best practices 6
General best practices 6
Creating a monitoring plan 6
Ensuring that your file systems have sufficient resources 6
Security best practices 6
Network security 7
Active Directory 7
Avoid losing availability due to Active Directory misconfiguration 8
Windows ACLs 9
Configuring and right-sizing your file system 9
Selecting a deployment type 9
Selecting a throughput capacity 9
Increasing storage capacity and throughput capacity 10
Modifying throughput capacity during idle periods 10
Getting started 11
Setting up your AWS account 11
.... 12
Step 1. Setting up an Active Directory 13
Step 2: Launch a Windows instance in the Amazon EC2 console 15
Step 3: Connect to your instance 16
Step 4: Join your instance to your Directory Service directory 19
Step 5. Create your file system 20

Step 6. Map your file share to an EC2 instance running Windows Server 26
Step 7. Write data to your file share 27
Step 8. Back up your file system 28
Step 9. Clean up resources 28
Accessing your data 30
Supported clients 30
Accessing data from within the AWS Cloud 31
Accessing data from a different VPC, AWS account, or AWS Region 32
Accessing data from on-premises 33
Accessing data using default DNS names 34
Using Kerberos authentication with DNS names 35
Support for Distributed File System (DFS) namespaces 35
Accessing data using DNS aliases 35
Using Kerberos authentication and encryption with DNS aliases 36
Associate DNS aliases with your file system 37
Configure service principal names (SPNs) for Kerberos 38
Update or create a DNS CNAME record 41
Enforcing Kerberos authentication using Group Policy Objects (GPOs) 43
Accessing data using file shares 44
Mapping file shares 44
Mapping a file share on an Amazon EC2 Windows instance 45
Mounting a file share on an Amazon EC2 Mac instance 47
Mounting a file share on an Amazon EC2 Linux instance 50
Automatically mount file shares on an Amazon EC2 Linux instance 55
Managing file shares 58
New-FSxSmbShare command fails with a one-way trust 64
Availability and durability 65
Choosing Single-AZ or Multi-AZ file system deployment type 65
Feature support by deployment type 66
Failing over process 66
Failover experience on Windows clients 67
Failover experience on Linux clients 67
Testing failover on a file system 68
Single-AZ and Multi-AZ file system resources 68
Subnets 68
File system elastic network interfaces 69

Working with Active Directory 70
Using AWS Managed Microsoft AD 71
Networking prerequisites 72
Using a resource forest isolation model 77
Test your Active Directory configuration 77
Using AWS Managed Microsoft AD in different VPC or account 78
Validating connectivity to your Active Directory domain controllers 79
Using a self-managed Active Directory 82
Prerequisites 83
Service account permissions 88
Best practices when using a self-managed Active Directory 88
Amazon FSx service account 98
Delegating privileges to Amazon FSx 99
Validating your Active Directory configuration 101
Join FSx to a self-managed Active Directory 105
Getting IP addresses for manual DNS entries 115
Update self-managed Active Directory 116
Changing the Amazon FSx service account 118
Monitoring self-managed Active Directory updates 120
Performance 123
File system performance 123
Additional performance considerations 124
Latency 125
Throughput and IOPS 125
Single-client performance 125
Burst performance 125
Throughput capacity & performance 126
Choosing throughput capacity 128
Storage configuration & performance 130
HDD burst performance 130
Example: storage capacity and throughput capacity 131
Measuring performance using CloudWatch metrics 132
Troubleshooting performance 132
Determine file system throughput and IOPS limits 132
What is network I/O vs. disk I/O? Why are they different? 133
Why is CPU or memory usage high when network I/O is low? 133

What is bursting? How much bursting is my file system using? What happens when burst credits run out? ... 134

I see a warning on the **Monitoring & performance** page – do I need to change my file system’s configuration? ... 134

My metrics were temporarily missing, should I be concerned? ... 135

Administering file systems ... 136

Amazon FSx file system status ... 137

Using the Amazon FSx CLI for PowerShell ... 138

Starting an Amazon FSx remote PowerShell session ... 140

One-time file system setup tasks ... 141

Managing storage consumption ... 141

Turning on shadow copies to enable end-users to recover files and folders to previous versions ... 142

Enforcing encryption in transit ... 142

Troubleshooting access to the Amazon FSx CLI on PowerShell ... 142

The file system's security group lacks the required inbound rules to allow a remote PowerShell connection ... 143

You have an external trust configured between the AWS managed Microsoft Active Directory and your on-premises Active Directory ... 143

A language localization error occurs when trying to initiate a remote PowerShell session . 143

Maintenance window ... 143

Changing the weekly maintenance window ... 144

DNS aliases ... 145

DNS alias status ... 147

Using DNS aliases with Kerberos ... 147

Viewing existing DNS aliases ... 148

Associating DNS aliases with file systems ... 148

Managing DNS aliases on existing file systems ... 150

User sessions and open files ... 152

Using the GUI to manage users and sessions ... 153

Using PowerShell to manage user sessions and open files ... 156

File Server Resource Manager ... 156

Key capabilities ... 157

How to Get Started ... 158

Quota Management ... 161

File Groups ... 174

- File Screening ... 179
- File Classification ... 191
- Storage Reports ... 211
- File Management Tasks ... 225
- FSRM Settings ... 225
- Event Logs ... 231
- Common Use Cases ... 232

Managing storage ... 239
- Optimizing storage costs ... 240
- Managing storage capacity ... 240
- Managing storage types ... 244
- Managing SSD IOPS ... 245
- Data deduplication ... 246
- Managing storage quotas ... 250
- Increasing storage capacity ... 252
- Monitoring storage increases ... 253
- Increasing storage capacity dynamically ... 256
- Updating storage type ... 262
- Monitoring storage type updates ... 263
- Updating the SSD IOPS ... 264
- Monitoring provisioned SSD IOPS updates ... 265
- Managing data deduplication ... 266
- Troubleshooting data deduplication ... 270

Using DFS Namespaces ... 272
- Using DFS Namespaces ... 272
- Improving performance with shards ... 273
- Group file systems into one namespace ... 273
- Sharding data using DFS Namespaces for scale-out performance ... 275

Managing throughput capacity ... 277
- How throughput scaling works ... 277
- Knowing when to modify throughput capacity ... 278
- Modifying throughput capacity ... 279
- Monitoring throughput capacity updates ... 280

Managing network type ... 283
- Using dual-stack mode ... 283
- Changing network type ... 284

Tagging resources ... 285
Tag basics ... 285
Tagging your resources ... 286
Tag restrictions ... 286
Permissions required to tag resources ... 287
Update a file system using the AWS CLI ... 287
Protecting your data ... 289
Protecting your data with backups ... 289
Working with automatic daily backups ... 291
Working with user-initiated backups ... 291
Using AWS Backup with Amazon FSx ... 292
Copying backups ... 293
Restoring backups to new file system ... 295
Creating user-initiated backups ... 296
Deleting backups ... 297
Size of backups ... 297
Copying backups ... 298
Restoring a backup ... 299
Protecting data with shadow copies ... 300
Best practices ... 301
Setting up shadow copies ... 302
Configure shadow copies to use default settings ... 307
Setting the maximum amount of shadow copy storage ... 309
Viewing shadow copy storage ... 311
Creating a custom shadow copy schedule ... 312
Viewing the shadow copy schedule ... 313
Creating a shadow copy ... 314
Viewing existing shadow copies ... 314
Deleting shadow copies ... 314
Deleting a shadow copy schedule ... 316
Deleting a shadow copy configuration ... 316
Troubleshooting shadow copies ... 317
Scheduled replication ... 318
Using FSx for Windows File Server with Microsoft SQL Server ... 319
Using Amazon FSx for Active SQL Server Data Files ... 319
Create a Continuously Available Share ... 320

Configure SMB timeout settings 320
Using Amazon FSx as an SMB File Share Witness 320
Migrating to Amazon FSx 321
Migrating files to FSx for Windows File Server 321
Migrating best practices 322
Migrating files using AWS DataSync 322
Migrating files using Robocopy 325
Migrating file share configurations 329
Migrating your on-premises DNS configuration to FSx for Windows File Server 331
Cutting over to FSx for Windows File Server 334
Preparing for the cutover to Amazon FSx 335
Configure SPNs for Kerberos authentication 335
Update the DNS CNAME records for the Amazon FSx file system 339
Monitoring file systems 341
Automated and manual monitoring 341
Automated tools 341
Manual monitoring tools 342
Monitoring with Amazon CloudWatch 343
Metrics and dimensions 344
Using CloudWatch metrics 350
Performance warnings and recommendations 354
Accessing file system metrics 355
Creating CloudWatch alarms 360
CloudTrail logs 362
Amazon FSx information in CloudTrail 363
Understanding Amazon FSx log file entries 364
Security 366
Data protection 367
Data encryption 368
Encryption at rest 368
Encryption in transit 370
Windows ACLs 372
Related Links 373
File system access control with Amazon VPC 373
Amazon VPC Security Groups 373
Amazon VPC Network ACLs 377

- Logging end user access ... 378
  - Audit event log destinations ... 379
  - Migrating your audit controls ... 381
  - Viewing event logs ... 381
  - Setting file and folder auditing controls ... 388
  - Managing file access auditing ... 390
- Identity and access management ... 395
  - Audience ... 395
  - Authenticating with identities ... 396
  - Managing access using policies ... 397
  - How Amazon FSx for Windows File Server works with IAM ... 399
  - Identity-based policy examples ... 404
  - AWS managed policies ... 407
  - Troubleshooting ... 422
  - Using tags with Amazon FSx ... 424
  - Using service-linked roles ... 429
- Compliance Validation ... 436
- Interface VPC endpoints ... 436
  - Considerations for Amazon FSx interface VPC endpoints ... 436
  - Creating an interface VPC endpoint for Amazon FSx API ... 437
  - Creating a VPC endpoint policy for Amazon FSx ... 438

Working with other services ... 439

- Using Amazon FSx with Amazon WorkSpaces Applications ... 439
  - Providing personal persistent storage to each user ... 440
  - Providing a shared folder across users ... 442
- Using FSx for Windows File Server with Amazon Kendra ... 443
  - File system performance ... 444

Quotas ... 445

- Quotas that you can increase ... 445
- Resource quotas for each file system ... 446
- Additional considerations ... 447
- Quotas specific to Microsoft Windows ... 448

Troubleshooting ... 449

- You can't access your file system ... 449
  - The file system elastic network interface was modified or deleted ... 450
  - The Elastic IP address attached to the file system elastic network interface was deleted ... 450

The file system security group lacks the required inbound or outbound rules. 450
The compute instance's security group lacks the required outbound rules 450
Compute instance not joined to an Active Directory 450
The file share doesn't exist 451
Active Directory user lacks required permissions 451
Allow Full control NTFS ACL permissions removed 451
Can't access a file system using an on-premises client 451
New file system is not registered in DNS 452
Can't access the file system using a DNS alias 452
Can't access the file system using an IP address 454
Creating file system fails 454
Misconfigured VPC security group 455
Duplicate file system administrators group names 455
DNS servers or domain controllers unreachable 456
Invalid service account credentials 457
Amazon FSx can't access your Active Directory service account credentials in AWS Secrets Manager 459
Insufficient service account permissions 460
Service account capacity exceeded 461
Can't access the OU 462
Bad file system admin group 463
Amazon FSx lost connectivity in domain 463
Service account does not have correct permissions 464
Unicode characters used in creation parameters 465
Switching storage type to HDD while restoring a backup fails 465
File system is in a misconfigured state 466
Misconfigured file system: Amazon FSx can't reach either the DNS servers or domain controllers for your domain. 468
Misconfigured file system: The service account credentials are invalid 468
Misconfigured file system: The AWS Secrets Manager secret or KMS key is not configured correctly 469
Misconfigured file system: The service account provided doesn't have permission to join the file system to the domain 469
Misconfigured file system: The service account can't join any more computers to domain . 470
Misconfigured file system: The service account doesn't have access to the OU 471
You can't configure DFS-R on a Multi-AZ or Single-AZ 2 file system 471

Storage or throughput capacity updates fail ... 471
Storage capacity increase fails because Amazon FSx can't access the file system's AWS KMS key ... 472
Storage or throughput capacity update fails because the self-managed Active Directory is misconfigured ... 472
Storage capacity increase fails because of insufficient throughput capacity ... 473
Throughput capacity update to 8 MBps fails ... 473
Document history ... 474

What is FSx for Windows File Server?

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

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

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

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

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

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

FSx for Windows File Server backups are file-system-consistent, highly durable, and incremental. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in

Microsoft Windows. Automatic daily backups are turned on by default when you create a file system, and you can also take additional manual backups at any time. For more information, see Protecting your data with backups.

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

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

Accessing file shares

Amazon FSx is accessible from compute instances with the SMB protocol (supporting versions 2.0 to 3.1.1). You can access your shares from all Windows versions starting from Windows Server 2008 and Windows 7, and also from current versions of Linux. You can map your Amazon FSx file shares on Amazon Elastic Compute Cloud (Amazon EC2) instances, and on WorkSpaces instances, Amazon AppStream 2.0 instances, and VMware Cloud on AWS VMs.

You can access your file shares from on-premises compute instances using AWS Direct Connect or Site-to-Site 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 from compute instances that are in a different Amazon VPC, account, or AWS Region. You do so using VPC peering or transit gateways. For more information, see Accessing data from within the AWS Cloud.

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.

Availability and durability

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

Managing file systems

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

Price and performance flexibility

FSx for Windows File Server gives you the price and performance flexibility by offering both solid state drive (SSD) and hard disk drive (HDD) storage types. HDD storage is designed for a broad spectrum of workloads, including home directories, user and departmental shares, and content management systems. SSD storage is designed for the highest-performance and most latency-sensitive workloads, including databases, media processing workloads, and data analytics applications.

With FSx for Windows File Server, you can provision file system storage, SSD IOPS, and throughput independently to achieve the right mix of cost and performance. You can modify your file system's storage, SSD IOPS, and throughput capacities to meet changing workload needs, so that you pay only for what you need.

Pricing for Amazon FSx

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

Assumptions

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

In this guide, we make the following assumptions:

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

Prerequisites

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

- An AWS account with the permissions necessary to create an Amazon FSx file system and an Amazon EC2 instance. For more information, see Setting up your AWS account.
- 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.*
- Amazon FSx works with Microsoft Active Directory to perform user authentication and access control. You join your Amazon FSx file system to a Microsoft Active Directory while creating it. For more information, see Working with Microsoft Active Directory.

- 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.
- Install and configure the AWS Command Line Interface (AWS CLI). Supported versions are 1.9.12 and newer. For more information, see Installing, updating, and uninstalling the AWS CLI in the *AWS Command Line Interface User Guide.*

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

Amazon FSx for Windows File Server forums

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

Are you a first-time user of Amazon FSx?

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

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

Best practices for FSx for Windows File Server

We recommend that you follow these best practices when working with Amazon FSx for Windows File Server.

Topics

- General best practices
- Security best practices
- Active Directory
- Configuring and right-sizing your file system

General best practices

Creating a monitoring plan

You can use file system metrics to monitor your storage and performance usage, understand your usage patterns, and trigger notifications when your usage approaches your file system’s storage or performance limits. Monitoring your Amazon FSx file systems along with the rest of your application environment enables you to quickly debug any issues that may impact performance.

Ensuring that your file systems have sufficient resources

Having insufficient resources can result in increased latency and queuing for I/O requests, which might appear as complete or partial unavailability of your file system. For more information about monitoring performance and accessing performance warnings and recommendations, see Performance warnings and recommendations.

Security best practices

We recommend that you follow these best practices for administering your file system’s security and access controls. For more detailed information on configuring Amazon FSx to meet your security and compliance objectives, see Security in Amazon FSx.

Network security

Don't modify or delete the ENI that's associated with your file system

Your Amazon FSx file system is accessed through an elastic network interface (ENI) that resides in the virtual private cloud (VPC) that'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.

Using security groups and network ACLs

You can use security groups and network access control lists (ACLs) to limit access to your file systems. For VPC security groups, the default security group is already added to your file system in the console. Make sure that the security group and the network ACLs for the subnets where you create your file system allow traffic on the ports.

Active Directory

When you create an Amazon FSx file system, you can join it to your Microsoft Active Directory domain to provide user authentication, and share-, file-, and folder-level access control authorization. Your users can use their existing Active Directory accounts to connect to file shares and access files and folders within them. In addition, you can migrate the existing security ACL configuration to Amazon FSx without any modifications. Amazon FSx provides you with two options for Active Directory: **AWS managed Microsoft Active Directory** or **self-managed Microsoft Active Directory**.

If you’re using an **AWS managed Microsoft Active Directory**, we recommend leaving the default settings of your Active Directory security group. If you do modify these settings, ensure that you maintain a network configuration that satisfies the network requirements. For more information, see Networking prerequisites.

If you’re using a **self-managed Microsoft Active Directory**, you have additional options for configuring your file system. We recommend the following best practices for initial configuration when using Amazon FSx with your self-managed Microsoft Active Directory:

- **Assign subnets to a single Active Directory site**: If your Active Directory environment has a large number of domain controllers, use **Active Directory Sites and Services** to assign the subnets used by your Amazon FSx file systems to a single Active Directory site with the highest availability and reliability. Make sure that the VPC security group, VPC network ACL, Windows firewall rules on your DCs, and any other network routing controls you have in your Active Directory infrastructure allow communication from Amazon FSx on the required ports. This allows Windows to revert to other DCs if it can't use the assigned Active Directory site. For more information, see File system access control with Amazon VPC.

- **Use a separate Organizational Unit (OU)**: Use an OU for your Amazon FSx file systems that’s separate from any other organizational units that you might have.
- **Configure your service account with minimum privileges required**: Configure or delegate the service account that you provide to Amazon FSx with the minimum privileges required. For more information, see Using a self-managed Microsoft Active Directory.
- **Continuously verify your Active Directory configuration**: Run the **Amazon FSx Active Directory validation tool** against your Active Directory configuration prior to creating your Amazon FSx file system to verify that your configuration is valid for use with Amazon FSx, and to discover any warnings and errors that the tool might expose.
- **Storing Active Directory credentials using AWS Secrets Manager**: You can use AWS Secrets Manager to securely store and manage your Microsoft Active Directory domain join service account credentials. This approach eliminates the need to store sensitive credentials in plaintext in application code or configuration files, strengthening your security posture. For more information, see Storing Active Directory credentials using AWS Secrets Manager.

Avoid losing availability due to Active Directory misconfiguration

When using Amazon FSx with your self-managed Microsoft Active Directory, it's important to have a valid Active Directory configuration not only during the creation of your file system, but also for ongoing operations and availability. During failure recovery events, routine maintenance events, and throughput capacity update actions, Amazon FSx rejoins file server resources to your Active Directory. If the Active Directory configuration is not valid during an event, your file system changes to a status of **Misconfigured**, and is at risk of becoming unavailable. Here are some ways that you can avoid losing availability:

- **Keep your Active Directory configuration updated with Amazon FSx**: If you make changes, such as resetting the password of your service account, make sure you update the configuration for any file systems using this service account.
- **Monitor for Active Directory misconfiguration**: Set Misconfigured status notifications for yourself so that you can reset your file system’s Active Directory configuration, if necessary. For an example that uses a Lambda-based solution to achieve this, see Monitoring the health of Amazon FSx file systems using Amazon EventBridge and AWS Lambda.

- **Validate your Active Directory configuration regularly**: If you want to proactively detect an Active Directory misconfiguration, we recommend that you run the Active Directory Validation tool against your Active Directory configuration on an ongoing basis. If you receive warnings or errors when running the validation tool, it means that your file system is at risk of becoming misconfigured.
- **Don't move or modify computer objects created by FSx**: Amazon FSx creates and manages computer objects in your Active Directory, using the service account and permissions that you provide. Moving or modifying these computer objects can result in your file system becoming misconfigured.

Windows ACLs

With Amazon FSx, you use standard Windows access control lists (ACLs) for fine-grained share-, file-, and folder-level access control. Amazon FSx file systems automatically verify the credentials of users who access file system data to enforce these Windows ACLs.

- **Don’t change the NTFS ACL permissions for the SYSTEM user**: Amazon FSx requires that the SYSTEM user have full control NTFS ACL permissions on all folders within your file system. Changing the NTFS ACL permissions for the SYSTEM user may result in your file system becoming inaccessible and future file system backups may become unusable.

Configuring and right-sizing your file system

Selecting a deployment type

Amazon FSx provides two deployment options: Single-AZ and Multi-AZ. We recommend using **Multi-AZ file systems** for most production workloads that require high availability to shared Windows file data. For more information, see Availability and durability: Single-AZ and Multi-AZ file systems.

Selecting a throughput capacity

Configure your file system with sufficient throughput capacity to meet not only the expected traffic of your workload, but also additional performance resources required to support the features you want to enable on your file system. For example, if you’re running data deduplication, the throughput capacity that you select must provide enough memory to run deduplication based on the storage that you have. If you’re using shadow copies, increase throughput capacity to a value that's at least three times the value that's expected to be driven by your workload to avoid Windows Server deleting your shadow copies. For more information, see Impact of throughput capacity on performance.

Increasing storage capacity and throughput capacity

Increase the storage capacity of your file system when it’s running low on free storage, or when you expect your storage requirements to grow larger than the current storage limit. We recommend maintaining at least 20% of free storage capacity at all times on your file system. We also recommend increasing throughput capacity by at least 20% before increasing storage capacity to offset any performance impact during a storage increase. You can use the *FreeStorageCapacity* CloudWatch metric to monitor the amount of free storage available and understand how it trends. For more information, see Managing storage capacity.

You should also increase the throughput capacity of your file system if your workload is constrained by the current performance limits. You can use the **Monitoring and performance** page on the FSx console to see when workload demands have approached or exceeded performance limits to determine whether your file system is under-provisioned for your workload.

To minimize the duration of storage scaling and avoid reduction in write performance, we recommend increasing your file system's throughput capacity before increasing storage capacity and then scaling back throughput capacity after the storage capacity increase is complete. Most workloads experience minimal performance impact during storage scaling. However, file systems with HDD storage type and workloads involving large numbers of end users, high levels of I/O, or datasets with large numbers of small files could temporarily experience a reduction in performance. For more information, see Storage capacity increases and file system performance.

Modifying throughput capacity during idle periods

Updating throughput capacity interrupts availability for a few minutes for Single-AZ file systems and causes failover and failback for Multi-AZ file systems. For Multi-AZ file systems, if there is ongoing traffic during failover and failback, any data changes made during this time will need to be synchronized between the file servers. The data synchronization process can take up to multiple hours for write-heavy and IOPS-heavy workloads. Although your file system will continue to be available during this time, we recommend scheduling maintenance windows and performing throughput capacity updates during idle periods when there is minimal load on your file system to reduce the duration of data synchronization. To learn more, see Managing throughput capacity.

Getting started with Amazon FSx for Windows File Server

Following, you can learn how to get started using FSx for Windows File Server. This getting started exercise includes the following steps.

1. Sign up for an AWS account and create an administrative user in the account.
2. Create an AWS Managed Microsoft AD Active Directory using the Directory Service. You will join your file system and compute instance to the Active Directory.
3. Create an Amazon Elastic Compute Cloud compute instance running Microsoft Windows Server. You will use this instance to access your file system.
4. Create an Amazon FSx for Windows File Server file system using the Amazon FSx console.
5. Map your file system to your EC2 instance
6. Write data to your file system.
7. Back up your file system.
8. Clean up the resources you created.

Topics

- Setting up your AWS account
- Step 1. Setting up an Active Directory
- Step 2: Launch a Windows instance in the Amazon EC2 console
- Step 3: Connect to your instance
- Step 4: Join your instance to your Directory Service directory
- Step 5. Create your file system
- Step 6. Map your file share to an EC2 instance running Windows Server
- Step 7. Write data to your file share
- Step 8. Back up your file system
- Step 9. Clean up resources

Setting up your AWS account

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

1. Sign up for an AWS account
2. Create a user with administrative access

Sign up for an AWS account

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

To sign up for an AWS account

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

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

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

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

Create a user with administrative access

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

Secure your AWS account root user

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

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.
2. Turn on multi-factor authentication (MFA) for your root user.

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

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.
2. In IAM Identity Center, grant administrative access to a user.

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

Sign in as the user with administrative access

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

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

Assign access to additional users

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

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

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

Step 1. Setting up an Active Directory

With Amazon FSx, you can operate fully managed file storage for Windows-based workloads. Likewise, Directory Service provides fully managed directories to use in your workload deployment.

If you have an existing corporate Active Directory 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 Active Directory 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 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 Active Directory

- If you don't already have one, use the Directory Service to create your AWS Managed Microsoft Active Directory directory. For more information, see Create Your AWS Managed Microsoft Active 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 Directory Service directory and Admin user.

- If you have an existing Active Directory, create a trust relationship between your AWS Managed Microsoft Active Directory and your existing Active Directory. 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 later. Notice that these AMIs are marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the `t2.micro` type, which is selected by default. Notice that this instance type is eligible for the free tier.
5. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, a security group appears that the wizard created and selected for you. You can use this security group, or you can choose the security group that you created when getting set up using the following steps:

   a. Choose **Edit security groups**.

   b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.

   c. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

   Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

> **Warning**
>
> Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
10. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. (If the **Public DNS (IPv4)** or **(IPv6)** 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)** or **(IPv6)**.)
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 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 Directory Service directory.

Step 4: Join your instance to your Directory Service directory

The following procedure shows you how to manually join an existing Amazon EC2 Windows instance to your Directory Service directory.

To join a Windows instance to your Directory Service directory

1. Connect to the instance using any Remote Desktop Protocol client.
2. Open the TCP/IPv4 or IPv6 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** or **Internet Protocol Version 6**.
3. (Optional) Select **Use the following DNS server addresses**, change the **Preferred DNS server** and **Alternate DNS server** addresses to the IPv4 or IPv6 addresses of the 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 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 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.

Step 5. Create your file system

To create your file system (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. On the **Select file system type** page, choose **FSx for Windows File Server**, and then choose **Next**. The **Create file system** page appears.
4. For **Creation method** choose **Standard create**.

File system details

1. 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 + - = . _ : /
2. 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.

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

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

4. For **Provisioned SSD IOPS**, you can choose either **Automatic** or **User-provisioned** mode.

   If you choose Automatic mode, FSx for Windows File Server automatically scales your SSD IOPS to maintain 3 SSD IOPS per GiB of storage capacity. If you choose User-provisioned mode, enter any whole number in the range of 96–400,000. Scaling SSD IOPS above 80,000 is available in US East (N. Virginia), US West (Oregon), US East (Ohio), Europe (Ireland), Asia Pacific (Tokyo), and Asia Pacific (Singapore). For more information, see Managing SSD IOPS.

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

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

Note

If you are going to enable file access auditing, you must choose a throughput capacity of 32 MBps or greater. For more information, see Logging end user access with file access auditing.

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

Network & security

1. 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 Directory Service directory and your Amazon EC2 instance.
2. 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 the security group you choose is in the same AWS Region as your file system. To ensure that you can connect an EC2 instance with your file system, you will need to add the following rules to your chosen security group:

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

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

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

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

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

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

For more information, see File system access control with Amazon VPC.

3. Multi-AZ file systems have a primary and a standby file server, each in its own Availability Zone and subnet. If you are creating a Multi-AZ file system (see step 5), choose a **Preferred subnet** value for the primary file server and a **Standby subnet** value for the standby file server.

   If you are creating a Single-AZ file system, choose the **Subnet** for your file system.
4. For **Network type**, select either **IPv4** (for only IPv4 support) or **Dual-stack** (for both IPv4 and IPv6 support). You can change the network type of an existing file system at any time. For more information, see Changing network type.

> **Note**
>
> If you intend to create an FSx for Windows File Server file system that uses dual-stack mode, you must first assign an Amazon-provided IPv6 CIDR block to your VPC and subnets. For more information, see Add IPv6 support for your VPC in the *Amazon Virtual Private Cloud User Guide*.

Windows authentication

- For **Windows authentication**, you have the following options:

Choose **AWS Managed Microsoft Active Directory** if you want to join your file system to a Microsoft Active Directory domain that is managed by AWS, and then choose your Directory Service directory from the list. For more information, see Working with Microsoft Active Directory.

Choose **Self-managed Microsoft Active Directory** if you want to join your file system to a self-managed Microsoft Active Directory domain, and provide the following details for your Active Directory. For more information see Using a self-managed Microsoft Active Directory.

- The fully qualified domain name of your Active Directory.

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

- **DNS server IP addresses**—the IPv4 or IPv6 addresses of the DNS servers for your domain.

  > ⓘ **Note**
  >
  > Your DNS server must have EDNS (Extension Mechanisms for DNS) enabled. If EDNS is disabled, your file system might fail to create.

- Credentials for an Active Directory service account that Amazon FSx uses to join the file system to your domain. You can provide these as either:
  - **Option 1**: AWS Secrets Manager secret ARN - The secret containing the username and password for a service account on your Active Directory domain. For more information, see Storing Active Directory credentials using AWS Secrets Manager.
  - **Option 2**: Plaintext credentials
    - **Service account username** – The user name of the service account in your existing Microsoft Active Directory. Don't include a domain prefix or suffix. For example, for `EXAMPLE\ADMIN`, use only `ADMIN`.
    - **Service account password** – The password for the service account.

- (Optional) **Organizational Unit (OU)**—the distinguished path name of the organizational unit in which you want to join your file system.
- (Optional) **Delegated file system administrators group**— the name of the group in your Active Directory that can administer your file system. The default group is 'Domain Admins'. For more information, see Amazon FSx service account.

Encryption, Auditing, and Access (DNS aliases)

1. For **Encryption**, choose the AWS KMS key **Encryption key** used to encrypt the data on your file system at rest. You can choose the default **aws/fsx (default)** that is managed by AWS KMS, an existing key, or a customer managed key by specifying the ARN for the key. For more information, see Encryption of data at rest.
2. For **Auditing - optional**, file access auditing is disabled by default. For information about enabling and configuring file access auditing, see Logging end user access with file access auditing.
3. For **Access - optional**, enter any DNS aliases that you want to associate with the file system. Each alias name must be formatted as a fully qualified domain name (FQDN). For more information, see Managing DNS aliases.

Backup and maintenance

For more information about automatic daily backups and the settings in this section, see Protecting your data with backups.

1. **Daily automatic backup** is enabled by default. You can disable this setting if you do not want Amazon FSx to take backups of your file system automatically on a daily basis.
2. If automatic backups are enabled, they occur within a time period known as the backup window. You can use the default window, or choose an **Automatic backup window start time** that is best for your workflow.
3. For **Automatic backup retention period**, you can use the default setting of **30** days, or set a value between 1 and 90 days that Amazon FSx will retain automatic daily backups of your file system for. This setting does not apply to user initiated backups, or backups taken by AWS Backup.
4. 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. For more information, see Tagging your Amazon FSx resources.

Choose **Next**.

Review your configuration and create

1. Review the file system configuration shown on the **Create file system** page. For your reference, you can see which file system settings you can and can't modify after file system is created. Choose **Create file system**.
2. After Amazon FSx creates the file system, choose the file system ID from the list in the **File Systems** dashboard to view the details. Choose **Attach**, and note the **DNS name** for your file system the **Network & security** tab. You will need it in the following procedure to map a share to an EC2 instance.

Step 6. 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 Directory Service directory. The name of your file share is not the same as the name of your file system.

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

1. Before you can mount a file share on a Windows instance, you must launch the EC2 instance and join it to the AWS Directory Service for Microsoft Active Directory that your file system has joined. 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*.
3. When you're connected, open File Explorer.
4. From the navigation pane, open the context (right-click) menu for **Network** and choose **Map Network Drive**.
5. Choose a drive letter of your choice for **Drive**.

6. You can map your file system using either its default DNS name assigned by Amazon FSx, or using a DNS alias of your choosing. This procedure describes mapping a file share using the default DNS name. If you want to map a file share using a DNS alias, see Accessing data using DNS aliases.

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

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

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

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

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

   For example, enter `\\fs-0123456789abcdef0.ad-domain.com\share`.

7. Choose whether the file share should **Reconnect at sign-in**, and then choose **Finish**.

Step 7. 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 8. 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 9. 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.*
2. On the Amazon FSx console, delete your file system. All automatic backups are deleted automatically. However, you still need to delete the manually created backups. The following steps outline this process:

    a. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.

b. From the console dashboard, choose the name of the file system you created for this exercise.

c. For **Actions**, choose **Delete file system**.

d. In the **Delete file system** dialog box that opens, decide whether you want to create a final backup. If you do, provide a name for the final backup. Any automatically created backups are also deleted.

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

e. Enter the ID of the file system that you want to delete in the **File system ID** box.

f. Choose **Delete file system**.

g. The file system is now being deleted, and its status in the dashboard changes to **DELETING**. When the file system has been deleted, it no longer appears in the dashboard.

h. Now you can delete any manually created backups for your file system. From the left-side navigation, choose **Backups**.

i. From the dashboard, choose any backups that have the same **File system ID** as the file system that you deleted, and choose **Delete backup**.

j. The **Delete backups** dialog box opens. Leave the check box checked for the ID of the backup you selected, and choose **Delete backups**.

Your Amazon FSx file system and related automatic backups are now deleted.

3. To delete the Directory Service directory you created for this exercise, see [Delete your directory]() in the AWS Directory Service Administration Guide.

Accessing your data

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

Topics

- Supported clients
- Accessing data from within the AWS Cloud
- Accessing data from on-premises
- Accessing data using default DNS names
- Support for Distributed File System (DFS) namespaces
- Accessing data using DNS aliases
- Accessing data using file shares
- Creating, updating, removing file shares

Supported clients

FSx for Windows File Server supports the Server Message Block (SMB) protocol versions 2.0 through 3.1.1, giving you the flexibility to connect to your file systems using a wide variety of compute instances and operating systems.

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

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

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

- Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019, and Windows Server 2022.
- Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 (including the Windows 7 and Windows 10 desktop experiences of WorkSpaces), and Windows 11.
- Linux, using the `cifs-utils` tool.
- macOS

Accessing data from within the AWS Cloud

Each Amazon FSx file system is associated with a Virtual Private Cloud (VPC). You can access your FSx for Windows File Server file system from anywhere in the file system's VPC, regardless of Availability Zone. You can also access your file system from VPCs that are in different AWS accounts or AWS Regions than the file system. In addition to the requirements described in the following sections for accessing FSx for Windows File Server resources, you also need to ensure that your file system's VPC security group is configured so that data and management traffic can flow between your file system and clients. For more information about configuring security groups with the required ports, see File system access control with Amazon VPC.

You can access FSx for Windows File Server file system from supported clients that are in the same VPC as your file system.

The following table illustrates the environments from which Amazon FSx supports access from clients in each of the supported environments, depending on when the file system was created.

Clients located in...	Access to file systems created before February 22, 2019	Access to file systems created before December 17, 2020	Access to file systems created after December 17, 2020
Subnets in which the file system is created	✓	✓	✓

Clients located in...	Access to file systems created before February 22, 2019	Access to file systems created before December 17, 2020	Access to file systems created after December 17, 2020
Primary CIDR blocks of the VPC in which the file system was created	✓	✓	✓
Secondary CIDRs of the VPC in which the file system was created		Clients with IP addresses in an RFC 1918 private IP address range: • 10.0.0.0/8 • 172.16.0.0/12 • 192.168.0.0/16	Clients with IP addresses outside the following CIDR block range: 198.19.0.0/16
Other CIDRs or peered networks			

Note

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

Accessing data from a different VPC, AWS account, or AWS Region

You can access your FSx for Windows File Server file system from support clients that are located in a different VPC, AWS account, or AWS Region than what is associated with your file system using 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 that are in another VPC. This access is possible even if the VPCs belong to different AWS accounts, and even if the VPCs reside in different AWS Regions.

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

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

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

Accessing data from on-premises

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

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

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

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

> **Note**
>
> Amazon FSx File Gateway is no longer available to new customers. Existing customers of FSx File Gateway can continue to use the service normally. For capabilities similar to FSx File Gateway, visit this blog post.

Accessing data using default DNS names

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

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

To find the DNS name:

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

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

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

Using Kerberos authentication with DNS names

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

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

Support for Distributed File System (DFS) namespaces

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

Accessing data using DNS aliases

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

Using DNS aliases, you can move your Windows file share data to FSx for Windows File Server and continue using the existing DNS names to access data on Amazon FSx. DNS aliases also allow you to use meaningful names that make it easier to administer tools and applications to connect to your Amazon FSx file systems. You can associate up to 50 DNS aliases with a file system at any one time. For more information about associating and disassociating DNS aliases with an FSx for Windows File Server file system, see Managing DNS aliases.

To configure access to your FSx for Windows File Server file systems using DNS aliases, you must perform the following steps:

1. Associate DNS aliases with your file system.
2. Create a DNS CNAME record for the file system and the DNS aliases associated with it.

For more information about using DNS aliases with FSx for Windows File Server file systems, see Managing DNS aliases.

Using Kerberos authentication and encryption with DNS aliases

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

To set up Kerberos authentication and encryption when accessing your file system using DNS aliases, see Configure service principal names (SPNs) for Kerberos.

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

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

To enforce Kerberos authentication and encryption when accessing your file system using DNS aliases, see Enforcing Kerberos authentication using Group Policy Objects (GPOs).

For more information about configure your file system to use DNS aliases, see the following procedures:

- Associate DNS aliases with your file system
- Configure service principal names (SPNs) for Kerberos
- Update or create a DNS CNAME record

- Enforcing Kerberos authentication using Group Policy Objects (GPOs)

Associate DNS aliases with your file system

You can associate DNS aliases with existing FSx for Windows File Server file systems, when you create new file systems, and when you create a new file system from a backup using the Amazon FSx console, CLI, and API. If you are creating an alias with a different domain name, input the full name, including parent domain, to associate an alias.

This procedure describes how to associate DNS aliases when creating a new file system using the Amazon FSx console. For information about associating DNS aliases with existing file systems, and details about using the CLI and API, see Managing DNS aliases.

To associate DNS aliases when creating a new file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system as described in Step 5. Create your file system of the Getting Started section.
3. In the **Access - optional** section of the **Create file system** wizard, enter the DNS aliases that you want to associate with your file system.

   Use the following guidelines when specifying DNS aliases:

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

   For DNS alias names, Amazon FSx stores alphabetic characters as lowercase letters (a-z), regardless of how you specify them: as uppercase letters, lowercase letters, or the corresponding letters in escape codes.
4. For **Maintenance preferences**, make any changes that you want.
5. In the **Tags - optional** section, add any tags that you need, and then choose **Next**.
6. Review the file system configuration shown on the **Create file system** page. Choose **Create file system** to create the file system.

Configure service principal names (SPNs) for Kerberos

We recommend that you use Kerberos-based authentication and encryption in transit with Amazon FSx. Kerberos provides the most secure authentication for clients that access your file system.

To enable Kerberos authentication for clients that access Amazon FSx using a DNS alias, you must add service principal names (SPNs) that correspond to the DNS alias on your Amazon FSx file system's Active Directory computer object. An SPN can only be associated with a single Active Directory computer object at a time. If you have existing SPNs for the DNS name configured for your original file system's Active Directory computer object, you must delete them first.

There are two required SPNs for Kerberos authentication:

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

If the alias is `finance.domain.com`, the following are the two required SPNs:

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

Note

You will need to delete any existing HOST SPNs that correspond to the DNS alias on the Active Directory computer object before you create new HOST SPNs for your Amazon FSx file system's Active Directory (AD) computer object. Attempts to set SPNs for your Amazon FSx file system will fail if an SPN for the DNS alias exists in the AD.

The following procedures describes how to do the following:

- Find any existing DNS alias SPNs on the original file system's Active Directory computer object.
- Delete the existing SPNs found, if any.
- Create new DNS alias SPNs for your Amazon FSx file system's Active Directory computer object.

To install the required PowerShell Active Directory module

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

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

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

If you have SPNs configured for the DNS alias that you've assigned to another file system on a computer object in your Active Directory, you must first remove those SPNs before adding SPNs to your file system's computer object.

1. Find any existing SPNs by using the following commands. Replace *alias_fqdn* with the DNS alias that you associated with the file system in Step 1.

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

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

   - Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Step 1.
   - Replace *file_system_DNS_name* with the original file system's DNS name.

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

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

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Step 1.

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

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

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

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

     You can also get the DNS name in the response of the DescribeFileSystems API operation.
   - Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Step 1.

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

   ##Use the following command to set both the full FQDN and Alias SPNs
   Set-AdComputer -Identity $FSxAdComputer -Add @{"msDS-AdditionalDnsHostname" =
    @($Alias, $Alias.Split(".")[0])}
   ```

   Note

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

2. Verify that the new SPNs are configured for the DNS alias using the following example script. Ensure that the response includes two HOST SPNs, HOST/*alias* and HOST/*alias_fqdn*, as described previously in this procedure.

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

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

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

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Step 1.

Update or create a DNS CNAME record

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

The `dnsserver` and `activedirectory` Windows modules are required to run the commands presented in this section.

To install the required PowerShell modules

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

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide*.

2. Open PowerShell as administrator.
3. The PowerShell DNS Server module is required to perform the instructions in this procedure. Install it using the following command.

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

To update or create a custom DNS name to your Amazon FSx file system

1. Connect to your Amazon EC2 instance as a user that is a member of a group that has DNS administration permissions (**AWS Delegated Domain Name System Administrators** in AWS Managed Active Directory, and **Domain Admins** or another group to which you've delegated DNS administration permissions in your self-managed Active Directory).

   For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide*.
2. At the command prompt, run the following script. This script migrates any existing DNS CNAME records to your Amazon FSx file system. If none are found, it creates a new DNS CNAME record for the DNS alias *alias_fqdn* that resolves to the default DNS name for your Amazon FSx file system.

   To run the script:

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

   ```
   $Alias="alias_fqdn"
   $FSxDnsName="file_system_dns_name"
   $AliasHost=$Alias.Split('.')[0]
   $ZoneName=((Get-WmiObject Win32_ComputerSystem).Domain)
   $DnsServerComputerName = (Resolve-DnsName $ZoneName -Type NS | Where Type -eq 'A' |
    Select -ExpandProperty Name) | Select -First 1
   Add-DnsServerResourceRecordCName -Name $AliasHost -ComputerName
    $DnsServerComputerName -HostNameAlias $FSxDnsName -ZoneName $ZoneName
   ```
3. Repeat the previous step for each DNS alias that you associated with the file system in Step 1.

You've now added a DNS CNAME value for your Amazon FSx file system with the DNS alias. You can now use the DNS alias to access your data.

Note

When updating a DNS CNAME record to point to an Amazon FSx file system previously pointed to another file system, clients might not be able to connect with file system for a brief period of time. When the client DNS cache refreshes, they should be able to connect using the DNS alias. For more information, see Can't access the file system using a DNS alias.

Enforcing Kerberos authentication using Group Policy Objects (GPOs)

You can enforce Kerberos authentication when accessing the file system by setting the following Group Policy Objects (GPOs) in your Active Directory:

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

1. Log on to a Windows instance joined to the Active Directory to which your Amazon FSx file system is joined as an administrator. If you are configuring a self-managed Active Directory, apply these steps directly to your Active Directory.
2. Choose **Start**, choose **Administrative Tools**, and then choose **Group Policy Management**.
3. Choose **Group Policy Objects**.
4. If your Group Policy Object does not already exist, create it.
5. Locate the existing **Network Security: Restrict NTLM: Outgoing NTLM traffic to remote servers** policy. (If there is no existing policy, create a new policy.) In the **Local security setting** tab, open the context (right-click) menu, and choose **Properties**.
6. Choose **Deny all**.

7. Choose **Apply** to save the security setting.
8. To set exceptions for NTLM connections to specific remote servers for the client, locate the **Network security: Restrict NTLM: Add remote server exceptions**.

   Open the context (right-click) menu, and choose **Properties** in the **Local security setting** tab.
9. Enter the names of any servers to add to the exception list.
10. Choose **Apply** to save the security setting.

Accessing data using file shares

A Microsoft Windows *file share* is a specific folder or directory on your file system. It includes any sub folders that might exist. Clients access the file shares on your file system using the Server Message Block (SMB) protocol. Your FSx for Windows File Server file system comes with a default Windows file share, named `share`. You can create and manage as many other file shares as you want by using the Windows *Shared Folders* graphical user interface (GUI) tool.

Microsoft Windows continuously available (CA) shares provide the primary benefit of maintaining uninterrupted access to shared files even when a server node within a cluster fails. Using CA file shares can minimize interruptions to the server applications that are storing their data files on these file shares during file system maintenance windows.

For more information about creating and managing file shares on your FSx for Windows File Server file system, including CA shares, see Creating, updating, removing file shares.

Mapping file shares

To access your file shares, 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.

For more information about mapping and mounting file shares to access data on your file system, see the following procedures:

- Mapping a file share on an Amazon EC2 Windows instance.

- Mounting a file share on an Amazon EC2 Mac instance
- Mounting a file share on an Amazon EC2 Linux instance

Mapping a file share on an Amazon EC2 Windows instance

You can map a file share on an EC2 Windows instance to access your FSx for Windows File Server file system by using the Windows File Explorer or the command prompt.

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

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*.
3. After you're connected, open File Explorer.
4. In the navigation pane, open the context (right-click) menu for **Network**, and choose **Map Network Drive**.
5. For **Drive**, choose a drive letter.
6. For **Folder**, enter either the file system's DNS name or a DNS alias associated with the file system, and the share name.

   > ⚠ **Important**
   >
   > Using an IP address instead of the DNS name could result in unavailability during the failover process of the Multi-AZ file system. Also, DNS names or associated DNS aliases are required for Kerberos-based authentication in Multi-AZ and Single-AZ file systems.

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

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

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

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

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

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

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

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

```
\\amznfsxaa11bb22.ad-domain.com\share
```

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

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

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

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

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

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

2. Connect to your EC2 Windows instance as a user in your AWS Managed Microsoft AD directory. For more information, see Connecting to your Windows instance in the *Amazon EC2 User Guide*.

3. After you're connected, open a command prompt window.
4. Mount the file share using a drive letter of your choice, the file system's DNS name, and the share name. You can find the DNS name using the Amazon FSx console by choosing **Windows File Server**, **Network & security**. Or, you can find them in the response of the `CreateFileSystem` or `DescribeFileSystems` API operation.

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

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

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

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

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

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

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

Mounting a file share on an Amazon EC2 Mac instance

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

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

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

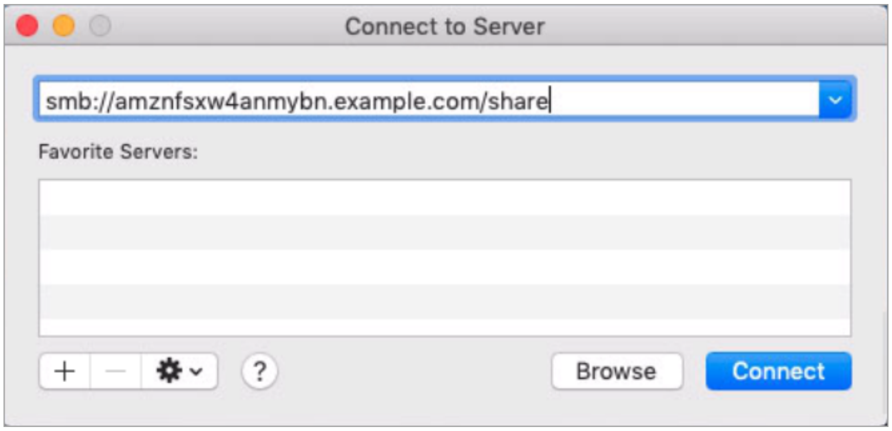
- Launch a Mac instance using the console
- Launch a Mac instance using the AWS CLI

2. Connect to your EC2 Mac instance using Virtual Network Computing (VNC). For more information, see Connect to your instance using VNC in the *Amazon EC2 User Guide*.
3. On your EC2 Mac instance, connect to your Amazon FSx file share, as follows:

   a. Open Finder, choose **Go**, and then choose **Connect to Server**.

   b. In the **Connect to Server** dialog box, enter either the file system's DNS name or a DNS alias associated with the file system, and the share name. Then choose **Connect**.

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

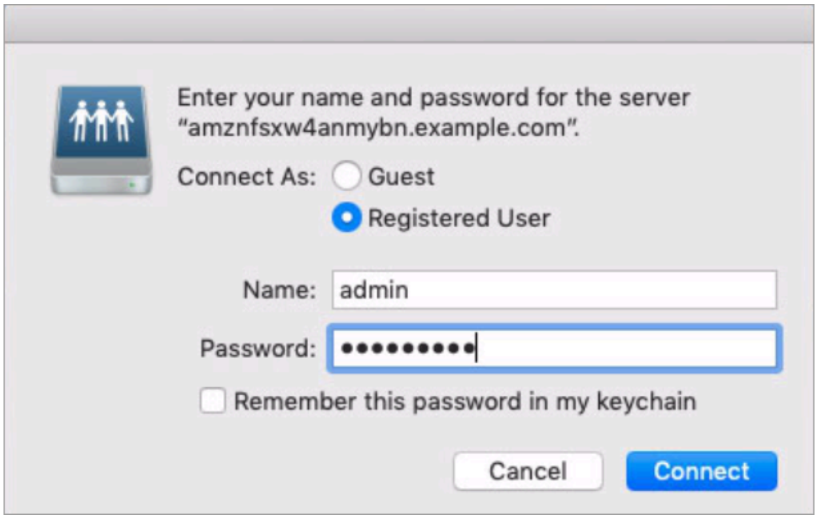


Connect to Server

smb://amznfsxw4anmybn.example.com/share

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

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



Enter your name and password for the server "amznfsxw4anmybn.example.com".

Connect As: ○ Guest
● Registered User

Name: admin

Password: ●●●●●●●●●

□ Remember this password in my keychain

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

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

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

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

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

3. Mount the file share with the following command.

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

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

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

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

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

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

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

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

Mounting a file share on an Amazon EC2 Linux instance

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

> **Note**
>
> - The following commands specify parameters such as SMB protocol, caching, and read and write buffer size as examples only. Parameter choices for the Linux `cifs` command, as well as the Linux kernel version used, can impact throughput and latency for network operations between the client and the Amazon FSx file system. For more information, see `cifs` documentation for the Linux environment you are using.
> - Linux clients do not support automatic DNS-based failover. For more information, see Failover experience on Linux clients.

To mount a file share on an Amazon EC2 Linux instance joined to an 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*.

3. Run the following command to install the `cifs-utils` package. This package is used to mount network file systems like Amazon FSx on Linux.

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

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

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

5. Authenticate with kerberos using the following command.

   ```
   $ kinit
   ```

6. Mount the file share with the following command.

   ```
   $ sudo mount -t cifs //file_system_dns_name/file_share mount_point --verbose -o
    vers=SMB_version,sec=krb5,cruid=ad_user,rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize,cache=no
   file-server-Ip
   ```

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

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

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

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

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

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

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

The output shows that the maximum buffer size is 130048.

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

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

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

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

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

The output shows that the maximum buffer size is 130048.

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

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

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

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

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

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

To launch and configure the Amazon Linux EC2 instance

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

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

4. Create the mount point **`/mnt/fsxx`** where you plan to mount the Amazon FSx file system.

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

5. Create the `creds.txt` credentials file in the `/home/ec2-user` directory, using the format shown previously.

6. Set the `creds.txt` file permissions so that only you (the owner) can read and write to the file by running the following command.

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

To mount the file system

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

2. Mount the file system using the following command:

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

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

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

   The output shows that the maximum buffer size is 130048.

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

   ```
   $ mount -l -t cifs
   ```

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

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 you're mounting.
- `-t cifs vers=SMB_version` – Specifies the type of file system as CIFS and the SMB protocol version. Amazon FSx for Windows File Server supports SMB versions 2.0 through 3.1.1.
- `sec=ntlmsspi` – Specifies to use NT LAN Manager Security Support Provider Interface (NTLMSSPI) for authentication.
- `cache=cache_mode` – Sets the cache mode. This option for CIFS cache can impact performance, and you should test which settings work best (and review Linux documentation) for your kernel and workload. Options `strict` and none are recommended, because `loose` can cause data inconsistency due to the looser protocol semantics.
- `cred=/home/ec2-user/creds.txt` – Specifies where to get the user credentials.
- `/mnt/fsx` – Specifies the mount point for the Amazon FSx file share on your EC2 instance.
- `rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize` – Specifies the read and write buffer size as the maximum allowed by the CIFS protocol. Replace `CIFSMaxBufSize` with the largest value allowed by your kernel. Determine the `CIFSMaxBufSize` by running the following command.

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

Automatically mount file shares on an Amazon EC2 Linux instance

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

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

The following procedure uses Microsoft NTLM authentication. You mount the file system as a user that is a member of the Microsoft Active Directory domain to which the FSx for Windows File Server file system is joined. You can retrieve the credentials for the user account from the `creds.txt` file using the following command.

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

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

To launch and configure the Amazon Linux EC2 instance

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

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

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

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

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

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

To automatically mount the file system

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

2. To automatically mount the file share using its private IP address, add the following line to the `/etc/fstab` file.

   ```
   //file-system-IP-address/file_share /mnt/fsx cifs
    vers=SMB_version,sec=ntlmsspi,cred=/home/ec2-user/
   creds.txt,rsize=CIFSMaxBufSize,wsize=CIFSMaxBufSize,cache=none 0 0
   ```

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

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

   The output shows that the maximum buffer size is 130048.

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

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

4. To mount the file share, reboot the Amazon EC2 instance.

5. When the instance is available again, verify that the file system is mounted by running the following command.

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

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

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

Creating, updating, removing file shares

This topic describes how you can manage file shares by performing the following tasks.

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

You can use the Windows-native Shared Folders GUI and the Amazon FSx CLI for remote management on PowerShell to manage file shares on your FSx for Windows File Server file system. You might experience delays when using the Shared Folder GUI (**fsmgmt.msc**) when first opening the context menu for shares located on a different file system. To avoid these delays, use PowerShell to manage file shares that are located on multiple file systems.

Microsoft Windows enforces rules and limitations for naming files and directories. To ensure that you can successfully create and access your data, you should name your files and directories according to these Windows guidelines. For more information, see Naming Conventions.

⚠ **Warning**

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

Managing file shares with the Shared Folders GUI

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 describe how to manage your file shares.

To connect shared folders to your FSx for Windows File Server file system

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

To find your file system's DNS name on the Amazon FSx console, choose **File systems**, choose your file system, and then check the **Network & Security** section of the file system details page. You can also get the DNS name in the response of the DescribeFileSystems API operation.

6. Choose **OK**. An entry for your Amazon FSx file system then appears in the list for the Shared Folders tool.

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

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

  You have to create the local folder prior to creating the new file share. You can do so as follows:

  - Using the Shared Folders tool: click on "Browse" when specifying local folder path and click on "Make new folder" to create the local folder.
  - Using command line:

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

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

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

Note

If you are using the **fsmgmt.msc** Shared Folders GUI tool to access shares located on multiple FSx for Windows File Server file systems, you may experience delays when first opening the file share context menu for a share that is located on a different file system. To avoid these delays, you can manage file shares using PowerShell as described below.

Managing file shares with PowerShell

You can manage file shares using custom FSx for Windows File Server remote-management commands for PowerShell. These commands can help you to automate managing file share tasks such as:

- Migrating file shares from existing file servers to Amazon FSx
- Synchronizing file shares across AWS Regions for disaster recovery
- Programmatically managing ongoing file shares workflows, such as team file-share provisioning

To learn how to use the Amazon FSx CLI for remote management on PowerShell, see Using the Amazon FSx CLI for PowerShell.

The following table lists the Amazon FSx CLI remote management PowerShell commands that you can use to manage file shares on FSx for Windows File Server file systems.

Share Management Command	Description
New-FSxSmbShare	Creates a new file share.
Remove-FSxSmbShare	Removes a file share.
Get-FSxSmbShare	Retrieves existing file shares.
Set-FSxSmbShare	Sets properties for a share.
Get-FSxSmbShareAccess	Retrieves the access control list (ACL) of a share.

Share Management Command	Description
Grant-FSxSmbShareAccess	Adds an allow access control entry (ACE) for a trustee to the security descriptor of a share.
Revoke-FSxSmbShareAccess	Removes all of the allow ACEs for a trustee from the security descriptor of a share.
Block-FSxSmbShareAccess	Adds a deny ACE for a trustee to the security descriptor of a share.
Unblock-FSxSmbShareAccess	Removes all of the deny ACEs for a trustee from the security descriptor of a share.

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

Passing credentials to New-FSxSmbShare

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

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

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

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

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

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

To create a continuously available (CA) share

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

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

You can modify the `-ContinuouslyAvailable` option on an existing file share using the `Set-FSxSmbShare` command.

Determine if an existing file share is continuously available

Use the following command to view the value of the Continuously Available property for an existing file share.

```
Invoke-Command -ComputerName powershell_endpoint -ConfigurationName FSxRemoteAdmin -
scriptblock { get-fsxsmbshare -name share_name }
```

If CA is enabled, the output will include the following line:

```
[...]
ContinuouslyAvailable : True
[...]
```

If CA is not enabled, the output will include the following line:

```
[...]
ContinuouslyAvailable : False
[...]
```

To enable Continuously Available on an existing file share, use the following command:

```
Invoke-Command -ComputerName powershell_endpoint -ConfigurationName FSxRemoteAdmin -
scriptblock { set-fsxsmbshare -name share_name -ContinuouslyAvailable $True}
```

New-FSxSmbShare command fails with a one-way trust

Amazon FSx does not support executing the `New-FSxSmbShare` PowerShell command in cases where you have a one-way trust and the domain in which the user resides is not configured to trust the domain associated with Amazon FSx file system.

You can resolve this situation using one of following solutions:

- The user executing the `New-FSxSmbShare` command needs to be in the same domain as the FSx file system.
- You can use the fsmgmt.msc GUI to create shares on your file system. For more information, see Managing file shares with the Shared Folders GUI.

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. The following sections provide information to help you choose the right deployment type for your workloads. For information on the service's availability SLA (Service Level Agreement), see Amazon FSx Service Level Agreement.

Single-AZ file systems are composed of a single Windows file server instance and a set of storage volumes within a single Availability Zone (AZ). With Single-AZ file systems, data is automatically replicated to protect it from the failure of a single component in most cases. Amazon FSx continuously monitors for hardware failures, and automatically recovers from failure events by replacing the failed infrastructure component. Single-AZ file systems usually experience about 30 minutes of downtime during failure recovery events, and during the planned maintenance window that you configure for your file system. With Single-AZ file systems, file system failure may be unrecoverable in rare cases, such as due to multiple component failures or due to a non-graceful failure of the single file server that leaves the file system in an inconsistent state, in which case you can recover your file system from the most recent backup.

Multi-AZ file systems are composed of a high-availability cluster of Windows file servers spread across two AZs (a preferred AZ and a standby AZ), leveraging Windows Server Failover Clustering (WSFC) technology and a set of storage volumes on each of the two AZs. Data is replicated synchronously within each individual AZ and between the two AZs. Relative to Single-AZ deployment, Multi-AZ deployments provide enhanced durability by further replicating data across AZs, and enhanced availability during planned system maintenance and unplanned service disruption by failing over automatically to the standby AZ. This allows you to continue accessing your data, and helps to protect your data against instance failure and AZ disruption.

Choosing Single-AZ or Multi-AZ file system deployment type

We recommend using Multi-AZ file systems for most production workloads given the high availability and durability model it provides. Single-AZ deployment is designed as a cost-efficient solution for test and development workloads, certain production workloads that have replication built into the application layer and do not require additional storage-level redundancy, and production workloads that have relaxed availability and Recovery Point Objective (RPO) needs. Workloads with relaxed availability and RPO needs can tolerate temporary loss of availability for up to 20 minutes in the event of planned file system maintenance or unplanned service disruption and, in rare cases, the loss of data updates since the most recent backup.

We also recommend reviewing the availability model for your file system and ensuring that your workload is resilient to the expected recovery behavior for the deployment type you choose during events such as file system maintenance, throughput capacity changes, and unplanned service disruptions.

Feature support by deployment type

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

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

Note

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

Failing over process

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

- An Availability Zone outage occurs.

- The preferred file server becomes unavailable.
- The preferred file server undergoes planned maintenance.

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

During the brief period in which your file system is failing over and failing back, I/O may be paused and Amazon CloudWatch metrics may be temporarily unavailable. For Multi-AZ file systems, any file read and write activity that occurs during failover and failback will need to be synchronized between the primary and secondary file servers. This process can take up to multiple hours for file systems with HDD storage, and for workloads that are write-heavy and IOPS-heavy. We recommend testing the impact of failovers on your application while your file system is under a lighter load.

Failover experience on Windows clients

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

Failover experience on Linux clients

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

Testing failover on a file system

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

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

Single-AZ and Multi-AZ file system resources

Single-AZ and Multi-AZ file systems consume subnets and elastic network interfaces differently, as explained in the following sections.

Subnets

When you create a virtual private cloud (VPC), it spans all the Availability Zones (AZs) in the AWS 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. A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

FSx for Windows File Server Single-AZ file systems require one subnet which you specify at creation. The subnet you choose defines the Availability Zone in which the file system gets created.

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

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

File system elastic network interfaces

An elastic network interfaces is a logical networking component in a VPC that represents a virtual network card. When you create an Amazon FSx file system, Amazon FSx provisions one or more elastic network interface in the VPC that you associate with your file system. The elastic network interface enables clients to communicate with and mount the file system. The elastic network interface is considered to be within the service scope of Amazon FSx, despite it 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**

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

The following table summarizes resource utilization for FSx for Windows File Server Single-AZ and Multi-AZ file systems:

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

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

⚠ **Important**

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

Working with Microsoft Active Directory

When you create an FSx for Windows File Server file system, you join it to your Active Directory domain to provide user authentication and file- and folder-level access control. Amazon FSx works with Microsoft Active Directory to integrate with your existing Microsoft Windows environments. Amazon FSx provides two options using your FSx for Windows File Server file system with Active Directory: Using Amazon FSx with AWS Directory Service for Microsoft Active Directory and Using a self-managed Microsoft Active Directory.

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.

Your users can then use their existing user identities in Active Directory to authenticate themselves and access the FSx for Windows File Server 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 along with their security access control list (ACL) configuration to Amazon FSx without any modifications.

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

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

- Service user credentials
- DNS server IP addresses

You *cannot* change the following properties for your joined Microsoft AD after you've created the file system:

- DomainName
- OrganizationalUnitDistinguishedName
- FileSystemAdministratorsGroup

However, you can create a new file system from a backup and change these properties in the new file system's Microsoft Active Directory integration configuration. For more information, see Restoring backups to new file system.

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

Your FSx for Windows File Server may become **Misconfigured** if there is a change in your Active Directory configuration that disrupts the connection to your file system. To return your file system to the **Available** state, select the **Attempt Recovery** button in the Amazon FSx console, or use the `StartMisconfiguredStateRecovery` command in the Amazon FSx API or console. For more information see File system is in a misconfigured state.

Topics

- Using Amazon FSx with AWS Directory Service for Microsoft Active Directory
- Using a self-managed Microsoft Active Directory

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 directories in the cloud. You can use these Active Directory directories in your workload deployment.

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

To use Amazon FSx with your AWS Managed Microsoft AD setup, you can use the Amazon FSx console. When you create a new FSx for Windows File Server file system in the console, choose **AWS Managed Active Directory** under the **Windows Authentication** section. You also choose the specific directory that you want to use. For more information, see Step 5. Create your file system.

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 Active Directory domain. For more information, see Using a self-managed Microsoft Active Directory.

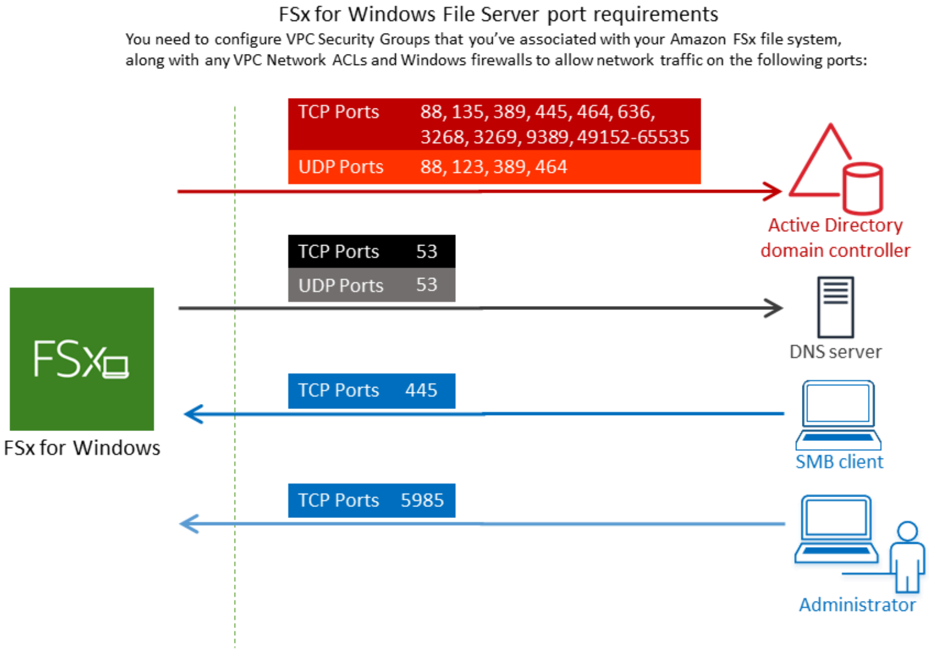
Additionally, you can also set up your system to benefit from a resource forest isolation model. In this model, you isolate your resources, including your Amazon FSx file systems, into a separate Active Directory forest from the one where your users are.

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

Networking prerequisites

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

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



FSx for Windows File Server port requirements

You need to configure VPC Security Groups that you've associated with your Amazon FSx file system, along with any VPC Network ACLs and Windows firewalls to allow network traffic on the following ports:

TCP Ports 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535
UDP Ports 88, 123, 389, 464
Active Directory domain controller
TCP Ports 53
UDP Ports 53
DNS server
FSx
FSx for Windows
TCP Ports 445
SMB client
TCP Ports 5985
Administrator

The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Doma Name Syste (DNS)
TCP/UDP	88	Kerbe authe ation
TCP/UDP	464	Chang Se t passw

Protocol	Ports	Role
TCP/UDP	389	Lightw ht Direct Access Protoc (LDAP
UDP	123	Netwo Time Protoc (NTP)
TCP	135	Distrib ed Comp Enviro nt / End Point Mapp (DCE / EPMA
TCP	445	Direct Servic SMB file sharin

Protocol	Ports	Role
TCP	636	Lightw ht Direct Access Protoc over TLS/ SSL (LDAP
TCP	3268	Micros Globa Catalo
TCP	3269	Micros Globa Catalo over SSL
TCP	5985	WinRM 2.0 (Micro t Windo Remo Manag t)

Protocol	Ports	Role
TCP	9389	Micros AD DS Web Servic Power l
TCP	49152 - 65535	Epher ports for RPC

Important

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

Note

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

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

Important

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

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

Using a resource forest isolation model

You join your file system to an AWS Managed Microsoft AD setup. You then establish a one-way forest trust relationship between an AWS Managed Microsoft AD domain that you create and your existing self-managed Active Directory 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 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 AWS managed domain. In a Kerberos authentication scenario, authentication requests originating from a corporate client get validated by the corporate domain, which then refers it to the AWS Managed Microsoft AD, and eventually the client presents its service ticket to your FSx for Windows File Server file system. For more information about trusts, see the post Everything you wanted to know about trusts with AWS Managed Microsoft AD in the AWS Security Blog.

Test your Active Directory configuration

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

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

- What is Directory Service in the *AWS Directory Service Administration Guide*
- Create your AWS Managed Active Directory in the *AWS Directory Service Administration Guide*
- When to Create a Trust Relationship in the *AWS Directory Service Administration Guide*

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

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

> **Note**
>
> You can only select an AWS Managed Microsoft AD within the same AWS Region as your file system. If you want to use a cross-Region VPC peering setup, you should use a self-managed Microsoft Active Directory. For more information, see Using a self-managed Microsoft Active Directory.

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

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

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

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

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

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

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

Validating connectivity to your Active Directory domain controllers

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

To validate connectivity to your Active Directory domain controllers

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

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

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

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

```
PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
```

5. Download the network validation tool using the following command.

    ```
    PS C:\> Invoke-WebRequest "https://docs.aws.amazon.com/fsx/latest/WindowsGuide/
    samples/AmazonFSxADValidation.zip" -OutFile "AmazonFSxADValidation.zip"
    ```

6. Expand the zip file by using the following command.

    ```
    PS C:\> Expand-Archive -Path "AmazonFSxADValidation.zip"
    ```

7. Add the AmazonFSxADValidation module to the current session.

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

8. Set the value for the Active Directory domain controller IP address and run the connectivity test using the following commands:

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

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

    ```
    PS C:\AmazonFSxADValidation> $Result

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


    PS C:\AmazonFSxADValidation> $Result.TcpDetails

    Port Result    Description
    ---- ------    -----------
    ```

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

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

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

PS C:\AmazonFSxADValidation> $Result

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


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

Windows socket error code mapping

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

> **Note**
>
> As an alternative to the above procedure, you can use the `AWSSupport-ValidateFSxWindowsADConfig` runbook to validate your self-managed Active Directory configuration. For more information, see `AWSSupport-ValidateFSxWindowsADConfig` in the *AWS Systems Manager Automation runbook reference*.

Using a self-managed Microsoft Active Directory

If your organization manages identities and devices using a self-managed Active Directory on-premises or in the cloud, you can join an FSx for Windows File Server file system to your Active Directory domain at creation.

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

> **Note**
>
> You can isolate your resources—including your Amazon FSx file systems—into a separate Active Directory forest from the one where your users reside. To do this, join your file system to an AWS Managed Microsoft Active Directory and establish a one-way forest trust relationship between an AWS Managed Microsoft Active Directory that you create and your existing self-managed Active Directory.

- User name and password for a service account on your Active Directory domain, for Amazon FSx to use to join the file system to your Active Directory domain. You can provide these credentials as plaintext or store them in AWS Secrets Manager and provide the secret ARN (recommended).
- (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 Access Control Lists (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 Active Directory domain by default.

> **Note**
>
> The domain group name you provide must be unique in your Active Directory. FSx for Windows File Server will not create the domain group under the following circumstances:
>
> - If a group already exists with the name you specify
> - If you do not specify a name, and a group named "Domain Admins" already exists in your Active Directory.

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

Topics

- Prerequisites
- Service account permissions
- Best practices when using a self-managed Active Directory
- Amazon FSx service account
- Delegating permissions to the Amazon FSx service account or group
- Validating your Active Directory configuration
- Joining an Amazon FSx file system to a self-managed Microsoft Active Directory domain
- Getting the correct file system IP addresses to use for manual DNS entries
- Updating a self-managed Active Directory configuration
- Changing the Amazon FSx service account
- Monitoring self-managed Active Directory updates

Prerequisites

Before you join an FSx for Windows File Server file system to your self-managed Microsoft Active Directory domain, review the following prerequisites to help ensure that you can successfully join your Amazon FSx file system to your self-managed Active Directory.

On-premises configurations

These are the prerequisites for your self-managed Microsoft Active Directory, either an on-premises or cloud-based, that you will join the Amazon FSx file system to.

- The Active Directory domain controllers:
  - Must have a domain functional level at Windows Server 2008 R2 or higher.
  - Must be writable.
  - At least one of the reachable domain controllers must be a Global Catalog of the forest.
- The DNS server must be able to resolve names as follows:
  - In the domain that you are joining the file system
  - In the root domain of the forest
- The DNS server and Active Directory domain controller IP addresses must meet the following requirements, which vary depending on when your Amazon FSx file system was created:

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

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

- The domain name of your self-managed Active Directory must meet the following requirements:
  - The domain name isn't in Single Label Domain (SLD) format. Amazon FSx doesn't support SLD domains.

  - For Single-AZ 2 and all Multi-AZ file systems, the domain name cannot exceed 47 characters.
- Any Active Directory sites that you have defined must meet the following prerequisites:
  - The subnets in the VPC that's associated with your file system must be defined in an Active Directory site.
  - There are no conflicts between the VPC subnets and any of the Active Directory site subnets.

Amazon FSx requires connectivity to the domain controllers or Active Directory sites you have defined in your Active Directory environment. Amazon FSx will ignore any domain controllers with TCP and UDP blocked on port 389. For the remaining domain controllers in your Active Directory, ensure that they meet the Amazon FSx connectivity requirements. Additionally, verify that any changes to your service account are propagated to all these domain controllers.

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

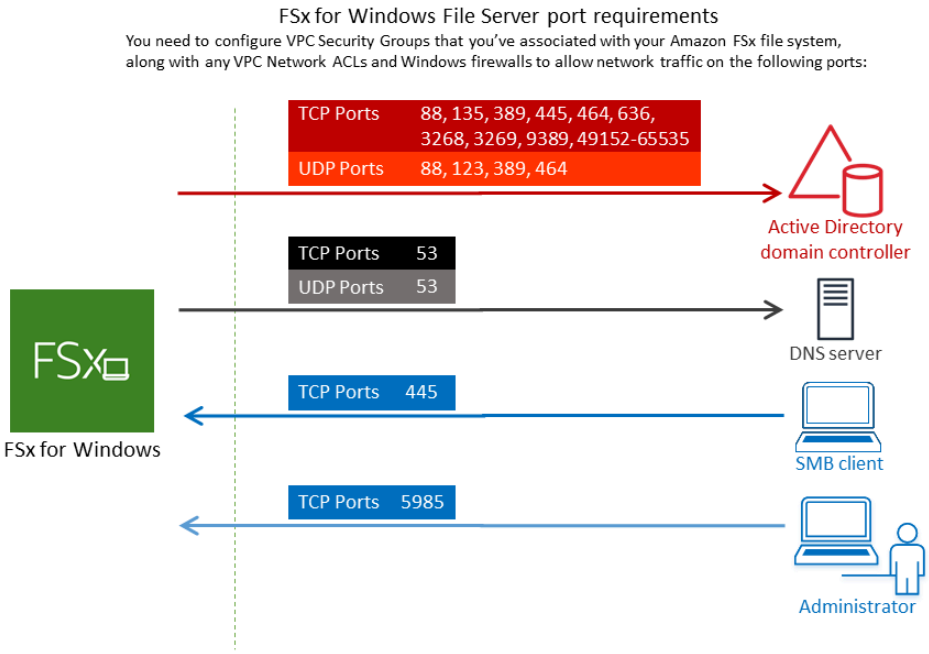
You can validate your Active Directory configuration, including testing connectivity of multiple domain controllers, using the Amazon FSx Active Directory Validation tool. To limit the number of domain controllers that require connectivity, you can also build a trust relationship between your on-premise domain controllers and AWS Managed Microsoft AD. For more information, see Using a resource forest isolation model.

> ⚠ **Important**
>
> Amazon FSx only registers the DNS records for a file system if you are using Microsoft DNS as the default DNS service. If you are using a third-party DNS, you will need to manually set up DNS record entries for your file system after you create it.

Network configurations

This section describes the network configuration requirements for joining a file system to your self-managed Active Directory. We strongly recommend that you use the Amazon FSx Active Directory validation tool to test your network settings before attempting to join your file system to your self-managed Active Directory.

- Ensure that your firewall rules will allow ICMP traffic between your Active Directory domain controllers and Amazon FSx.
- Connectivity must be configured between the Amazon VPC where you want to create the file system and your self-managed Active Directory. You can set up this connectivity using Direct Connect, AWS Virtual Private Network, VPC peering, or AWS Transit Gateway.
- The default VPC security group for your default Amazon VPC must be added to your file system using the Amazon FSx console. Ensure that the security group and the VPC Network ACLs for the subnets where you create your file system allow traffic on the ports and in the direction shown in the following diagram.


FSx for Windows File Server port requirements
You need to configure VPC Security Groups that you've associated with your Amazon FSx file system, along with any VPC Network ACLs and Windows firewalls to allow network traffic on the following ports:
TCP Ports 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535
UDP Ports 88, 123, 389, 464
Active Directory domain controller
TCP Ports 53
UDP Ports 53
DNS server
FSx
FSx for Windows
TCP Ports 445
SMB client
TCP Ports 5985
Administrator

The following table identifies the protocol, ports, and its role.

Protocol	Ports	Role
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Kerberos authentication
TCP/UDP	464	Change/set password

Protocol	Ports	Role
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
UDP	123	Network Time Protocol (NTP)
TCP	135	Distributed Computing Environment/End Point Mapper (DCE/EPMAP)
TCP	445	Directory Services SMB file sharing
TCP	636	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)
TCP	3268	Microsoft Global Catalog
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management)
TCP	9389	Microsoft Active Directory DS Web Services, PowerShell **Important** Allowing outbound traffic on TCP port 9389 is required for Single-AZ 2 and Multi-AZ file system deployments.
TCP	49152 - 65535	Ephemeral ports for RPC

These traffic rules need to also be mirrored on the firewalls that apply to each of the Active Directory domain controllers, DNS servers, FSx clients, and FSx administrators.

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

⚠ **Important**

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

Service account permissions

You need to have a service account in your self-managed Microsoft Active Directory with delegated permissions to join computer objects to your self-managed Active Directory domain. A *service account* is a user account in your self-managed Active Directory that has been delegated certain tasks.

The following is the minimum set of permissions that must be delegated to the Amazon FSx service account in the OU that you're joining the file system to.

- If using *Delegate Control* in the Active Directory User and Computers MMC:
  - Reset passwords
  - Read and write Account Restrictions
  - Validated write to DNS host name
  - Validated write to service principal name
- If using *Advanced Features* in the Active Directory User and Computers MMC:
  - Modify permissions
  - Create computer objects
  - Delete computer objects

For more information, see the Microsoft Windows Server documentation topic Error: Access is denied when non-administrator users who have been delegated control try to join computers to a domain controller.

For more information about setting the required permissions, see Delegating permissions to the Amazon FSx service account or group.

Best practices when using a self-managed Active Directory

Topics

- Storing Active Directory credentials using AWS Secrets Manager

We recommend that you follow these best practices when joining an Amazon FSx for Windows File Server file systems to your self-managed Microsoft Active Directory. These best practices will help you in maintaining continuous, uninterrupted availability of your file system.

Use a separate service account for Amazon FSx

Use a separate service account to delegate the required privileges for Amazon FSx to fully manage file systems that are joined to your self-managed Active Directory. We do not recommend using the **Domain Admins** for this purpose.

Use an Active Directory group

Use an Active Directory group to manage Active Directory permissions and configurations associated with the Amazon FSx service account.

Segregate the Organizational Unit (OU)

To make it easier to find and manage your Amazon FSx computer objects, we recommend that you segregate the Organizational Unit (OU) you use for your FSx for Windows File Server file systems from other domain controller concerns.

Keep the Active Directory configuration up-to-date

It is imperative that you keep your file system's Active Directory configuration up-to-date with any changes. For example, if your self-managed Active Directory uses a time-based password reset policy, as soon as the password is reset, make sure to update the service account password on your file system. For more information, see Updating a self-managed Active Directory configuration.

Changing the Amazon FSx service account

If you update your file system with a new service account, it must have the required permissions and privileges to join your Active Directory and have **Full control** permissions for the existing computer objects associated with the file system. For more information, see Changing the Amazon FSx service account.

Assign subnets to a single Microsoft Active Directory site

If your Active Directory environment has a large number of domain controllers, use **Active Directory Sites and Services** to assign the subnets used by your Amazon FSx file systems to a single Active Directory site with the highest availability and reliability. Make sure that the VPC security group, VPC network ACL, Windows firewall rules on your DCs, and any other network routing controls you have in your Active Directory infrastructure allow communication from Amazon FSx on the required ports. This allows Windows to revert to other domain controllers if it can't use the assigned Active Directory site. For more information, see File system access control with Amazon VPC.

Use security group rules to limit traffic

Use security group rules to implement the principle of least privilege in your virtual private cloud (VPC). You can limiting the type of inbound and outbound network traffic allowed for your file using VPC security group rules. For example, we recommend only allowing outbound traffic to your self-managed Active Directory domains controllers or to within the subnet or security group you are using. For more information, see File system access control with Amazon VPC.

Do not move computer objects created Amazon FSx

⚠ **Important**

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

Validate your Active Directory configuration

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

Storing Active Directory credentials using AWS Secrets Manager

You can use AWS Secrets Manager to securely store and manage your Microsoft Active Directory domain join service account credentials. This approach eliminates the need to store sensitive credentials in plaintext in application code or configuration files, strengthening your security posture.

You can also configure IAM policies to manage access to your secrets, and set up automatic rotation policies for your passwords.

Store Active Directory credentials in AWS Secrets Manager (Console)

Step 1: Create a KMS key

Create a KMS key to encrypt and decrypt your Active Directory credentials in Secrets Manager.

To create a key

> **Note**
>
> For **Encryption Key**, create a new key, don't use the AWS default KMS key. Be sure to create the AWS KMS key in the same Region that contains the file system that you want to join to your Active Directory.

1. Open the AWS KMS console at https://console.aws.amazon.com/kms.
2. Choose **Create key**.
3. For **Key Type**, choose **Symmetric**.
4. For **Key Usage**, choose **Encrypt and decrypt**.
5. For **Advanced options**, do the following:

   a. For **Key material origin**, choose **KMS**.

   b. For **Regionality**, choose **Single-Region key** and choose **Next**.
6. Choose **Next**.
7. For **Alias**, provide a name for the KMS key.
8. (Optional) For **Description**, provide a description of the KMS key.
9. (Optional) For **Tags**, provide a tag for the KMS key and choose **Next**.
10. (Optional) For **Key administrators**, provide the IAM users and roles authorized to manage this key.
11. For **Key deletion**, keep the box selected for **Allow key administrators** to delete this key and choose **Next**.
12. (Optional) For **Key users**, provide the IAM users and roles authorized to use this key in cryptographic operations. Choose **Next**.
13. For **Key policy**, choose **Edit** and include the following to the policy **Statement** to allow Amazon FSx to use the KMS key and choose **Next**. Make sure to replace the *`us-west-2`* to the AWS Region where the file system is deployed and *`123456789012`* to your AWS account ID.

```
{
    "Sid": "Allow FSx to use the KMS key",
    "Version": "2012-10-17",
    "Effect": "Allow",
    "Principal": {
        "Service": "fsx.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:kms:us-west-2:123456789012:key:*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:*",
            "kms:ViaService": "secretsmanager.us-west-2.amazonaws.com",
            "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:fsx:us-west-2:123456789012:file-system/*"
        }
    }
}
```

14. Choose **Finish**.

> **Note**
>
> You can set more granular access control by modifying the `Resource` and `aws:SourceArn` fields to target specific secrets and file systems.

Step 2: Create an AWS Secrets Manager secret

To create a secret

1. Open the Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
2. Choose **Store a new secret**.
3. For **Secret type**, choose **Other type of secret**.

4. For **Key/value pairs**, do the following to add your two keys:

    a. For the first key, enter `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`.

    b. For the value of the first key, enter only the username (without the domain prefix) of the AD user.

    c. For the second key, enter `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD`.

    d. For the value of the second key, enter the password that you created for the AD user on your domain.

5. For **Encryption key**, enter the ARN of the KMS key that you created in a previous step and choose **Next**.

6. For **Secret name**, enter a descriptive name that helps you find your secret later.

7. (Optional) For **Description**, enter a description for the secret name.

8. For **Resource permission**, choose **Edit**.

    Add the following policy to the permission policy to allow Amazon FSx to use the secret, then choose **Next**. Make sure to replace the *us-west-2* to the AWS Region where the file system is deployed and *123456789012* to your AWS account ID.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "fsx.amazonaws.com"
            },
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:123456789012:secret:*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:fsx:us-west-2:123456789012:file-
system/*"
                }
            }
        }
    ]
}
```

> **Note**
>
> You can set more granular access control by modifying the `Resource` and `aws:SourceArn` fields to target specific secrets and file systems.

9. (Optional) You can configure Secrets Manager to rotate your credentials automatically. Choose **Next**.
10. Choose **Finish**.

Store Active Directory credentials in AWS Secrets Manager (CLI)

Step 1: Create a KMS key

Create a KMS key to encrypt and decrypt your Active Directory credentials in Secrets Manager.

To create a KMS key, use the AWS CLI command create-key.

In this command, set the `--policy` parameter to specify the key policy that defines permissions for the KMS key. The policy must include the following:

- The service principal for Amazon FSx, which is `fsx.amazonaws.com`.
- Required KMS actions: `kms:Decrypt` and `kms:DescribeKey`.
- Resource ARN pattern for your AWS Region and account.
- Condition keys that restrict key usage:
  - `kms:ViaService` to ensure requests come through Secrets Manager.
  - `aws:SourceAccount` to limit to your account.
  - `aws:SourceArn` to restrict to specific Amazon FSx file systems.

The following example creates a symmetric encryption KMS key with a policy that allows Amazon FSx to use the key for decryption and key description operations. The command automatically retrieves your AWS account ID and Region, then configures the key policy with these values to ensure proper access controls between Amazon FSx, Secrets Manager, and the KMS key. Make sure your AWS CLI environment is in the same region as the file system that will join the Active Directory.

```
# Set region and get Account ID
REGION=${AWS_REGION:-$(aws configure get region)}
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)

# Create Key
KMS_KEY_ARN=$(aws kms create-key --policy "{
  \"Version\": \"2012-10-17\",
  \"Statement\": [
    {
      \"Sid\": \"Enable IAM User Permissions\",
      \"Effect\": \"Allow\",
      \"Principal\": {
        \"AWS\": \"arn:aws:iam::$ACCOUNT_ID:root\"
      },
      \"Action\": \"kms:*\",
      \"Resource\": \"*\"
    },
    {
      \"Sid\": \"Allow FSx to use the KMS key\",
      \"Effect\": \"Allow\",
      \"Principal\": {
        \"Service\": \"fsx.amazonaws.com\"
      },
      \"Action\": [
        \"kms:Decrypt\",
        \"kms:DescribeKey\"
      ],
      \"Resource\": \"*\",
      \"Condition\": {
        \"StringEquals\": {
          \"kms:ViaService\": \"secretsmanager.$REGION.amazonaws.com\",
          \"aws:SourceAccount\": \"$ACCOUNT_ID\"
        },
        \"ArnLike\": {
          \"aws:SourceArn\": \"arn:aws:fsx:$REGION:$ACCOUNT_ID:file-system/*\"
        }
      }
    }
  ]
}" --query 'KeyMetadata.Arn' --output text)
```

```
echo "KMS Key ARN: $KMS_KEY_ARN"
```

> **Note**
>
> You can set more granular access control by modifying the `Resource` and `aws:SourceArn` fields to target specific secrets and file systems.

Step 2: Create an AWS Secrets Manager secret

To create a secret for Amazon FSx to access your Active Directory, use the AWS CLI command create-secret and set the following parameters:

- `--name`: The identifier for your secret.
- `--description`: A description of the secret's purpose.
- `--kms-key-id`: The ARN of the KMS key you created in Step 1 for encrypting the secret at rest.
- `--secret-string`: A JSON string containing your AD credentials in the following format:
  - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`: Your AD service account username without the domain prefix, such as `svc-fsx`. **Don't** provide the domain prefix, such as `CORP\svc-fsx`.
  - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD`: Your AD service account password.
- `--region`: The AWS Region where your Amazon FSx file system will be created. This defaults to your configured region if `AWS_REGION` is not set.

After creating the secret, attach a resource policy using the put-resource-policy command, and set the following parameters:

- `--secret-id`: The name or ARN of the secret to attach the policy to. The following example uses **`FSxSecret`** as the `--secret-id`.
- `--region`: The same AWS Region as your secret.
- `--resource-policy`: A JSON policy document that grants Amazon FSx permission to access the secret. The policy must include the following:
  - The service principal for Amazon FSx, which is **`fsx.amazonaws.com`**.
  - Required Secrets Manager actions: `secretsmanager:GetSecretValue` and `secretsmanager:DescribeSecret`.

- Resource ARN pattern for your AWS Region and account.
- The following condition keys that restrict access:
  - `aws:SourceAccount` to limit to your account.
  - `aws:SourceArn` to restrict to specific Amazon FSx file systems.

The following example creates a secret with the required format and attaches a resource policy that allows Amazon FSx to use the secret. This example automatically retrieves your AWS account ID and Region, then configures the resource policy with these values to ensure proper access controls between Amazon FSx and the secret.

Make sure to replace the `KMS_KEY_ARN` with the ARN from the key you created in Step 1, `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`, and `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD` with your Active Directory service account credentials. Additionally, verify that your AWS CLI environment is configured for the same region as the file system that will join the Active Directory.

```
# Set region and get account ID
REGION=${AWS_REGION:-$(aws configure get region)}
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)

# Replace with your KMS key ARN from Step 1
KMS_KEY_ARN="arn:aws:kms:us-east-2:123456789012:key/1234542f-d114-555b-9ade-
fec3c9200d8e"

# Replace with your Active Directory credentials
AD_USERNAME="Your_Username"
AD_PASSWORD="Your_Password"

# Create the secret
SECRET_ARN=$(aws secretsmanager create-secret \
  --name "FSxSecret" \
  --description "Secret for FSx access" \
  --kms-key-id "$KMS_KEY_ARN" \
  --secret-string "{\"CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME\":\"$AD_USERNAME\",
\"CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD\":\"$AD_PASSWORD\"}" \
  --region "$REGION" \
  --query 'ARN' \
  --output text)

echo "Secret created with ARN: $SECRET_ARN"
```

```
# Attach the resource policy with proper formatting
aws secretsmanager put-resource-policy \
  --secret-id "FSxSecret" \
  --region "$REGION" \
  --resource-policy "{
    \"Version\": \"2012-10-17\",
    \"Statement\": [
      {
        \"Effect\": \"Allow\",
        \"Principal\": {
          \"Service\": \"fsx.amazonaws.com\"
        },
        \"Action\": [
          \"secretsmanager:GetSecretValue\",
          \"secretsmanager:DescribeSecret\"
        ],
        \"Resource\": \"$SECRET_ARN\",
        \"Condition\": {
          \"StringEquals\": {
            \"aws:SourceAccount\": \"$ACCOUNT_ID\"
          },
          \"ArnLike\": {
            \"aws:SourceArn\": \"arn:aws:fsx:$REGION:$ACCOUNT_ID:file-system/*\"
          }
        }
      }
    ]
  }"

echo "Resource policy attached successfully"
```

Note

You can set more granular access control by modifying the `Resource` and `aws:SourceArn` fields to target specific secrets and file systems.

Amazon FSx service account

Amazon FSx file systems that are joined to a self-managed Active Directory require a valid service account throughout their lifetime. Amazon FSx uses the service account to fully manage your file systems and perform administrative tasks that require unjoining and rejoining computer objects to your Active Directory domain. These tasks include replacing a failed file server and patching Microsoft Windows Server software. For Amazon FSx to perform these tasks, the Amazon FSx service account must have, at a minimum, the set of permissions that are described in Service account permissions delegated to it.

Although members of the **Domain Admins** group have sufficient privileges to perform these tasks, we strongly recommend that you use a separate service account to delegate the required privileges to Amazon FSx.

For more information about how to delegate privileges using either the **Delegate Control** or **Advanced Features** features in the **Active Directory User and Computers** MMC snap-in, see Delegating permissions to the Amazon FSx service account or group.

If you update your file system with a new service account, the new service account must have the required permissions and privileges to join your Active Directory and have **Full control** permissions for the existing computer objects associated with the file system. For more information, see Changing the Amazon FSx service account.

We recommend storing your Active Directory service account credentials in AWS Secrets Manager for enhanced security. This eliminates the need to store sensitive credentials in plaintext and aligns with security best practices. For more information, see Using a self-managed Microsoft Active Directory.

Delegating permissions to the Amazon FSx service account or group

The Amazon FSx service account or admin group must have the privileges necessary for it to join FSx for Windows File Server file systems to your self-managed Active Directory domain. To delegate these permissions, you can use either **Delegate Control** or **Advanced Features** in the Active Directory User and Computers MMC snap-in, as described in the following procedures.

To assign permissions using Delegate Control

To assign permissions to a service account or group using Delegate Control

1. Log in to your system 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 the name of your Amazon FSx service account or group, 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**.
12. Close the **Active Directory User and Computers** MMC snap-in.

To assign permissions using Advanced Features

1. Log in to your system as a domain administrator for your Active Directory domain.
2. Open the **Active Directory User and Computers** MMC snap-in.
3. Select **View** from the menu bar and ensure that **Advanced Features** is enabled (a check mark will appear next to it if the feature is enabled).
4. In the task pane, expand the domain node.
5. Locate and open (right-click) the context menu for the OU that you want to modify, and then choose **Properties**.
6. In the **OU Properties** pane, select the **Security** tab.
7. In the **Security** tab, select **Advanced**. Then select **Add**.
8. On the **Permission Entry** page, choose **Select a principal** and enter the name of your Amazon FSx service account or group. For **Applies to:**, choose **This Object and all Descendant Computer**. Ensure that the following are selected:

- **Modify permissions**
- **Create Computer Objects**
- **Delete Computer Objects**

9. Select **Apply**, and then select **OK**.
10. Close the **Active Directory User and Computers** MMC snap-in.

Validating your Active Directory configuration

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

To validate your Active Directory configuration

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

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

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

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

```
PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
```

5. Download the network validation tool using the following command.

```
PS C:\> Invoke-WebRequest "https://docs.aws.amazon.com/fsx/latest/WindowsGuide/
samples/AmazonFSxADValidation.zip" -OutFile "AmazonFSxADValidation.zip"
```

6. Expand the zip file by using the following command.

```
PS C:\> Expand-Archive -Path "AmazonFSxADValidation.zip"
```

7. Add the `AmazonFSxADValidation` module to the current session.

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

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

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

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

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

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

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

```
PS C:\>
$FSxADValidationArgs = @{
```

```
    # DNS root of ActiveDirectory domain
    DomainDNSRoot = 'DOMAINNAME.COM'

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

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

    Credential = $Credential
}
```

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

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

10. Run the validation tool by using this command.

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

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

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

Test computer object amznfsxtestd53f deleted!
...
SUCCESS - All tests passed! Please proceed to creating an Amazon FSx file system.
 For your convenience, SelfManagedActiveDirectoryConfiguration of result can be
 used directly in CreateFileSystemWindowsConfiguration for New-FSXFileSystem
PS C:\AmazonFSxADValidation> $Result.Failures.Count
0
PS C:\AmazonFSxADValidation> $Result.Warnings.Count
```

```
0
```

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

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

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



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

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



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

```
PS C:\AmazonFSxADValidation> $Result.Failures

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


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

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

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

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

- The fully qualified domain name (FQDN) of your on-premises Microsoft Active Directory 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 an Active Directory service account that Amazon FSx uses to join the file system to your domain. You can provide these as either:
  - **Option 1**: AWS Secrets Manager secret ARN - The secret containing the username and password for a service account on your Active Directory domain. For more information, see Storing Active Directory credentials using AWS Secrets Manager.
  - **Option 2**: Plaintext credentials

- **Service account username –** The user name of the service account in your existing Microsoft Active Directory. Don't include a domain prefix or suffix. For example, for `EXAMPLE\ADMIN`, use only `ADMIN`.
- **Service account password –** The password for the service account.

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. The domain group name you provide must be unique in your Active Directory.

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

> ⚠ **Important**
>
> Amazon FSx only registers DNS records for a file system if the Active Directory domain that you are joining it to is using Microsoft DNS as the default DNS. If you are using a third-party DNS, you will need to manually setup DNS entries for your Amazon FSx file systems after you create your file system. For more information on choosing the correct IP addresses to use for the file system, see Getting the correct file system IP addresses to use for manual DNS entries.

Before you begin

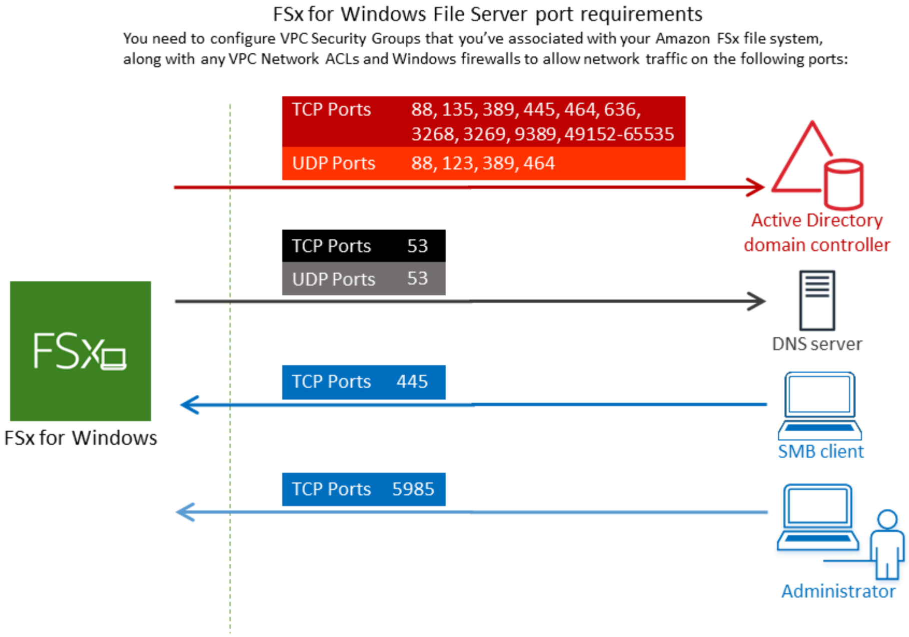
Make sure that you have completed the Prerequisites detailed in Using a self-managed Microsoft Active Directory.

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

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the dashboard, choose **Create file system** to start the file system creation wizard.

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

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


FSx for Windows File Server port requirements
You need to configure VPC Security Groups that you've associated with your Amazon FSx file system, along with any VPC Network ACLs and Windows firewalls to allow network traffic on the following ports:
TCP Ports 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535
UDP Ports 88, 123, 389, 464
Active Directory domain controller
TCP Ports 53
UDP Ports 53
DNS server
FSx for Windows
TCP Ports 445
SMB client
TCP Ports 5985
Administrator

The following table identifies the role of each port.

Protocol	Ports	Role
TCP/UDP	53	Doma Name Syste (DNS)
TCP/UDP	88	Kerbe authe ation
TCP/UDP	464	Chang Se t passw

Protocol	Ports	Role
TCP/UDP	389	Lightw ht Direct Access Protoc (LDAP
UDP	123	Netwo Time Protoc (NTP)
TCP	135	Distrib ed Comp Enviro nt / End Point Mapp (DCE / EPMA
TCP	445	Direct Servic SMB file sharin

Protocol	Ports	Role
TCP	636	Lightw ht Direct Access Protoc over TLS/ SSL (LDAP
TCP	3268	Micros Globa Catalo
TCP	3269	Micros Globa Catalo over SSL
TCP	5985	WinRM 2.0 (Micro t Windo Remo Manag t)

Protocol	Ports	Role
TCP	9389	Micros Active Direct DS Web Servic Power l
TCP	49152 - 65535	Epher ports for RPC

Important

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

Note

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

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

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

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

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

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

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

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

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

13. Enter at least one, and no more than two, values for **DNS Server IP Addresses** for the self-managed Microsoft Active Directory directory.

14. **Service account credentials** – Choose how to provide your service account credentials:

    - **Option 1**: AWS Secrets Manager secret ARN - The secret containing the username and password for a service account on your Active Directory domain. For more information, see Storing Active Directory credentials using AWS Secrets Manager.
    - **Option 2**: Plaintext credentials
      - **Service account username** – The user name of the service account in your existing Microsoft Active Directory. Don't include a domain prefix or suffix. For example, for `EXAMPLE\ADMIN`, use only `ADMIN`.
      - **Service account password** – The password for the service account.
      - **Confirm password** – The password for the service account.

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

15. For **Delegated file system administrators group**, specify the `Domain Admins` group or a custom delegated file system administrators group (if you've created one). The group you specify should have the delegated authority to perform administrative tasks on your file system. If you don't provide a value, Amazon FSx uses the Builtin `Domain Admins` group. Note that Amazon FSx does not support having a `Delegated file system administrators group` (either the `Domain Admins` group or a custom group you specify) that is located in the Builtin container.

⚠ **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 Active Directory domain. If the name of this Builtin group has been changed or if you're using a different group for domain administration, you must provide that name for the group here.

⚠ **Important**

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

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

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

```
aws fsx --region us-east-2 \
create-file-system \
--file-system-type WINDOWS \
--storage-capacity 300 \
--security-group-ids security-group-id \
--subnet-ids subnet-id\
--windows-configuration
 SelfManagedActiveDirectoryConfiguration='{DomainName="corp.example.com", \
OrganizationalUnitDistinguishedName="OU=FileSystems,DC=corp,DC=example,DC=com",FileSystemAdmini
 \
UserName="FSxService",Password="password", \
   DnsIps=["10.0.1.18"]}',ThroughputCapacity=8
```

⚠ **Important**

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

Getting the correct file system IP addresses to use for manual DNS entries

Amazon FSx only registers DNS records for a file system if you are using Microsoft DNS as the default DNS service. If you are using a third-party DNS, you will need to manually setup DNS entries for your Amazon FSx file systems. This section describes how to obtain the correct file system IP addresses to use if you have to manually add the file system to your DNS. Note that once a file system is created, its IP addresses don't change until the file system is deleted.

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

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

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

> **Note**
>
> If you need to setup DNS entries for your Windows Remote PowerShell Endpoint for Single-AZ 2 or Multi-AZ file systems, you should use the **Primary private IPv4 address** for the elastic network interface for your **Preferred subnet**. For more information, see Using the Amazon FSx CLI for PowerShell.

Updating a self-managed Active Directory configuration

To help ensure continuous, uninterrupted availability of your Amazon FSx file system, you must update the file system's Active Directory configuration when any of the following Active Directory properties change:

- The DNS server IP addresses
- The service account credentials of the self-managed Active Directory

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

If there's an issue with the updated self-managed Active Directory configuration, the file system state switches to **Misconfigured**. This state shows an error message and recommended corrective action beside the file system description in the console, API, and CLI. After taking the recommended corrective action, verify that your file system's state eventually changes to **Available**.

> **Important**
>
> If you update your file system with a new service account, ensure that the new service account has **Full control** permissions for the existing computer objects associated with the file system.

For information about troubleshooting possible issues related to self-managed Active Directory configurations, see File system is in a misconfigured state.

You can use the AWS Management Console, Amazon FSx API, or AWS CLI to update the service account credentials and the DNS server IP addresses of a file system's self-managed Active Directory configuration. You can track the progress of a self-managed Active Directory configuration update at any time using the AWS Management Console, CLI, and API. For more information, see Monitoring self-managed Active Directory updates.

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

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system for which you want to update self-managed Active Directory configuration.
3. In the **Network & security** tab, then choose **Update** for the **DNS server IP addresses**, or for the service account username, depending on which Active Directory properties you are updating.
4. Enter the new DNS server IP addresses, or the new service account credentials (username and password) or secret ARN in the dialog that appears. You can use AWS Secrets Manager to store your credentials. For more information, see Storing Active Directory credentials using AWS Secrets Manager.
5. Choose **Update** to initiate the Active Directory configuration update.

   You can monitor the update progress using the AWS Management Console or the AWS CLI.

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

- To update the self-managed Active Directory configuration of an FSx for Windows File Server file system, use the AWS CLI command update-file-system. Set the following parameters:
  - `--file-system-id` to the ID of the file system you are updating.
  - `UserName` the new username for the self-managed Active Directory service account.
  - `Password` the new password for the self-managed Active Directory service account.
  - `DomainJoinServiceAccountSecret` the AWS Secrets Manager secret containing the username and password for a service account on your Active Directory domain

> **Note**
>
> You can't provide both username/password and a domain join service account secret to connect to your Active Directory. Provide only one set of credentials.

- `DnsIps` the IP addresses for the self-managed Active Directory DNS servers.

```
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 \
  --windows-configuration
 'SelfManagedActiveDirectoryConfiguration={UserName=username,Password=password,\
     DnsIps=[192.0.2.0,192.0.2.24]}'
```

If the update action is successful, the service sends back an HTTP 200 response. The `AdminstrativeActions` object in the response describes the request and its status.

Changing the Amazon FSx service account

If you update your file system with a new service account, the new service account must have the required permissions and privileges to join your Active Directory and has **Full control** permissions for the existing computer objects associated with the file system. In addition, make sure that new service account is part of the trusted accounts with the enabled **Group Policy** setting **Domain controller: Allow computer account re-use during domain join**.

We strongly recommend using an Active Directory group to manage Active Directory permissions and configurations associated with service accounts.

When changing the service account for Amazon FSx, ensure that the service accounts have the following settings:

- The new service account (or the Active Directory group it is a member of) has **Full control** permissions for the existing computer objects associated with the file system.
- The new and previous service accounts (or the Active Directory group they are a member of) are part of the trusted accounts (or trusted Active Directory group) with the **Domain controller: Allow computer account re-use during domain join** Group Policy setting enabled on all domain controllers in the Active Directory.

If the service accounts do not meet these requirements, the following conditions could occur:

- For Single-AZ file systems, the file system could become **MISCONFIGURED_UNAVAILABLE**.
- For Multi-AZ file systems, the file system could become **MISCONFIGURED** and the RemotePowerShell endpoint name might change.

Configuring a domain controller's Group Policy

The following Microsoft recommended procedure describes how to use the domain controller Group Policy to configure the allow list policy.

To configure a domain controller's allow list policy

1. Install the September 12, 2023 or later Microsoft Windows updates on all member computers and domain controllers in your self-managed Microsoft Active Directory.
2. In a new or existing group policy that applies to all domain controllers in your self-managed Active Directory, configure the following settings.

   a. Navigate to **Computer Configuration>Policies>Windows Settings>Security Settings> Local Policies>Security Options**.

   b. Double-click **Domain controller: Allow computer account re-use during domain join**.

   c. Select **Define this policy setting and <Edit Security...>**.

   d. Use the object picker to add users or groups of trusted computer account creators and owners to the **Allow** permission. (As a best practice, we highly recommend that you use groups for permissions.) **Do not add the user account that performs the domain join.**

   > ⚠ **Warning**
   >
   > Limit membership to the policy to trusted users and service accounts. Do not add authenticated users, everyone or other large groups to this policy. Instead, add specific trusted users and service accounts to groups and add those groups to the policy.

3. Wait for the Group Policy refresh interval or run **gpupdate /force** on all domain controllers.
4. Verify that the HKLM\System\CCS\Control\SAM – “ComputerAccountReuseAllowList” registry key is populated with the desired SDDL. **Do not manually edit the registry**.

5. Attempt to join a computer that has the September 12, 2023, or later updates installed. Ensure that one of the accounts listed in the policy owns the computer account. Also ensure that its registry does not have the **NetJoinLegacyAccountReuse** key enabled (set to 1). If the domain join fails, check the **`c:\windows\debug\netsetup.log`**.

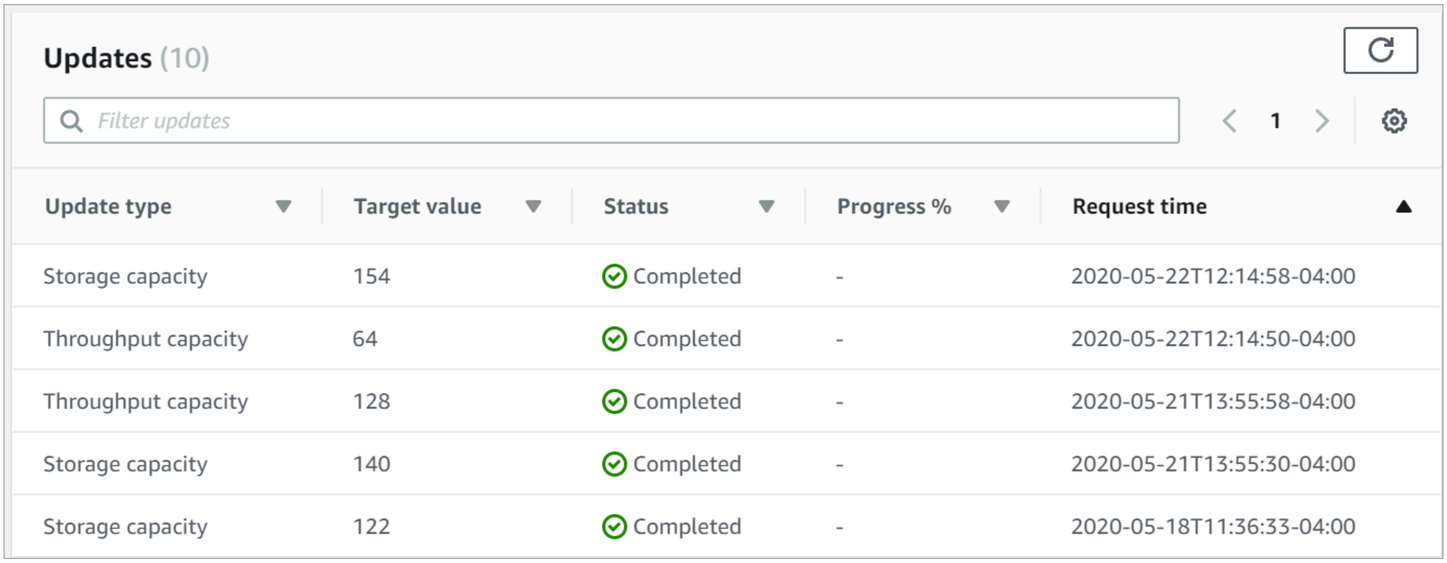
Monitoring self-managed Active Directory updates

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

When you update your file system's self-managed Active Directory configuration, the file system's state switches from **Available** to **Updating** while the update is applied. Once the update is complete, the state switches back to **Available**. An Active Directory configuration update can take up to several minutes to complete.

Monitoring updates in the console

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



Update type	Target value	Status	Progress %	Request time
Storage capacity	154	✓ Completed	-	2020-05-22T12:14:58-04:00
Throughput capacity	64	✓ Completed	-	2020-05-22T12:14:50-04:00
Throughput capacity	128	✓ Completed	-	2020-05-21T13:55:58-04:00
Storage capacity	140	✓ Completed	-	2020-05-21T13:55:30-04:00
Storage capacity	122	✓ Completed	-	2020-05-18T11:36:33-04:00

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

Update type

Supported types are as follows:

- DNS server IP address

- Service account credentials

Target value

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

Status

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

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

Progress %

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

Request time

The time that Amazon FSx received the update action request.

Monitoring updates using the AWS CLI and API

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

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

```
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 1000,
            "AdministrativeActions": [
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "RequestTime": 1581694766.757,
                    "Status": "PENDING",
                    "TargetFileSystemValues": {
                        "WindowsConfiguration": {
                            "SelfManagedActiveDirectoryConfiguration": {
                                "UserName": "serviceUser",
                            }
                        }
                    }
                },
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "RequestTime": 1619032957.759,
                    "Status": "FAILED",
                    "TargetFileSystemValues": {
                        "WindowsConfiguration": {
                            "SelfManagedActiveDirectoryConfiguration": {
                            "DnsIps": [
                                    "10.0.138.161"
                                ]
                            }
                        }
                    },
                    "FailureDetails": {
                        "Message": "Failure details message."
                    }
                }
            ],
     .
     .
     .
```

FSx for Windows File Server performance

FSx for Windows File Server offers file system configuration options to meet a variety of performance needs. Following is an overview of Amazon FSx file system performance, with a discussion of the available performance configuration options and useful performance tips.

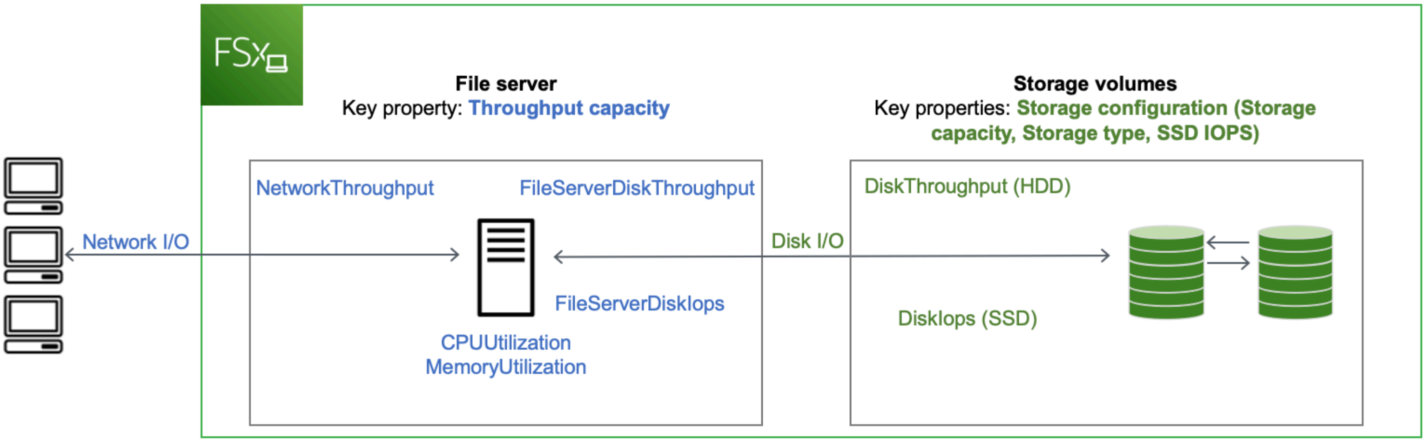
Topics

- File system performance
- Additional performance considerations
- Impact of throughput capacity on performance
- Choosing the right level of throughput capacity
- Impact of storage configuration on performance
- Example: storage capacity and throughput capacity
- Measuring performance using CloudWatch metrics
- Troubleshooting file system performance issues

File system performance

Each FSx for Windows File Server file system consists of a Windows file server that clients communicate with and a set of storage volumes, or disks, attached to the file server. Each file server employs a fast, in-memory cache to enhance performance for the most frequently accessed data.

The following diagram illustrates how data is accessed from an FSx for Windows File Server file system.



When a client accesses data that is stored in the in-memory cache, the data is served directly to the requesting client as *network I/O*. The file server doesn't need to read it from or write it into the disk. The performance of this data access is determined by the network I/O limits and the size of the in-memory cache.

When a client accesses data that is not in cache, the file server reads it from or writes it into the disk as *disk I/O*. The data is then served from the file server to the client as network I/O. The performance of this data access is determined by the network I/O limits as well as the disk I/O limits.

Network I/O performance and file server in-memory cache are determined by a file sytem's throughput capacity. Disk I/O performance is determined by a combination of throughput capacity and storage configuration. The maximum disk I/O performance, which consists of disk throughput and disk IOPS levels, that your file system can achieve is the lower of:

- The disk I/O performance level provided by your file server, based on the throughput capacity you select for your file system.
- The disk I/O performance level provided by your storage configuration (the storage capacity, storage type, and SSD IOPS level you select for your file system).

Additional performance considerations

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

Latency

FSx for Windows File Server file servers employ a fast, in-memory cache to achieve consistent sub-millisecond latencies for actively accessed data. For data that is not in the in-memory cache, that is, for file operations that need to be served by performing I/O on the underlying storage volumes, Amazon FSx provides sub-millisecond file operation latencies with solid state drive (SSD) storage, and single-digit millisecond latencies with hard disk drive (HDD) storage.

Throughput and IOPS

Amazon FSx file systems provide up to 2 GBps and 80,000 IOPS in all AWS Regions where Amazon FSx is available, and 12 GBps of throughput and 400,000 IOPS in US East (N. Virginia), US West (Oregon), US East (Ohio), Europe (Ireland), Asia Pacific (Tokyo), and Asia Pacific (Singapore). The specific amount of throughput and IOPS that your workload can drive on your file system depends on the throughput capacity, storage capacity and storage type 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 GBps 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. Although there's a theoretical limit to the number of SMB connections supported by Windows, this limit is in the millions, and practically you can have an unlimited number of SMB connections.

Burst performance

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 an 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 throughput capacity on performance

Throughput capacity determines file system performance in the following categories:

- Network I/O – The speed at which the file server can serve file data to clients accessing it.
- File server CPU and memory – Resources that are available for serving file data and performing background activities such as data deduplication and shadow copies.
- Disk I/O – The speed at which the file server can support I/O between the file server and the storage volumes.

The following tables provide details about the maximum levels of network I/O (throughput and IOPS) and disk I/O (throughput and IOPS) that you can drive with each provisioned throughput capacity configuration, and the amount of memory available for caching and supporting background activities such as data deduplication and shadow copies. While you can select levels of throughput capacity below 32 megabytes per second (MBps) when you use the Amazon FSx API or CLI, keep in mind that these levels are meant for test and development workloads, not for production workloads.

> **Note**
>
> Note that throughput capacity levels of 4,608 MBps and higher are supported only in the following regions: US East (N. Virginia), US West (Oregon), US East (Ohio), Europe (Ireland), Asia Pacific (Tokyo), and Asia Pacific (Singapore).

Network I/O and memory

FSx throughput capacity (MBps)	Network throughput (MBps)		Network IOPS	Memory (GB)
	Baseline	**Burst (for a few minutes a day)**		
32	32	600	Thousands	4
64	64	600	Tens of thousands	8

FSx throughput capacity (MBps)	Network throughput (MBps)		Network IOPS	Memory (GB)
128	150	1,250		8
256	300	1,250	Hundreds of thousands	16
512	600	1,250		32
1,024	1,500	–		72
2,048	3,125	–		144
4,608	9,375	–	Millions	192
6,144	12,500	–		256
9,216	18,750	–		384
12,288	21,250	–		512

Disk I/O

FSx throughput capacity (MBps)	Disk throughput (MBps)		Disk IOPS	
	Baseline	Burst (for 30 mins a day)	Baseline	Burst (for 30 mins a day)
32	32	260	2K	12K
64	64	350	4K	16K
128	128	600	6K	20K
256	256	600	10K	20K
512	512	–	20K	–
1,024	1,024	–	40K	–

FSx throughput capacity (MBps)	Disk throughput (MBps)		Disk IOPS	
2,048	2,048	–	80K	–
4,608	4,608	–	150K	–
6,144	6,144	–	200K	–
9,216	9,216[1]	–	300K[1]	–
12,288	12,288[1]	–	400K[1]	–

Note

[1] If you have a Multi-AZ file system with a throughput capacity of 9,216 or 12,288 MBps, performance will be limited to 9,000 MBps and 262,500 IOPS for write traffic only. Otherwise, for read traffic on all Multi-AZ file systems, read and write traffic on all Single-AZ file systems, and all other throughput capacity levels, your file system will support the performance limits shown in the table.

Choosing the right level of throughput capacity

When you create a file system using the Amazon Web Services Management Console, Amazon FSx automatically picks the recommended throughput capacity level for your file system based on the amount of storage capacity you configure. While the recommended throughput capacity should be sufficient for most workloads, you have the option to override the recommendation and configure a specific amount of throughput capacity to meet your workload's needs. For example, if your workload requires driving 1 GBps of traffic to your file system, you should select a throughput capacity of at least 1,024 MBps. The following table provides the minimum recommended throughput capacity level for a file system based on the amount of provisioned storage capacity.

SSD storage capacity (GiB)	HDD storage capacity (GiB)	Minimum recommended throughput capacity (MBps)
Up to 640	Up to 3,200	32

SSD storage capacity (GiB)	HDD storage capacity (GiB)	Minimum recommended throughput capacity (MBps)
641—1,280	3201—6,400	64
1281—2,560	6,401—12,800	128
2,561—5,120	12,801—25,600	256
5,121—10,240	25,601—51,200	512
10,241—20,480	>51,200	1,024
>20,480	NA	2,048

You should also consider the features you're planning to enable on your file system in deciding the level of throughput to configure. For example, enabling Shadow Copies may require you to increase your throughput capacity to a level up to three times your expected workload to ensure the file server can maintain the shadow copies with the available I/O performance capacity. If you are enabling Data Deduplication, you should determine the amount of memory associated with your file system's throughput capacity and ensure this amount of memory is sufficient for the size of your data.

You can adjust the amount of throughput capacity up or down at any time after you create it. For more information, see Managing throughput capacity.

You can monitor your workload's utilization of file server performance resources and get recommendations on which throughput capacity to select by viewing the **Monitoring & performance > Performance** tab of your Amazon FSx console. We recommend testing in a pre-production environment to ensure the configuration you've selected meets your workload's performance requirements. For Multi-AZ file systems, we also recommend testing the impact of the failover process that occurs during file system maintenance, throughput capacity changes, and unplanned service disruption on your workload, as well as ensuring that you have provisioned sufficient throughput capacity to prevent performance impact during these events. For more information, see Accessing file system metrics.

Impact of storage configuration on performance

Your file system's storage capacity, storage type, and SSD IOPS level all impact the disk I/O performance of your file system. You can configure these resources to deliver the desired performance levels for your workload.

You can increase storage capacity and scale SSD IOPS at any time. For more information, see Managing storage capacity and Managing SSD IOPS. You can also upgrade your file system from HDD storage type to SSD storage type. For more information, see Managing your file system's storage type.

Your file system provides the following default levels of disk throughput and IOPS:

Storage type	Disk throughput (MBps per TiB of storage)	Disk IOPS (per TiB of storage)
SSD	750	3,000[1]
HDD	12 baseline; 80 burst (up to a max. of 1 GBps per file system)	12 baseline; 80 burst

Note

[1] For file systems with SSD storage type, you can provision additional IOPS, up to a maximum ratio of 500 IOPS per GiB of storage and 400,000 IOPS per file system.

HDD burst performance

For HDD storage volumes, Amazon FSx uses a burst bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

The available throughput of an HDD storage volume is expressed by the following formula:

```
(Volume size) × (Credit accumulation rate per TiB) = Throughput
```

For a 1-TiB HDD volume, burst throughput is limited to 80 MiBps, the bucket fills with credits at 12 MiBps, and it can hold up to 1 TiB-worth of credits.

HDD storage volumes can experience significant performance variations depending on the workload. Sudden spikes in IOPS or throughput can lead to disk performance degradation. The `DiskThroughputBalance` metric provides information about the burst credit balance for both disk throughput and disk IOPS utilization. For example, if your workload exceeds the baseline HDD IOPS limits (12 IOPS per TiB of storage), the Disk IOPS utilization (HDD) will be above 100% and result in depleting the burst credit balance, which you can see in the `DiskThroughputBalance` metric. In order for your workload to continue driving high levels of I/O, you may need to do one of the following:

- Reduce the I/O demands for your workload so that the burst credit balance is replenished.
- Increase the file system's storage capacity to provide higher baseline level of disk IOPS.
- Upgrade the file system to use SSD storage, which provides a higher baseline level of disk IOPS to better match your workload’s requirements.

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 the file system's throughput capacity
    - the disk throughput levels of 24 MBps baseline (12 MBps per TB * 2 TiB) and 160 MBps burst (80 MBps per TiB * 2 TiB) supported by the storage volumes, based on storage type and 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 Monitoring with Amazon CloudWatch.

Troubleshooting file system performance issues

The performance of your FSx for Windows File Server file system depends on several factors, including the traffic that you drive to your file system, how you provision your file system, and the resources being consumed by features that are enabled, such as Data Deduplication or Shadow Copies. For information about understanding your file system’s performance, see FSx for Windows File Server performance.

Topics

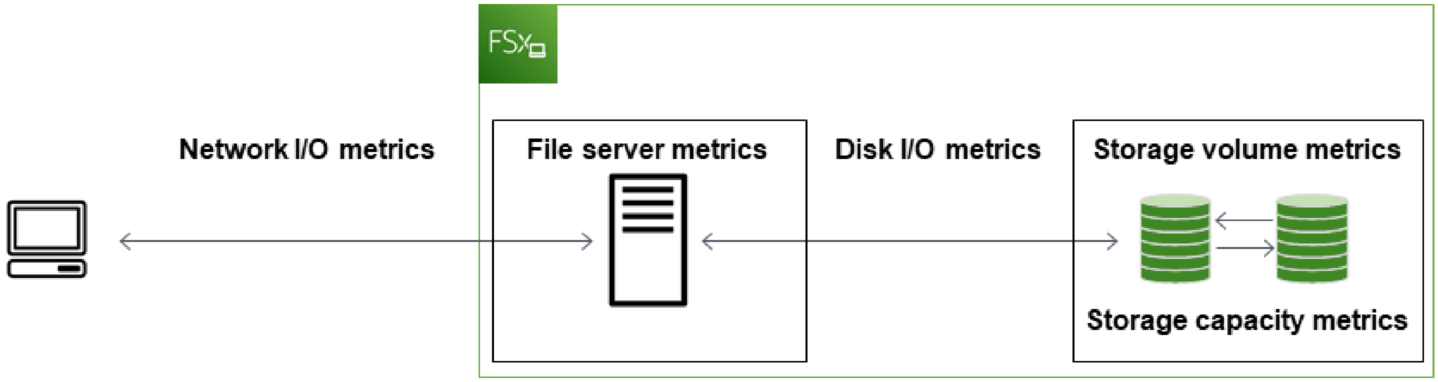
- How do I determine the throughput and IOPS limits for my file system?
- What is the difference between network I/O and disk I/O? Why is my network I/O different from my disk I/O?
- Why is my CPU or memory usage high, even when my network I/O is low?
- What is bursting? How much bursting is my file system using? What happens when burst credits run out?
- I see a warning on the Monitoring & performance page – do I need to change my file system’s configuration?
- My metrics were temporarily missing, should I be concerned?

How do I determine the throughput and IOPS limits for my file system?

To view a file system's throughput and IOPS limits, refer to the table showing performance levels based on the amount of provision throughput capacity.

What is the difference between network I/O and disk I/O? Why is my network I/O different from my disk I/O?

Amazon FSx file systems include one or more file servers that serve data over the network to the clients accessing the file system. This is the network I/O. The file server has a fast, in-memory cache to enhance performance for the most frequently accessed data. The file servers also drives traffic to the storage volumes that host your file system data. This is the disk I/O. The following diagram illustrates network and disk I/O for an Amazon FSx file system.



For more information, see Monitoring with Amazon CloudWatch.

Why is my CPU or memory usage high, even when my network I/O is low?

The file server CPU and memory usage depends not only on the network traffic you drive, but also the features you have enabled on your file system. How you configure and schedule these features can impact CPU and memory utilization.

Data Deduplication jobs in progress can consume memory. You can modify the configuration of deduplication jobs to reduce memory requirements. For example, you can constrain the optimization to run on specific file types or folders, or set a minimum file size and age for optimization. We also recommend configuring deduplication jobs to run during idle periods when there is minimal load on your file system. For more information, see Reducing storage costs with Data Deduplication.

If you have access-based enumeration enabled, you might see high CPU utilization when your end-users view or list file shares, or during the Optimization phase of a storage scaling job. For more information, see Enable access-based enumeration on a namespace in the *Microsoft Storage Documentation*.

What is bursting? How much bursting is my file system using? What happens when burst credits run out?

File-based workloads are typically spiky, characterized by short, intense periods of high I/O with idle time between bursts. To support these types of workloads, in addition to the baseline speeds that a file system can sustain, 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 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 to burst above the baseline limits (up to the burst limits) when required. For more information about the burst limits and duration for your file system, see FSx for Windows File Server performance.

I see a warning on the Monitoring & performance page – do I need to change my file system's configuration?

The **Monitoring & performance** page includes warnings that indicate when recent workload demands have approached or exceeded resource limits determined by how you've configured your file system. This doesn't necessarily mean you need to change your configuration, though your file system might be under-provisioned for your workload if you don't take the recommended action.

If the workload that caused the warning was atypical and you do not expect it to continue, it may be safe to take no action and closely monitor your utilization going forward. However, if the workload that caused the warning is typical and you expect it to continue, or even intensify, we advise following the recommended action to increase file server performance (by increasing throughput capacity) or increase storage volume performance (by increasing storage capacity, or by switching from HDD to SSD storage).

Note

Certain file system events can consume disk I/O performance resources and potentially trigger performance warnings. For example:

- The optimization phase of storage capacity scaling can generate increased disk throughput, as described in Storage capacity increases and file system performance
- For Multi-AZ file systems, events such as throughput capacity scaling, hardware replacement, or Availability Zone disruption result in automatic failover and failback

events. Any data changes that occur during this time need to be synchronized between the primary and secondary file servers, and Windows Server runs a data synchronization job that can consume disk I/O resources. For more information, see Managing throughput capacity.

My metrics were temporarily missing, should I be concerned?

Single-AZ file systems will experience unavailability during file system maintenance, infrastructure component replacement, and when an Availability Zone is unavailable. During these times, metrics will not be available.

In a Multi-AZ deployment, Amazon FSx automatically provisions and maintains a standby file server in a different Availability Zone. If there is file system maintenance or an unplanned service disruption, Amazon FSx automatically fails over to the secondary file server, allowing you to continue accessing your data without manual intervention. During the brief period in which your file system is failing over and failing back, metrics may be temporarily unavailable.

Administering FSx for Windows file systems

Amazon FSx provides a wide range of administrative capabilities that help you easily manage and grow your Amazon FSx for Windows File Server file systems to meet changing workload and user requirements, and your organizations regulatory and compliance needs. The following is a list of some of the file system configurations that you can manage using the AWS Management Console, AWS CLI and API, the Amazon FSx CLI for remote management on PowerShell, and native Microsoft Windows Server graphical interfaces.

- Storage capacity
- Storage type
- SSD IOPS
- Throughput capacity
- DNS aliases
- Data deduplication
- Shadow copies
- Storage quotas
- File access auditing
- File shares

The following sections provide information about the file system administrative features and setting that are available to you. We've included guidance to help you determine which options are best for your situation, and best practices where applicable.

Topics

- Amazon FSx file system status
- Using the Amazon FSx CLI for PowerShell
- Starting an Amazon FSx remote PowerShell session
- One-time file system setup tasks using the Amazon FSx CLI for remote management on PowerShell
- Troubleshooting access to the Amazon FSx CLI on PowerShell
- File system maintenance window

- Changing the weekly maintenance window
- Managing DNS aliases
- User sessions and open files
- File Server Resource Manager on FSx for Windows File Server
- Managing storage on FSx for Windows File Server
- Using DFS Namespaces
- Managing throughput capacity
- Managing network type
- Tagging your Amazon FSx resources
- Update a file system using the AWS CLI

Amazon FSx file system status

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

File system status	Description
AVAILABLE	The file system is in a healthy state, and is reachable and available for use.
CREATING	Amazon FSx is creating a new file system.
DELETING	Amazon FSx is deleting an existing file system.
UPDATING	The file system is undergoing a customer-initiated update.
MISCONFIGURED	The file system is in an impaired state due to a change in your Active Directory environment. Your file system is either currently unavailable or at risk of losing availability, and backups may not succeed. For information on restoring availability, see File system is in a misconfigured state.

File system status	Description
MISCONFIGURED_UNAVAILABLE	The file system is currently unavailable due to a change in your Active Directory environme nt. For information on restoring availability, see File system is in a misconfigured state.
FAILED	• When creating a new file system, Amazon FSx was unable to create the new file system. • The file system is unavailable. • The file system has failed and Amazon FSx can't recover it. • Amazon FSx is unable to create backups.

Using the Amazon FSx CLI for PowerShell

This chapter describes how to access the Amazon FSx CLI for remote management on PowerShell to perform file system administrative tasks for FSx for Windows file systems. You can also use the Microsoft Windows–native graphical user interface (GUI) to perform some administrative tasks.

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

- Be able to connect to a Windows compute instance that has network connectivity with your FSx for Windows File Server file system.
- Be logged into the Windows compute instance as a member of the file system administrators group. If you are using AWS Managed Microsoft AD, that is the *AWS Delegated FSx Administrators* group. If you are using a self-managed Microsoft Active Directory, that is the *Domain Admins* group or the custom group that you specified for administration when you created your file system. For more information, see Best practices when using a self-managed Active Directory.
- Your file system's VPC security group inbound rules allow traffic on port 5985.

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

- User credentials are authenticated using Kerberos authentication.
- Management session communications between the connected client and file system are encrypted using Kerberos.

You have two options to run remote management CLI commands on your Amazon FSx file system:

- You can establish a long-running Remote PowerShell session and run the commands inside the session.
- You can use the `Invoke-Command` to run a single command or a single block of commands without establishing a long-running Remote PowerShell session.

If you want to set and pass variables as parameters to the remote management command, you will need to use `Invoke-Command`.

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

You need to use the file system's *Windows Remote PowerShell Endpoint* to access the Remote PowerShell. The remote administration endpoint has the format of `amznfsxctlyaa1k.`*`ActiveDirectory-DNS-name`*, for example, `amznfsxctlyaa1k.corp.example.com`. You can find the endpoint name by using the AWS Management Console in the **File system details** page on the **Network & security** tab. Use the AWS CLI `describe-file-systems` command to view the `RemoteAdministrationEndpoint` property returned in the response.

You can use the `Get-Command` cmdlet to retrieve information about the cmdlets, functions, and aliases available in PowerShell. For more information, see the Microsoft Get-Command documentation.

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

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName
 amznfsxctlyaa1k.corp.example.com -ConfigurationName FSxRemoteAdmin -scriptblock { fsx-
command}
```

For instructions on how to start a long-lived Remote PowerShell session on your FSx for Windows File Server files system, see Starting an Amazon FSx remote PowerShell session

Starting an Amazon FSx remote PowerShell session

This topic provides instructions for starting a long-lived remote PowerShell session on your FSx for Windows File Server file server.

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 delegated FSx Administrators Group that you chose when you created the file system.
2. Open a Windows PowerShell window on the compute instance.
3. In the PowerShell, enter the following command to open a long-lived remote session on your Amazon FSx file system. Replace *`Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer. Use `FsxRemoteAdmin` as the session configuration name.

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

   If your instance is not part of the Amazon FSx Active Directory domain, you are prompted to enter user credentials in a pop-up. Enter the credentials of the user that is a member of the FSx Administrators Group. If your instance is joined to the domain, you will not be asked for credentials.

   > ⚠ **Important**
   >
   > The Windows Remote PowerShell endpoint might change if you are using self-managed Active Directory configuration and change the service account without

proper Active Directory Group Policy settings. For more information, see Changing the Amazon FSx service account for more details.

One-time file system setup tasks using the Amazon FSx CLI for remote management on PowerShell

Use the following Amazon FSx CLI for Remote Management on PowerShell commands to quickly implement the file system administration tasks following our best practices.

Managing storage consumption

Use the following commands to manage your file system storage consumption.

- To turn on data deduplication with the default schedule, run the following command.

  ```
  Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
   FSxRemoteAdmin -ScriptBlock { Enable-FsxDedup }
  ```

  Optionally, use the following command to get data deduplication operating on your files soon after a file is created, without requiring any minimum file age.

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

  For more information, see Reducing storage costs with Data Deduplication.

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

  ```
  $QuotaLimit = Quota limit in bytes
  $QuotaWarningLimit = Quota warning threshold in bytes
  Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
   FSxRemoteAdmin -ScriptBlock { Enable-FSxUserQuotas -Track -DefaultLimit
   $Using:QuotaLimit -DefaultWarningLimit $Using:QuotaWarningLimit }
  ```

  For more information, see Managing storage quotas.

Turning on shadow copies to enable end-users to recover files and folders to previous versions

Turn on shadow copies with the default schedule (weekdays 7 AM and 12 noon), as follows.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxShadowStorage -Default }

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxShadowCopySchedule -Default -Confirm:$False}
```

For more information, see Configuring shadow copies to use the default storage and schedule.

Enforcing encryption in transit

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

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Set-FsxSmbServerConfiguration -EncryptData $True -
RejectUnencryptedAccess $True -Confirm:$False}
```

You can close all open sessions and force clients currently connected to reconnect using encryption.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock { Close-FSxSmbSession -Confirm:$False}
```

For more information, see Managing encryption in transit and User sessions and open files.

Troubleshooting access to the Amazon FSx CLI on PowerShell

There are a number of potential causes for being unable to connect to your file system using Remote PowerShell, each with their own resolution, as follows.

To first ensure that you can connect successfully to the Windows Remote PowerShell Endpoint, you can also run a basic connectivity test. For example, you can run the `test-netconnection endpoint -port 5985` command.

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

The file system's security group must have an inbound rule that allows traffic on port 5985 in order to establish a Remote PowerShell session. For more information, see Amazon VPC Security Groups.

You have an external trust configured between the AWS managed Microsoft Active Directory and your on-premises Active Directory

In order to use the Amazon FSx Remote PowerShell with Kerberos authentication, you need to configure a local group policy on the client for forest search order. For more information, see the Microsoft documentation Configure Kerberos Forest Search Order (KFSO).

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

You need to add the following `-SessionOption` to your command: `-SessionOption (New-PSSessionOption -uiCulture "en-US")`

Following are two examples using `-SessionOption` when initiating a remote PowerShell session on your file system.

```
PS C:\Users\delegateadmin> Invoke-Command -ComputerName Windows Remote PowerShell
 Endpoint -ConfigurationName FSxRemoteAdmin -scriptblock {fsx-command} -SessionOption
 (New-PSSessionOption -uiCulture "en-US")
```

```
PS C:\Users\delegateadmin> Enter-Pssession -ComputerName Windows Remote PowerShell
 Endpoint -ConfigurationName FsxRemoteAdmin -SessionOption (New-PSSessionOption -
uiCulture "en-US")
```

File system maintenance window

Amazon FSx for Windows File Server performs routine software patching for the Microsoft Windows Server software that it manages. The maintenance window specifies the day of the week and the time of day when this maintenance process begins. You can specify the start period of the maintenance window during file system creation. If you do not specify one, a 30-minute default maintenance start window is assigned. The duration of the maintenance window depends on multiple factors, including the scope of the maintenance, and the process of synchronizing any file read and write activity that occurs during maintenance between the primary and secondary servers for Multi-AZ file systems. For more information, see Failing over process.

FSx for Windows File Server lets you adjust the start time of your maintenance window to accommodate your workload and operational requirements. You can move the start time of your maintenance window as frequently as required, provided that a maintenance window start time is scheduled at least once every 14 days. If a patch is released and you haven't scheduled a maintenance window within 14 days, FSx for Windows File Server proceeds with maintenance on the file system to ensure its security and reliability. For more information about how to adjust the start time of your file system's maintenance window, see Changing the weekly maintenance window.

While patching is in progress, expect your Single-AZ file systems to be unavailable, typically for less than 20 minutes. Multi-AZ file systems remain available and automatically fail over and fail back between the preferred and the standby file servers. For more information, see Failing over process. Because patching for Multi-AZ file systems involves failing over and failing back between the file servers, any file read and write activity occuring during this time must be synchronized between the preferred and the standby file servers. To reduce patching time, we recommend scheduling your maintenance window during idle periods when there's minimal load on your file system.

> **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.

Changing the weekly maintenance window

FSx for Windows File Server lets you adjust when your file system's maintenance window starts to accommodate your workload and operational requirements. You can use the AWS Management Console, AWS CLI, and Amazon FSx API to change when the weekly maintenance window starts, described in the following procedure.

To change the start time of the weekly maintenance window (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose **File systems** in the left hand navigation column.

3. Choose the file system that you want to change the weekly maintenance window for. The file system details page displays.
4. Choose **Administration** to display the file system administration **Settings** panel.
5. Choose **Update** to display the **Change maintenance window** window.
6. Enter the new day and time that you want the weekly maintenance window to start.
7. Choose **Save** to save your changes. The new maintenance start time is displayed in the **Administration Settings** panel.

To change the start time of the weekly maintenance window using the update-file-system CLI command, see Update a file system using the AWS CLI.

Managing DNS aliases

In addition to the default Domain Name System (DNS) name that Amazon FSx provides, you can also associate DNS aliases of your choosing with your file systems. With DNS aliases, you can continue using existing DNS names to access data stored on Amazon FSx when migrating file system storage from on-premises to Amazon FSx, without needing to update any tools or applications.

You can associate DNS aliases with new and existing FSx for Windows File Server file systems, and when you restore a backup to a new file system, using the AWS Management Console and AWS CLI. You can associate up to 50 DNS aliases with a file system at any one time.

Note

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

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

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

> **Note**
>
> The information presented here assumes that you're working entirely within Active Directory and that you're not using external DNS providers. Third-party DNS providers may result in unexpected behavior.
> Amazon FSx only registers DNS records for a file system if the Active Directory domain that you are joining it to is using Microsoft DNS as the default DNS. If you are using a third-party DNS, you will need to manually set up DNS entries for your Amazon FSx file systems after you create your file system. For more information on choosing the correct IP addresses to use for the file system, see Getting the correct file system IP addresses to use for manual DNS entries.

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

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

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

For more information, see Accessing data using DNS aliases.

A DNS alias name for your FSx for Windows File Server file system needs to meet the following requirements:

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

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

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

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

Topics

- DNS alias status
- Using DNS aliases with Kerberos authentication
- Viewing DNS aliases for file systems and backups
- Associating DNS aliases with file systems
- Managing DNS aliases on existing file systems

DNS alias status

DNS aliases can have one of the following status values:

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

Using DNS aliases with Kerberos authentication

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

If you have SPNs configured for the DNS alias that you've assigned to another file system on a computer object in your Active Directory, you must first remove those SPNs before adding SPNs to your file system's computer object. For more information, see Configure service principal names (SPNs) for Kerberos.

Viewing DNS aliases for file systems and backups

You can view the DNS aliases currently associated with your FSx for Windows File Server file systems and backups using the AWS Management Console, the AWS CLI, and API, as described in the following procedures.

To view DNS aliases associated with file systems

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

To view DNS aliases associated with backups

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

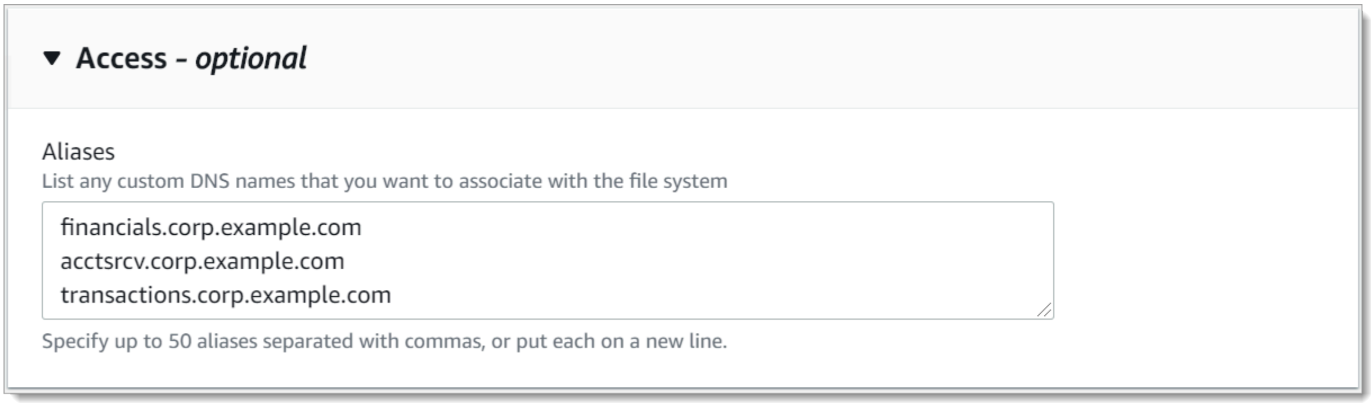
Associating DNS aliases with file systems

You can associate DNS aliases when creating a new FSx for Windows File Server file system from scratch, or when restoring a backup to a new file system, using the AWS Management Console, AWS CLI, and API, described the following procedures.

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

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

2. Follow the procedure for creating a new file system described in Step 5. Create your file system in the Getting Started section.
3. In the **Access - optional** section of the **Create file system** wizard, enter the DNS aliases that you want to associate with your file system.



▼ Access - *optional*

Aliases

List any custom DNS names that you want to associate with the file system

financials.corp.example.com
acctsrcv.corp.example.com
transactions.corp.example.com

Specify up to 50 aliases separated with commas, or put each on a new line.

4. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Accessing data using DNS aliases.

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

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

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

2. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Accessing data using DNS aliases.

To add or remove DNS aliases when restoring a backup (CLI)

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

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

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

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

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

2. When the file system is **Available**, you can access it using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Accessing data using DNS aliases.

Managing DNS aliases on existing file systems

You can add and remove aliases on existing FSx for Windows File Server file systems using the AWS Management Console and AWS CLI, as described in the following procedures.

To manage file system DNS aliases (console)

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

   - To associate DNS aliases – In the **Associate new aliases** box, enter the DNS aliases that you want to associate. Choose **Associate**.

- To disassociate DNS aliases – In the **Current aliases** list, choose the aliases to disassociate from. Choose **Disassociate**.

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

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

To associate DNS aliases with existing file systems (CLI)

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

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

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

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

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

2. Use the `describe-file-system-aliases` CLI command (DescribeFileSystemAliases is the equivalent API operation) to monitor the status of the aliases that you are associating.

3. When the `Lifecycle` has a value of AVAILABLE (a process that can take up to 2.5 minutes), you can access your file system using the DNS alias by configuring service principal names (SPNs) and updating or creating a DNS CNAME record for the alias. For more information, see Accessing data using DNS aliases.

To disassociate DNS aliases from file systems (CLI)

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

  The following command disassociates one alias from a file system.

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

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

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

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

User sessions and open files

You can monitor connected user sessions and open files on your FSx for Windows File Server file system using the Shared Folders tool. The Shared Folders tool provides a central location to monitor who is connected to the file system, along with what files are open and by whom. You can use this tool to do the following:

- Restore access to locked files.
- Disconnect a user session, which closes all files opened by that user.

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

Using the GUI to manage users and sessions

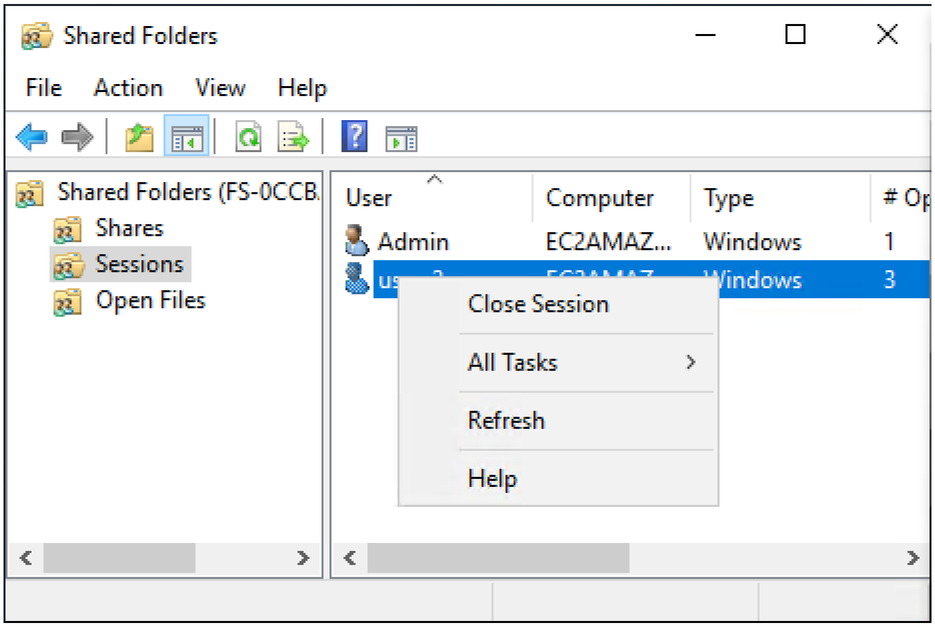
The following procedures detail how you can manage user sessions and open files on your Amazon FSx file system using the Microsoft Windows shared folders tool.

To launch the shared folders tool

1. Launch your Amazon EC2 instance and connect it to the Microsoft Active Directory that your Amazon FSx file system is joined to. To do this, choose one of the following procedures from the *AWS Directory Service Administration Guide*:
   - Seamlessly join a Windows EC2 instance
   - Manually join a Windows instance
2. Connect to your instance as a user that is a member of the file system administrators group. In AWS Managed Microsoft Active Directory, this group is called AWS Delegated FSx Administrators. In your self-managed Microsoft Active Directory, this group is called Domain Admins or the custom name for the administrators group that you provided during creation. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide*.
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.

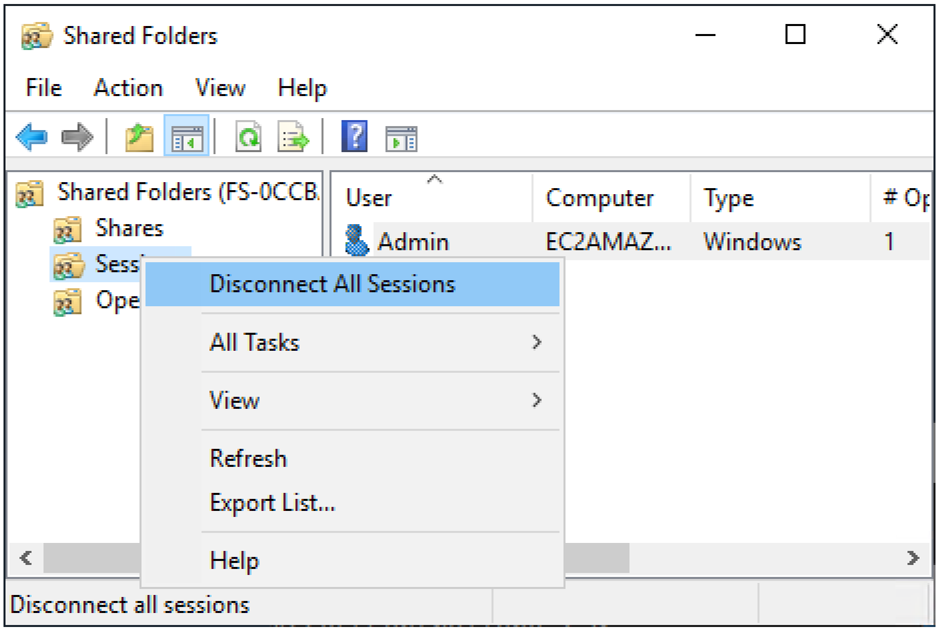
To manage user sessions (GUI)

In the Shared Folders tool, choose **Sessions** to view all the user sessions that are connected to your FSx for Windows File Server file system. If a user or application is accessing a file share on your Amazon FSx file system, this snap-in shows you their session. You can disconnect sessions by opening the context (right-click) menu for a session and choosing **Close Session**.



User	Computer	Type	# Op
Admin	EC2AMAZ...	Windows	1
us		Windows	3

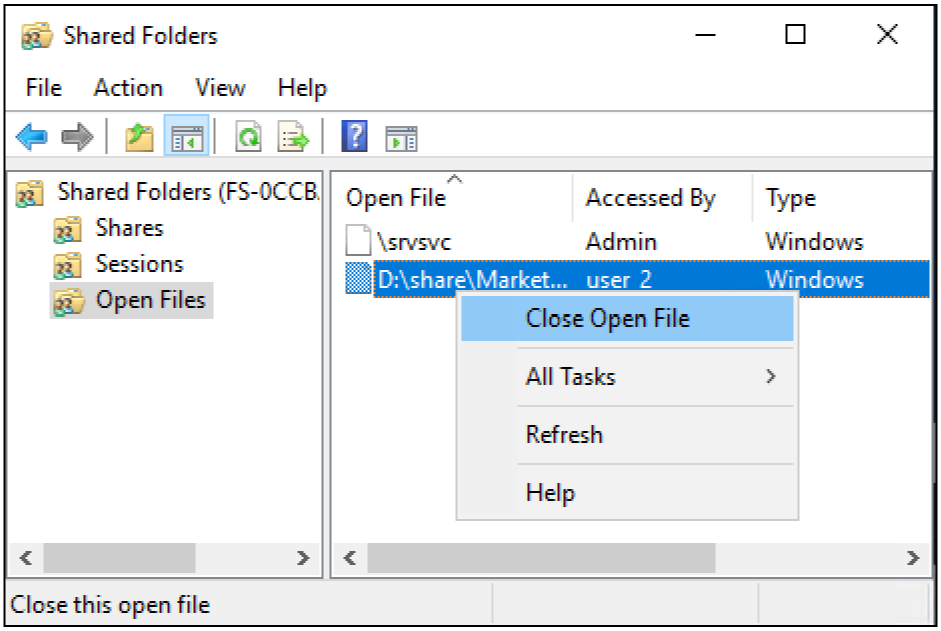
To disconnect all open sessions, open the context (right-click) menu for **Sessions**, choose **Disconnect All Sessions**, and confirm your action.



User	Computer	Type	# Op
Admin	EC2AMAZ...	Windows	1

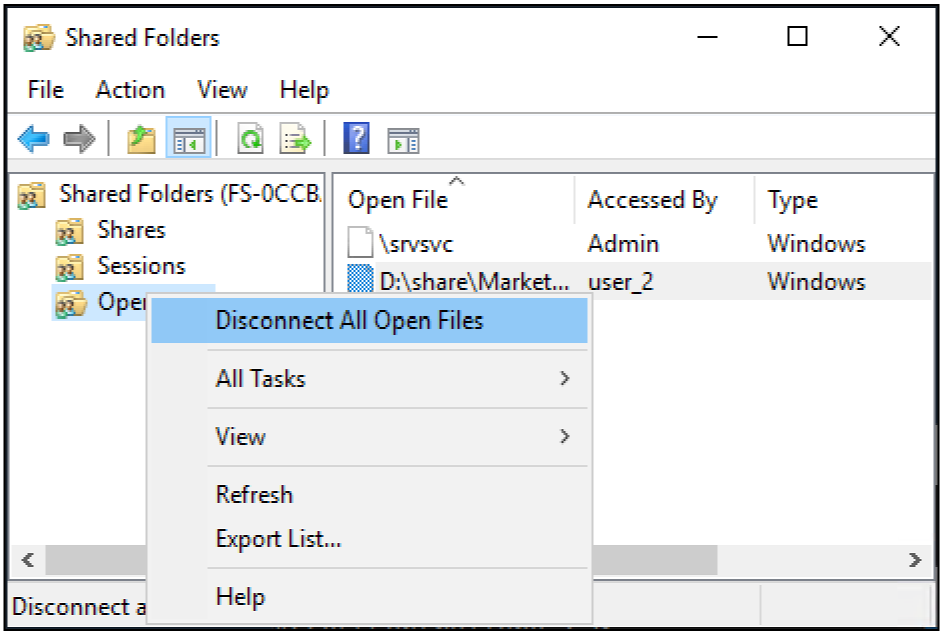
To manage open files (GUI)

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**.



Open File	Accessed By	Type
\srvsvc	Admin	Windows
D:\share\Market...	user 2	Windows

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.



Open File	Accessed By	Type
\srvsvc	Admin	Windows
D:\share\Market...	user_2	Windows

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 Using the Amazon FSx CLI for PowerShell.

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

Command	Description
Get-FSxSmbSession	Retrieves information about the Server Message Block (SMB) sessions that are currently established between the file system and the associated clients.
Close-FSxSmbSession	Ends an SMB session.
Get-FSxSmbOpenFile	Retrieves information about files that are open for the clients connected to the file system.
Close-FSxSmbOpenFile	Closes a file that is open for one of the clients of the SMB server.

The online help for each command provides a reference of all command options. To access this help, run the command with a **-?**, for example **Get-FSxSmbSession -?**.

File Server Resource Manager on FSx for Windows File Server

File Server Resource Manager (FSRM) is a Windows Server feature that helps you manage and classify data stored on your Amazon FSx for Windows File Server file system. FSRM provides automated policy enforcement and reporting capabilities that help you control storage costs, maintain compliance with data management policies, and organize files based on business rules.

With FSRM, you can set storage limits to prevent users from consuming excessive storage, automatically identify and classify sensitive data, block unauthorized file types from being saved to business folders, and generate detailed reports about storage usage patterns. These capabilities help you maintain an organized, efficient, and compliant file system without requiring manual intervention for every file or folder.

FSRM is particularly valuable for organizations that need to:

- Control storage costs by limiting how much disk space users and departments can store
- Identify sensitive data such as personally identifiable information or financial records
- Enforce policies about which file types are allowed in specific folders
- Generate compliance reports about data retention, file ownership, or storage usage
- Maintain visibility into how storage is being used across the organization

Key capabilities

- **Quota Management** - Set storage limits on folders to control how much space users and applications can consume. You can configure hard quotas that prevent users from exceeding limits or soft quotas that allow overages while sending notifications. Quotas help you manage storage costs and prevent users or departments from consuming disproportionate amounts of storage.
- **File Screening** - Control which types of files users can save to specific folders. You can block unauthorized file types such as executable files, media files, or personal documents in business folders. File screening helps you enforce data management policies, reduce security risks, and prevent storage waste from non-business files.
- **File Classification** - Automatically assign metadata properties to files based on their content or location. Classification helps you organize files, identify sensitive data, apply retention policies, and generate reports based on file characteristics. You can classify files by data sensitivity, department, retention period, or any other custom properties you define.
- **Storage Reports** - Generate detailed reports about file system usage, including large files, duplicate files, files by owner, files by type, and quota usage. Storage reports help you understand how storage is being consumed, identify files that can be archived or deleted, and make informed decisions about storage management.

Topics

- Getting Started with File Server Resource Manager
- Quota Management
- File Groups
- File Screening
- File Classification
- Storage Reports

- File Management Tasks
- FSRM Settings
- Event Logs
- Common Use Cases

Getting Started with File Server Resource Manager

You can enable File Server Resource Manager (FSRM) when creating a new Amazon FSx for Windows File Server file system, or you can update your existing file system to enable FSRM.

FSRM is supported only on Amazon FSx for Windows File Server file systems with SSD storage and a throughput capacity of 128 MB/s or greater. You can update the storage type to SSD and modify the throughput capacity at any time after you create the file system. For more information, see Updating the storage type of a FSx for Windows file system and Managing throughput capacity.

To enable FSRM when creating a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/
2. On the dashboard, choose **Create file system** to start the file system creation wizard.
3. Choose **Amazon FSx for Windows File Server** and then choose **Next**.
4. Select the **Standard Create** option
5. Provide the required information
6. Open the **File Server Resource Manager** block, choose **Enabled**.
7. For **Event log destination**, choose one of the following options:
   - **CloudWatch Logs** - Select a CloudWatch Logs log group to receive FSRM event logs. The name of the CloudWatch Logs log group must begin with the `'/aws/fsx/'` prefix.
   - **Kinesis Data Firehose** - Select a Kinesis Data Firehose delivery stream to receive FSRM event logs
8. Complete the remaining sections and choose **Create file system**.

To enable FSRM when creating a file system (CLI)

To enable FSRM when creating an FSx for Windows File Server file system, use the AWS CLI command `create-file-system`. Include the following FSRM configuration in the `--windows-configuration` parameter:

- `FsrmServiceEnabled` - Set to `true`
- `EventLogDestination` - The Amazon Resource Name (ARN) that specifies the destination of the FSRM event logs. Can be a CloudWatch Logs log group ARN or Kinesis Data Firehose delivery stream ARN.

```
aws fsx create-file-system \
        --file-system-type WINDOWS \
        --storage-capacity 300 \
        --storage-type SSD \
        --subnet-ids subnet-0123456789abcdef0 \
        --windows-configuration
 "ThroughputCapacity=128,WindowsFsrmConfiguration={FsrmServiceEnabled=true,EventLogDestination=
east-1:123456789012:log-group:/aws/fsx/fsrm}"
```

To modify FSRM configuration on an existing file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems** and choose the Windows file system that you want to modify.
3. Choose the **Administration** tab.
4. In the **File Server Resource Manager** section, choose **Manage**.
5. Make your required changes:

   - To change the event log destination, select a different CloudWatch Logs log group or Kinesis Data Firehose delivery stream
   - To enable FSRM, choose **Enabled**
   - To disable FSRM, choose **Disabled**

   > **Important**
   >
   > Multi-AZ file systems will experience automatic failover and failback events during this process, while Single-AZ file systems will experience a brief period of unavailability.

6. Choose **Save**.

You can monitor the update progress on the File systems detail page, in the **Updates** tab.

To modify FSRM configuration on an existing file system (CLI)

To enable and disable FSRM on an existing FSx for Windows File Server file system, use the AWS CLI command `update-file-system`.

Enabling FSRM

To enable FSRM, include the following FSRM configuration in the `--windows-configuration` parameter:

- `FsrmServiceEnabled` - Set to `true`
- `EventLogDestination` - The Amazon Resource Name (ARN) that specifies the destination of the FSRM event logs. Can be a CloudWatch Logs log group ARN or Kinesis Data Firehose delivery stream ARN.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --windows-configuration
 FsrmConfiguration='{FsrmServiceEnabled=true,EventLogDestination=”arn:aws:logs:us-
east-1:123456789012:log-group:/aws/fsx/fsrm”}'
```

Disabling FSRM

To disable FSRM:

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --windows-configuration FsrmConfiguration='{FsrmServiceEnabled=false}'
```

Important

Multi-AZ file systems will experience automatic failover and failback events during this process, while Single-AZ file systems will experience a brief period of unavailability.

FSx remote PowerShell

To configure and use FSRM features, you must use the Amazon FSx CLI for remote management on PowerShell. For information, see Starting an Amazon FSx remote PowerShell session.

Quota Management

You can use File Server Resource Manager (FSRM) quota management to control the amount of storage space that users consume on your FSx for Windows File Server file system. Quotas help you manage storage capacity by limiting the amount of data that can be stored in specific folders and by generating notifications when storage usage approaches or exceeds defined thresholds.

How quota management works

Quota management provides two types of quotas that you can apply to folders on your file system:

Hard quotas

Prevent users from saving files after the quota limit is reached. When a user attempts to save a file that would exceed the quota limit, the operation fails and the user receives an error message.

Soft quotas

Allow users to exceed the quota limit while logging the violation. Soft quotas are useful for monitoring storage usage without enforcing strict limits.

Quota templates

Quota templates provide a reusable configuration that defines quota settings, including size limits, quota type (hard or soft), and threshold notifications. After you create a quota template, you can apply it to multiple folders without having to reconfigure the same settings each time. When you update a quota template, you can optionally apply the changes to all quotas that were created from that template.

Using quota templates offers several benefits:

- **Consistency** - Ensure that similar folders have identical quota configurations
- **Efficiency** - Apply quota settings to multiple folders quickly
- **Maintainability** - Update quota settings across multiple folders by modifying the template

Auto apply quotas

Auto apply quotas automatically create quotas for subfolders based on a specified template. When you create an auto apply quota on a parent folder, FSRM automatically generates a quota for each existing subfolder and for any new subfolders that users create in the future. This approach is useful for scenarios where you want to apply consistent quota limits across multiple user directories or departmental folders.

Threshold notifications

Thresholds define usage levels at which FSRM takes specific actions. You can configure multiple thresholds for each quota, with each threshold set to a percentage of the quota limit. When storage usage reaches a threshold percentage, FSRM can perform the following actions:

Event logging

> Log an event to Amazon CloudWatch or Amazon Kinesis Data Firehose for monitoring and analysis. You can specify the event severity level (Information, Warning, or Error) and provide a custom message body. Event logging is useful for monitoring quota usage and integrating with existing monitoring systems.

Storage reports

> Generate a storage usage report that provides detailed information about the files and folders consuming storage space. Storage reports help you identify which users or applications are consuming the most storage and make informed decisions about storage management. For more information, see Storage Reports.

You can configure multiple thresholds with different actions for each quota. For example, you might configure a quota with an Information event at 75 percent usage and a Warning event at 90 percent usage.

Quota Management commands

You can access three families of FSx remote PowerShell commands for managing Quotas:

1. **Quota commands** - Create, retrieve, modify, remove, and update quotas on specific folders. Use these commands when you need to manage quotas on a folder-by-folder basis.
2. **Quota Template commands** - Create, retrieve, and modify quota templates that define reusable quota configurations. Use these commands to establish standard quota policies that you can apply across multiple folders.
3. **Auto Quota commands** - Create, retrieve, modify, remove, and update auto apply quotas that automatically generate quotas for subfolders. Use these commands when you need to apply consistent quota limits across multiple subfolders without manually creating individual quotas.

List of Quota Management FSx remote PowerShell commands

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

Quota Commands

New-FSxFSRMQuota

Creates a new quota on a folder. A quota limits the amount of data that users can store in a folder. You can optionally configure the quota to generate notifications when users exceed quota thresholds.

Parameters:

- `Folder (string)` - Required. The folder path where the quota will be applied.
- `Size (string)` - Required when not using a Template: The quota size limit.
- `Template (string)` - Optional. The name of an existing quota template to use. When you specify a template, you can only use the Description parameter; all other settings are inherited from the template.
- `Description (string)` - Optional. A description for the quota.
- `SoftLimit (boolean)` - Optional. If set to true, creates a soft quota that allows users to exceed the limit while logging violations.
- `Disabled (boolean)` - Optional. If set to true, creates the quota in a disabled state.
- `ThresholdConfigurations (array)` - Optional. An array of threshold configurations that specify actions to take at different usage levels. Each configuration has the following properties:
  - `ThresholdPercentage (number)`: The percentage of the quota limit at which to trigger actions. Enter a value between 0 and 250.
  - `Action (array)`: One or more actions to take when the threshold is reached. Each action has the following properties:

- `ActionType`: The type of action to perform. You can specify the following values:
    1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
        - `EventType`: Information, Warning, or Error
        - `MessageBody`: The message text to log with the event.
    2. `Report`: Generates a storage usage report.

Examples:

1. Create a hard 5GB quota without using a quota template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMQuota -Folder "share\test" -Size 5GB
}
```

2. Create soft quota with a threshold notification

```
$thresholds = [System.Collections.ArrayList]@()
$warning = @{
    ThresholdPercentage = 75
    Action = @(
        @{
            ActionType = "Event"
            EventType = "Warning"
            MessageBody = "Quota usage has reached 75%"
        }
    )
}

$thresholds.Add($warning)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList ($thresholds) -ScriptBlock {
        param($thresholds)
        New-FSxFSRMQuota -Folder "share/test" -Size 1GB -Description "Test quota" -
SoftLimit -ThresholdConfigurations $Using:thresholds
}
```

Get-FSxFSRMQuota

Retrieves one or more quotas from your file system. The command returns details about quota configurations, including size limits, thresholds, and current usage.

Parameters:

- `Folder (string)` - Optional. The folder path from which to retrieve quotas. If you don't specify a folder path, the command returns all quotas on the file system.

Examples:

1. Get all existing quotas on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMQuota
}
```

Remove-FSxFSRMQuota

Removes a quota from a specified folder on your file system.

Parameters:

- `Folder (string)` - Required. The folder path from which to remove the quota.
- `PassThru (boolean)` - Optional. If set to true, returns the removed quota object.

Examples:

1. Remove a quota.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMQuota -Folder "share\test" -PassThru
}
```

Set-FSxFSRMQuota

Modifies the configuration of an existing quota.

Parameters:

- `Folder (string)` - Required. The folder path that contains the quota to modify.
- `Description (string)` - Optional. A new description for the quota.
- `Size (string)` - Optional. The new size limit for the quota.
- `SoftLimit (boolean)` - Optional. If set to true, changes the quota to a soft limit, allowing users to exceed the limit while logging violations.
- `Disabled (boolean)` - Optional. If set to true, disables the quota. If set to false, enables the quota.
- `ThresholdConfigurations (array)` - Optional. An array of new threshold configurations. Each threshold configuration has the following properties:
  - `ThresholdPercentage (number)`: The percentage of the quota limit at which to trigger actions. Enter a value between 0 and 250.
  - `Action (array)`: One or more actions to take when the threshold is reached. Each action has the following properties:
    - `ActionType`: The type of action to perform. You can specify the following values:
      1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
         - `EventType`: Information, Warning, or Error
         - `MessageBody`: The message text to log with the event.
      2. `Report`: Generates a storage usage report.
- `PassThru (boolean)` - Optional. If set to true, returns the modified quota object.

Examples:

1. Modify quota size and description.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMQuota  -Folder "share\department"  -Size 2GB  -Description "Updated
 quota for department share"
}
```

2. Disable a Quota

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMQuota -Folder "share\department" -Disabled: $true
}
```

Update-FSxFSRMQuota

Recalculates the current usage statistics for a quota by scanning the folder to determine the actual amount of space being used.

Parameters:

- `Folder (string)` - Required. The folder path that contains the quota to update.
- `PassThru (boolean)` - Optional. If set to true, returns the updated quota object.

Examples:

1. Recalculates the current usage statistics for a specified quota.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Update-FSxFSRMQuota -Folder "share\department" -PassThru
}
```

Quota Template Commands

New-FSxFSRMQuotaTemplate

Creates a new quota template that defines a reusable configuration for quotas.

Parameters:

- `Name (string)` - Required. A name for the quota template.
- `Size (string)` - Required. The size limit that the quota template enforces.
- `Description (string)` - Optional. A description for the quota template.
- `SoftLimit (boolean)` - Optional. If set to true, creates a template for soft quotas that report usage but don't enforce the limit.
- `ThresholdConfigurations (array)` - Optional. An array of threshold configurations that specify actions to take at different usage levels. Each configuration has the following properties:
  - `ThresholdPercentage (number)`: The percentage of the quota limit at which to trigger actions. Enter a value between 0 and 250.

- `Action (array)`: One or more actions to take when the threshold is reached. Each action has the following properties:
  - `ActionType`: The type of action to perform. You can specify the following values:
    1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
       - `EventType`: Information, Warning, or Error
       - `MessageBody`: The message text to log with the event.
    2. `Report`: Generates a storage usage report.

Examples:

1. Create hard 1 GB limit template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMQuotaTemplate -Name "1GB Hard Limit" -Size 1GB -Description "Standard
 1GB hard limit template"
}
```

2. Create a 5 GB soft limit template with a warning threshold at 90% usage

```
$threshold = @{
    ThresholdPercentage = 90
    Action = @(
        @{
            ActionType = "Event"
            EventType = "Warning"
            MessageBody = "Quota usage has reached 90% of the limit"
        }
    )
}

$thresholds = [System.Collections.ArrayList]@()
$thresholds.Add($threshold)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $thresholds -ScriptBlock {
    param($thresholds)
    New-FSxFSRMQuotaTemplate -Name "5GB Soft Limit" -Size 5GB -Description "5GB soft
 limit with 90% warning" -SoftLimit -ThresholdConfigurations $Using:thresholds
}
```

Get-FSxFSRMQuotaTemplate

Retrieves one or more quota templates from your file system.

Parameters:

- `Name (string)` - Optional. The name of a specific quota template to retrieve. If you don't specify a name, the command returns all quota templates.

Examples:

1. Retrieve all quota templates on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMQuotaTemplate
}
```

Set-FSxFSRMQuotaTemplate

Modifies the properties of a quota template.

Parameters:

- `Name (string)` - Required. The name of the quota template to modify.
- `Description (string)` - Optional. A new description for the template.
- `Size (string)` - Optional. The new size limit for the template.
- `SoftLimit (boolean)` - Optional. If set to true, changes the template to create soft quotas that report usage but don't enforce the limit.
- `ThresholdConfigurations (array)` - Optional. An array of threshold configurations that specify actions to take at different usage levels. Each configuration has the following properties:
  - `ThresholdPercentage (number)`: The percentage of the quota limit at which to trigger actions. Enter a value between 0 and 250.
  - `Action (array)`: One or more actions to take when the threshold is reached. Each action has the following properties:

  - `ActionType`: The type of action to perform. You can specify the following values:
    1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
       - `EventType`: Information, Warning, or Error
       - `MessageBody`: The message text to log with the event.
    2. `Report`: Generates a storage usage report.
- `UpdateDerived (boolean)` - Optional. If set to true, updates all quotas that were created from this template.
- `UpdateDerivedMatching (boolean)` - Optional. If set to true, updates only quotas that were created from this template and have not been modified since creation.
- `PassThru (boolean)` - Optional. If set to true, returns the modified template object.

Examples:

1. Modifies the size and description of a quota template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMQuotaTemplate -Name "5GB Soft Limit" -Size 10GB -Description "Updated to
 10GB soft limit"
}
```

2. Modifies a quota template and updates all quotas that were created from the template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMQuotaTemplate -Name "1GB Hard Limit" -Size 2GB -UpdateDerived
}
```

Reset-FSxFSRMQuota

Resets a quota to match the settings of a specified template.

Parameters

- `Folder (string)` - Required. The folder path that contains the quota to reset.
- `Template (string)` - Required. The name of the quota template to apply.

Examples

Examples: Reset a quota to match the settings defined in a quota template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Reset-FSxFSRMQuota -Folder "share\department" -Template "1GB Hard Limit"
}
```

Auto Quota Commands

New-FSxFSRMAutoQuota

The `New-FSxFSRMAutoQuota` command creates an auto apply quota on a specified folder. An auto apply quota automatically generates quotas based on the specified template for each existing subfolder and any new subfolders created in the specified folder.

Parameters

- `Folder (string)` - Required. The folder path where the auto apply quota will be created.
- `Template (string)` - Optional. The name of an existing quota template to use for the auto apply quota.
- `Disabled (boolean)` - Optional. If set to true, creates the auto apply quota in a disabled state.

Examples

1. Create an auto apply quota that automatically applies a specified template to all subfolders.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMAutoQuota -Folder "share\department" -Template "250 MB Extended Limit"
}
```

Get-FSxFSRMAutoQuota

The `Get-FSxFSRMAutoQuota` command retrieves one or more auto apply quotas from your file system.

Parameters

- `Folder (string)` - Optional. The folder path from which to retrieve auto apply quotas. You can also use `...` at the end of the path to include all subfolders.

If you don't specify a folder path, the command returns all auto apply quotas on the file system.

Examples

1. Retrieve all auto apply quotas on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMAutoQuota
}
```

Remove-FSxFSRMAutoQuota

The `Remove-FSxFSRMAutoQuota` command removes an auto-apply quota from a specified folder. When you remove an auto apply quota, the command also removes all quotas from subfolders that were derived from the associated quota template.

Parameters

- `Folder (string)` - Required. The folder path from which to remove the auto apply quota.
- `PassThru (boolean)` - Optional. If set to true, returns the removed auto apply quota object.

Examples

1. Remove an auto apply quota from a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMAutoQuota -Folder "share\department" -PassThru
}
```

Set-FSxFSRMAutoQuota

The `Set-FSxFSRMAutoQuota` command modifies the configuration settings of an auto apply quota.

Parameters

- `Folder (string)` - Required. The folder path that contains the auto apply quota to modify.
- `Template (string)` - Optional. The name of a quota template to apply.
- `Disabled (boolean)` - Optional. If set to true, disables the auto apply quota. If set to false, enables the auto apply quota.
- `UpdateDerived (boolean)` - Optional. If set to true, updates all existing quotas that were derived from this auto apply quota.
- `UpdateDerivedMatching (boolean)` - Optional. If set to true, updates only derived quotas that have not been modified since creation.
- `PassThru (boolean)` - Optional. If set to true, returns the modified auto apply quota object.

Examples

1. Change the quota template used by an auto apply quota.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMAutoQuota -Folder "share\department" -Template "100 MB Limit"
}
```

2. Disable an auto apply quota and update all quotas that were derived from it.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMAutoQuota -Folder "share\department" -Disabled: $true -UpdateDerived
}
```

Update-FSxFSRMAutoQuota

The `Update-FSxFSRMAutoQuota` command recalculates the properties of an auto apply quota and the quotas that are derived from it by scanning the folder to determine the actual amount of space being used.

Parameters

- `Folder (string)` - Required. The folder path that contains the auto apply quota to update.
- `PassThru (boolean)` - Optional. If set to true, returns the updated auto apply quota object.

Examples

1. Recalculate the usage statistics and return the updated auto apply quota object.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Update-FSxFSRMAutoQuota -Folder "share\department" -PassThru
}
```

File Groups

File groups define logical collections of file name patterns that you must use when configuring file screens, and that you can optionally use when generating storage reports. A file group contains include patterns (files to match) and exclude patterns (files to exclude from matches), which you reference by the file group name rather than specifying individual patterns each time.

How file groups are used

File groups are required for the following FSRM features:

- **File screens** - You must specify one or more file groups to define which file types to block or monitor.
- **File screen exceptions** - You must specify one or more file groups to define which file types to allow despite blocking file screens in parent folders.
- **File screen templates** - You must specify one or more file groups to define which file types the template will block or monitor.

File groups are optional for the following FSRM features:

- **Storage reports** - You can optionally filter reports by file group to analyze storage usage for specific file types. For example, you can generate a report showing only audio and video files.

File name patterns

File groups use wildcard patterns to match file names. You can specify both include patterns (files to match) and exclude patterns (files to exclude from matches).

FSRM supports the following wildcards:

- **Asterisk (*)** - Matches zero or more characters
- **Question mark (?)** - Matches exactly one character

For example, the pattern `*.doc*` matches files like `report.doc` , `report.docx`, and `document.doc`, while the exclude pattern `~$*` excludes temporary files created by Microsoft Office applications.

Default file groups

When you enable FSRM on your file system, the following file groups are created automatically:

Audio and Video Files

Matches common audio and video file formats including `*.mp3`, `*.wav`, `*.avi`, `*.mp4`, `*.mpeg`, and `*.wmv`

Backup Files

Matches backup file formats including `*.bak`, `*.backup`, and `*.old`

Compressed Files

Matches archive and compressed file formats including `*.zip`, `*.rar`, `*.7z`, `*.gz`, and `*.tar`

E-mail Files

Matches email message and mailbox formats including `*.eml`, `*.msg`, and `*.pst`

Executable Files

Matches executable and script file formats including `*.exe`, `*.dll`, `*.com`, `*.bat`, `*.cmd`, and `*.vbs`

Image Files

Matches common image file formats including `*.jpg`, `*.jpeg`, `*.png`, `*.gif`, `*.bmp`, and `*.tif`

Office Files

Matches Microsoft Office document formats including `*.doc`, `*.docx`, `*.xls`, `*.xlsx`, `*.ppt`, and `*.pptx`

System Files

Matches Windows system file formats including `*.sys`, `*.dll`, `*.ocx`, and `*.drv`

Temporary Files

Matches temporary file formats including `*.tmp`, `*.temp`, and `~*`

Text Files

Matches text-based file formats including `*.txt`, `*.log` , `*.csv`, and `*.xml`

Web Page Files

Matches web content file formats including `*.html`, `*.htm`, `*.asp`, `*.aspx`, `*.php`, and `*.js`

You can use these default file groups immediately in file screens and storage reports, or you can modify them to match your specific requirements.

File group management commands

FSRM provides PowerShell commands for creating and managing file groups. Use these commands to define custom file groups that match your organization's file management policies.

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the AWS FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

New-FSxFSRMFileGroup

Creates a file group that defines a logical collection of file name patterns. These patterns can be used for file screens, file screen exceptions, and storage reports.

Parameters:

- `Name (string)` - Required. A name for the file group.
- `Description (string)` - Optional. A description for the file group.
- `IncludePattern (array)` - Optional. An array of pattern strings that specify files to include.

- `ExcludePattern (array)` - Optional. An array of pattern strings that specify files to exclude.

Examples:

1. Create a file group for text files.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMFileGroup -Name "My Text Files" -IncludePattern "*.txt"
}
```

2. Create a file group for source code with include and exclude patterns.

```
$includePatterns = @("*.cpp", "*.h", "*.cs", "*.py")
$excludePatterns = @("*.tmp", "*.bak")

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList @($includePatterns, $excludePatterns) -ScriptBlock {
    param($includePatterns, $excludePatterns)
    New-FSxFSRMFileGroup -Name "Source Code" -Description "Programming source files"
 -IncludePattern $includePatterns -ExcludePattern $excludePatterns
}
```

Get-FSxFSRMFileGroup

Retrieves one or more file groups from your file system. File groups define collections of file patterns used in file screening and reporting.

Parameters:

- `Name (array)` - Optional. An array of names of file groups to retrieve. If you don't specify a name, the command returns all file groups on the file system.

Examples:

1. Retrieve all file groups on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileGroup
}
```

Remove-FSxFSRMFileGroup

Removes one or more file groups from your file system. After removal, the file group cannot be used in file screens or file screen exceptions.

Parameters:

- `Name (array)` - Required. An array of names of file groups to remove.
- `PassThru (boolean)` - Optional. If set to true, returns the removed file group object.

Examples:

1. Remove a single file group.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMFileGroup -Name "My Text Files" -PassThru
}
```

Set-FSxFSRMFileGroup

Modifies the properties of existing file groups.

Parameters:

- `Name (array)` - Required. An array of names of file groups to modify.
- `Description (string)` - Optional. A new description for the file group.
- `IncludePattern (array)` - Optional. A new array of pattern strings that specify files to include.
- `ExcludePattern (array)` - Optional. A new array of pattern strings that specify files to exclude.
- `PassThru (boolean)` - Optional. If set to true, returns the modified file group object.

Examples:

1. Update the description and patterns for a file group.

```
$includePatterns = @("*.docx", "*.pdf", "*.rtf")
$excludePatterns = @("~$*", "*.tmp")

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList @($includePatterns, $excludePatterns) -ScriptBlock {
    param($includePatterns, $excludePatterns)
    Set-FSxFSRMFileGroup -Name "Documents" -Description "Updated document types" -
IncludePattern $includePatterns -ExcludePattern $excludePatterns -PassThru
}
```

File Screening

File screening controls which types of files users can save to folders on your file system. File screening helps you enforce storage policies, prevent unauthorized file types, and maintain compliance with organizational requirements.

> **Note**
>
> File screens use file groups to define which file types to block or monitor. For more information about creating and managing file groups, see File Groups.

FSRM supports two types of file screens:

1. **Active file screens** - Block users from saving files that match the specified file groups and generate notifications when users attempt to save blocked files. Use active file screens when you need to enforce strict policies about which file types are allowed in specific folders.
2. **Passive file screens** - Monitor and log when users save files that match the specified file groups, but do not prevent the save operation. Use passive file screens when you want to track file usage patterns without disrupting user workflows.

File screen templates

File screen templates provide a reusable configuration that defines file screening settings, including which file groups to block or monitor and what notifications to generate. After you create a file screen template, you can apply it to multiple folders without having to reconfigure the same settings each time. When you update a file screen template, you can optionally apply the changes to all file screens that were created from that template.

Using file screen templates offers several benefits:

- **Consistency** - Ensure that similar folders have identical file screening configurations
- **Efficiency** - Apply file screening settings to multiple folders quickly
- **Maintainability** - Update file screening settings across multiple folders by modifying the template

File screen exceptions

File screen exceptions override file screening rules that would otherwise apply to a folder and all its subfolders. When you create a file screen exception, you specify which file groups to allow despite any blocking file screens in parent folders. File screen exceptions are useful when you need to permit specific file types in certain subfolders while maintaining broader restrictions at higher levels of the folder hierarchy.

For example, you might block executable files across an entire share but create an exception for a specific subfolder where administrators need to store installation files.

File screening notifications

When users attempt to save files that are blocked by an active file screen, FSRM can generate notifications to alert administrators or provide information to users. You can configure the following types of notifications:

- **Event logging** - Log an event to Amazon CloudWatch or Amazon Kinesis Data Firehose for monitoring and analysis. You can specify the event's severity level (Information, Warning, or Error) and provide a custom message body. Event logging is useful for tracking file screen violations and integrating with existing monitoring systems.
- **Storage reports** - Generate a storage usage report that provides detailed information about file screening activity. Storage reports help you identify patterns in file save attempts and make informed decisions about file screening policies. For more information, see Storage Reports.

File screening management commands

You can access three families of FSx remote PowerShell commands for managing file screens:

1. **File screen commands** - Create, retrieve, modify, remove, and reset individual file screens on specific folders. Use these commands when you need to manage file screens on a folder-by-folder basis.
2. **File screen template commands** - Create, retrieve, modify, and remove file screen templates that define reusable file screening configurations. Use these commands to establish standard file screening policies that you can apply across multiple folders.
3. **File screen exception commands** - Create, retrieve, modify, and remove file screen exceptions that override file screening rules in parent folders. Use these commands when you need to allow specific file types in certain subfolders while maintaining broader restrictions.

List of File Screening FSx remote PowerShell commands

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

File screen commands

New-FSxFSRMFileScreen

Creates a file screen that blocks users from saving specified types of files to a folder.

Parameters:

- `Folder (string)` - Required. The folder path where the file screen will be applied.
- `Description (string)` - Optional. A description for the file screen.
- `IncludeGroup (array)` - Optional. An array of file group names that specify which files to block or monitor.
- `Active (boolean)` - Optional. If set to true, creates an active file screen that blocks files. If set to false, creates a passive file screen that only monitors files. Default is true.
- `Template (string)` - Optional. The name of an existing file screen template to use.

- `NotificationConfigurations (array)` - Optional. An array of configurations for notifications when users attempt to save blocked files. Each configuration has the following properties:
  - `ActionType (string)`: The type of action to perform. You can specify the following values:
    1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
       - `EventType (string)`: Information, Warning, or Error
       - `MessageBody (string)`: The message text to log with the event.
    2. `Report`: Generates a storage usage report. When you specify Report, you must also specify:
       - `ReportType (string)`: The type of report. You can specify the following values: `DuplicateFiles`, `FilesByFileGroup`, `FilesByOwner`, `FilesByProperty`, `LargeFiles`, `LeastRecentlyAccessed`, `MostRecentlyAccessed`, or `QuotaUsage`.

Examples:

1. Create a basic active file screen that blocks Audio Files.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMFileScreen -Folder "share\department" -IncludeGroup "Audio and Video
 Files"
}
```

2. Create a file screen that blocks video files and generates an event log entry when a user attempts to save a video file.

```
$notifications = [System.Collections.ArrayList]@()
$eventNotification = @{
    ActionType = "Event"
    EventType = "Warning"
    MessageBody = "File screen violation detected"
}
$null = $notifications.Add($eventNotification)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $notifications -ScriptBlock {
    param($notifications)
    New-FSxFSRMFileScreen -Folder "share\projects" -IncludeGroup "Audio and Video
 Files" -NotificationConfigurations $Using:notifications
}
```

Get-FSxFSRMFileScreen

Retrieves one or more file screens from your file system.

Parameters:

- `Folder (string)` - Optional. The folder path from which to retrieve file screens. If you don't specify a folder path, the command returns all file screens on the file system.

Examples:

1. Retrieve all file screens on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileScreen
}
```

Set-FSxFSRMFileScreen

Modifies the properties of an existing file screen.

Parameters:

- `Folder (string)` - Required. The folder path that contains the file screen to modify.
- `Description (string)` - Optional. A new description for the file screen.
- `IncludeGroup (array)` - Optional. A new array of file group names that define which files to block or monitor.
- `Active (boolean)` - Optional. If set to true, sets the file screen to active mode (blocking). If set to false, sets the file screen to passive mode (monitoring only). Default is true.
- `NotificationConfigurations (array)` - Optional. A new array of notification configurations.
- `PassThru (boolean)` - Optional. If set to true, returns the modified file screen object.

Examples:

1. Modify the description and file groups for a file screen.

```
$includeGroups = @("Audio and Video Files")
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $includeGroups -ScriptBlock {
    param($includeGroups)
    Set-FSxFSRMFileScreen -Folder "share\projects" -Description "Updated screen" -
IncludeGroup $includeGroups
}
```

2. Set a file screen to active mode and add notifications.

```
$notifications = [System.Collections.ArrayList]@()
$eventNotification = @{
    ActionType = "Event"
    EventType = "Warning"
    MessageBody = "File screen violation detected"
}
$null = $notifications.Add($eventNotification)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $notifications -ScriptBlock {
    param($notifications)
    Set-FSxFSRMFileScreen -Folder "share\projects" -Active: $true -
NotificationConfigurations $Using:notifications -PassThru
}
```

Remove-FSxFSRMFileScreen

Removes a file screen from a specified folder.

Parameters:

- `Folder (string)` - Required. The folder path from which to remove the file screen.
- `PassThru (boolean)` - Optional. If set to true, returns the removed file screen object.

Examples:

1. Remove a file screen from a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMFileScreen -Folder "share\projects" -PassThru
}
```

Reset-FSxFSRMFileScreen

Resets a file screen to match the settings of a specified template.

Parameters:

- `Folder (string)` - Required. The folder path that contains the file screen to reset.
- `Template (string)` - Required. The name of an existing file screen template to apply.

Examples:

1. Reset a file screen to match the settings defined in a file screen template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Reset-FSxFSRMFileScreen -Folder "share\department" -Template "Block Audio Files"
}
```

File Screen Template Commands

Get-FSxFSRMFileScreenTemplate

The `Get-FSxFSRMFileScreenTemplate` command retrieves one or more file screen templates from your file system.

Parameters

- `Name (array)` - Optional. An array of names of file screen templates to retrieve. If you don't specify a name, the command returns all file screen templates on the file system.

Examples

1. Retrieve all file screen templates.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileScreenTemplate
}
```

New-FSxFSRMFileScreenTemplate

The `New-FSxFSRMFileScreenTemplate` command creates a file screen template that defines a reusable configuration for file screens. The template specifies which file groups to block and what notifications to generate when users attempt to save blocked files.

Parameters

- `Name (string)` - Required. A name for the file screen template.
- `Description (string)` - Optional. A description for the file screen template.
- `IncludeGroup (array)` - Optional. An array of file group names that specify which files to block or monitor.
- `Active (boolean)` - Optional. If set to true, creates an active file screen template that blocks files. If set to false, creates a passive template that only monitors files. Default is true.
- `NotificationConfigurations (array)` - Optional. An array of configurations for notifications when users attempt to save blocked files. Each configuration has the following properties:
  - `ActionType (string)`: The type of action to perform. You can specify the following values:
    1. `Event`: Logs an event to the file system's event log. When you specify Event, you must also specify the following properties:
       - `EventType (string)`: Information, Warning, or Error
       - `MessageBody (string)`: The message text to log with the event.
    2. `Report`: Generates a storage usage report. When you specify Report, you must also specify:
       - `ReportType (string)`: The type of report. You can specify the following values: `DuplicateFiles`, `FilesByFileGroup`, `FilesByOwner`, `FilesByProperty`, `LargeFiles`, `LeastRecentlyAccessed`, `MostRecentlyAccessed`, or `QuotaUsage`

Examples

1. Create a file screen template with notifications.

```
$notifications = [System.Collections.ArrayList]@()
$eventNotif = @{
    ActionType = "Event"
    EventType = "Warning"
    MessageBody = "Blocked file detected"
}
$null = $notifications.Add($eventNotif)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $notifications -ScriptBlock {
    param($notifications)
    New-FSxFSRMFileScreenTemplate -Name "Block Executables" -Description
 "Blocks executable files" -IncludeGroup "Executable Files" -Active: $true -
NotificationConfigurations $Using:notifications
}
```

Remove-FSxFSRMFileScreenTemplate

The `Remove-FSxFSRMFileScreenTemplate` command removes one or more file screen templates from your file system. When you remove a template, file screens that were created from that template remain unchanged.

Parameters

- `Name (array)` - Required. An array of names of file screen templates to remove.
- `PassThru (boolean)` - Optional. If set to true, returns the removed file screen template object.

Examples

1. Remove a single file screen template.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMFileScreenTemplate -Name "Block Executables" -PassThru
}
```

Set-FSxFSRMFileScreenTemplate

The `Set-FSxFSRMFileScreenTemplate` command modifies the properties of existing file screen templates. Optionally updates file screens that were created using the modified templates.

Parameters

- `Name (array)` - Required. An array of names of file screen templates to modify.
- `Description (string)` - Optional. A new description for the template.
- `IncludeGroup (array)` - Optional. A new array of file group names that define which files to block or monitor.
- `Active (boolean)` - Optional. If set to true, sets the template to active mode (blocking). If set to false, sets the template to passive mode (monitoring). Default is true.
- `NotificationConfigurations (array)` - Optional. A new array of notification configurations.
- `UpdateDerived (boolean)` - Optional. If set to true, updates all existing file screens created from this template, regardless of any modifications made to those file screens.
- `UpdateDerivedMatching (boolean)` - Optional. If set to true, updates only file screens that have not been modified since their creation from this template.
- `PassThru (boolean)` - Optional. If set to true, returns the modified file screen template object.

Examples

1. Update a file screen template with new file groups.

```
$includeGroups = @("Audio and Video Files")
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $includeGroups -ScriptBlock {
    param($includeGroups)
    Set-FSxFSRMFileScreenTemplate -Name "Block Executables" -IncludeGroup
 $includeGroups
}
```

2. Update a file screen template to active mode and update all derived file screens.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMFileScreenTemplate -Name "Block Executables" -Active: $true -
UpdateDerived
}
```

File Screen Exception Commands

New-FSxFSRMFileScreenException

The `New-FSxFSRMFileScreenException` command creates a file screen exception that overrides any file screening rules that would otherwise apply to a folder and all its subfolders. This allows specific file types to be created in the exception folder even if they are blocked by file screens in parent folders.

Parameters

- `Folder (string)` - Required. The folder path where the file screen exception will be applied. The exception applies to this folder and all its subfolders.
- `Description (string)` - Optional. A description for the file screen exception.
- `IncludeGroup (array)` - Optional. An array of file group names that specify which files to allow despite any blocking file screens that would otherwise apply from parent folders.

Examples

1. Create a file screen exception for a specific folder and file group.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMFileScreenException -Folder "share\department" -IncludeGroup "Text
 Files"
}
```

2. Create a file screen exception with multiple file groups.

```
$includeGroups = @("Audio and Video Files", "Documents")
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $includeGroups -ScriptBlock {
    param($includeGroups)
    New-FSxFSRMFileScreenException -Folder "share\projects" -Description "Allow media
 files in project folder" -IncludeGroup $includeGroups
    }
```

Get-FSxFSRMFileScreenException

The `Get-FSxFSRMFileScreenException` command retrieves one or more file screen exceptions from your file system.

Parameters

- `Folder (string)` - Optional. The folder path from which to retrieve file screen exceptions. If you don't specify a folder path, the command returns all file screen exceptions on the file system.

Examples

1. Retrieve all file screen exceptions on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileScreenException
}
```

2. Retrieve the file screen exception for a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileScreenException -Folder "share\department"
}
```

Remove-FSxFSRMFileScreenException

The `Remove-FSxFSRMFileScreenException` command removes a file screen exception from a specified folder. After removal, the folder and its subfolders will be subject to any file screening rules from parent folders that were previously overridden by the exception.

Parameters

- `Folder (string)` - Required. The folder path from which to remove the file screen exception.
- `PassThru (boolean)` - Optional. If set to true, returns the removed file screen exception object.

Examples

1. Remove a file screen exception from a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMFileScreenException -Folder "share\projects" -PassThru
}
```

Set-FSxFSRMFileScreenException

The `Set-FSxFSRMFileScreenException` command modifies the properties of a file screen exception.

Parameters

- `Folder (string)` - Required. The folder path that contains the file screen exception to modify.
- `Description (string)` - Optional. A new description for the file screen exception.
- `IncludeGroup (array)` - Optional. A new array of file group names that define which files to allow despite any blocking file screens that would otherwise apply from parent folders.
- `PassThru (boolean)` - Optional. If set to true, returns the modified file screen exception object.

Examples

1. Update the allowed file groups for a file screen exception.

```
$includeGroups = @("Audio and Video Files")
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $includeGroups -ScriptBlock {
    param($includeGroups)
    Set-FSxFSRMFileScreenException -Folder "share\projects" -IncludeGroup
 $includeGroups -PassThru
}
```

File Classification

File classification automatically assigns metadata properties to files based on their content, location, or other attributes. Classification helps you organize files, enforce data management policies, and meet compliance requirements by identifying files that contain sensitive information, belong to specific business categories, or require retention periods.

How file classification works

File classification uses a three-step process:

1. **Define properties** - Create classification property definitions that specify the types of metadata you want to assign to files, such as "`Data Sensitivity`" or "`ContainsPII`".
2. **Create rules** - Configure classification rules that automatically assign property values to files based on criteria you specify, such as file content patterns or folder locations. For example, files that contain a pattern such as a Social Security Number `(XXX-XX-XXXX)` can be automatically classified as `ContainsPII=Yes`.
3. **Run classification** - Execute the classification process to scan files and apply the rules. You can run classification manually on demand, on a schedule, or continuously in the background.

After classification is completed, you can use the assigned properties to generate storage reports, apply File Management Tasks, or search for files with specific characteristics.

Classification property definitions

Classification property definitions specify the types of metadata that can be assigned to files. Each property definition has a name, a property type, and optionally a list of allowed values. For example, you might create a property called "`Data Sensitivity`" with an `OrderedList` Type and possible values: `Public`, `Internal`, `Confidential`, and `Restricted`.

The following property types are supported:

- `OrderedList` - An ordered list where values have a specific sequence (for example, Low, Medium, High). Use this type when the order of values matters for reporting or policy decisions.
- `MultiChoice` - Allows multiple values to be selected from a list (for example, a file might be tagged with both "Financial" and "Legal" categories).
- `SingleChoice` - Allows only one value to be selected from a list.
- `String` - A single text value with no predefined options.
- `MultiString` - Multiple text values with no predefined options.
- `Integer` - A numeric value.
- `YesNo` - A boolean value (true or false).
- `DateTime` - A date and time value.

Property definitions are reusable across multiple classification rules. After you create a property definition, you can reference it in any classification rule that needs to assign values for that property.

Classification rules

Classification rules define the logic for automatically assigning property values to files. Each rule specifies:

- Which property to set
- What value to assign to that property
- Where to apply the rule (which folders)
- How to identify files that should receive the property value. You can use two classification mechanisms:

Content Classifier

Content Classifier scans file content for specific text patterns or regular expressions. Use this mechanism to identify files based on what they contain. Content Classifier provides three ways to match file content:

- `ContentString` - Searches for case-insensitive text strings. Use this option when you want to find specific words or phrases regardless of capitalization. For example, searching for "confidential" will match "Confidential", "CONFIDENTIAL", and "confidential".
- `ContentStringCaseSensitive` - Searches for case-sensitive text strings. Use this option when capitalization matters for your search. For example, searching for "SSN" will match "SSN" but not "ssn" or "Ssn". This is useful for acronyms, product codes, or other identifiers where case is significant.
- `ContentRegularExpression` - Searches for patterns using regular expressions. Use this option when you need to match complex patterns or variable formats. For example, you can use regular expressions to detect:
  - Social Security numbers in the format 123-45-6789: `\b\d{3}-\d{2}-\d{4}\b`
  - Credit card numbers with optional spaces or dashes: `\b\d{4}[\s-]?\d{4}[\s-]?\d{4}[\s-]?\d{4}\b`
  - Email addresses, phone numbers, or other structured data

You can specify multiple strings or patterns in a single rule and files will be classified if their content matches any of the specified values.

Folder Classifier

Folder Classifier assigns property values based on where files are stored. Use this mechanism to classify files by their location in the folder hierarchy. For example:

- Set a retention period property for all files in the Legal Documents folder
- Mark all files in a specific project folder with a project identifier

Additionally, you can use the `ReevaluateProperty` parameter to control what happens when classification runs on a file that already has a value for the property. You can select the following configurations:

- `Never` - Only classify files that don't have a value for this property
- `Overwrite` - Replace existing values when files change
- `Aggregate` - Combine new values with existing values (for multi-value properties)

Management properties

Management properties are classification properties that apply to folders instead of files. You use management properties to organize and categorize folders in your file system hierarchy. Unlike file properties that are assigned automatically through classification rules, you set management properties manually using the `Set-FSxFSRMMgmtProperty` command.

To classify folders, use the `FolderUsage_MS` property. You can specify the following values:

- `User Files`
- `Group Share`
- `Application Files`
- `Backup and Archival`

Running classification

You can run file classification in three ways:

1. **Manual classification** - Use Start-FSxFSRMClassification to run classification immediately. This approach is useful for testing new rules or performing one-time classification tasks.

2. **Scheduled classification** - Use Set-FSxFSRMClassification to configure a schedule for automatic classification. You can schedule classification to run weekly or monthly at specific times. Scheduled classification is appropriate for most production environments where you want regular, predictable classification runs.
3. **Continuous classification** - Use Set-FSxFSRMClassification with the `Continuous` parameter to enable background classification that runs continuously. Continuous classification automatically classifies new and modified files shortly after they're created or changed. This approach provides the most up-to-date classification but consumes more system resources.

When you start classification, you can specify a `RunDuration` to limit how long the process runs. If classification doesn't complete within the specified time, it stops and resumes during the next scheduled run or when you manually start it again.

After classification completes, you can view the classification properties assigned to files by right clicking a file in Windows File Explorer, selecting **Properties**, and choosing the **Classification** tab. This tab displays all classification properties and their values for the file.

Classification process management

You can monitor and control the classification process with the following commands:

- Get-FSxFSRMClassification - Check the current status of classification (`Running`, `Queued`, `NotRunning`, or `Unknown`)
- Stop-FSxFSRMClassification - Stop a running or queued classification job
- Wait-FSxFSRMClassification - Pause script execution until classification completes or a timeout expires

Use these commands to coordinate classification with other tasks. For example, you might wait for classification to complete before generating a storage report that depends on classified file properties.

Classification best practices

Follow these best practices to ensure efficient and effective file classification.

1. Performance considerations

Content-based classification is resource-intensive because FSRM must read and scan file content.

- **Test rules on a small dataset first** - Before applying classification rules to your entire file system, test them on a representative sample of files to verify they work as expected and to estimate how long classification will take.
- **Limit content scanning scope** - Content-based classification is resource-intensive because it requires reading file content. Use the `Namespace` parameter to limit rules to specific folders rather than scanning the entire file system.
- **Use folder classification when possible** - Folder Classifier is much faster than Content Classifier because it doesn't need to read file contents. When files can be classified based on their location, use the Folder Classifier instead of Content Classifier.
- **Schedule classification during off-peak hours** - Run scheduled classification during periods of low system activity to minimize impact on user performance. Avoid running classification during backup windows or other maintenance tasks.
- **Set appropriate RunDuration limits** - Use the `RunDuration` parameter to prevent classification from running too long and impacting system performance. If classification doesn't complete within the time limit, it will resume during the next scheduled run.
- **Monitor classification performance** - Use `Get-FSxFSRMClassification` to check classification status and identify if classification is taking longer than expected. Long-running classification may indicate that rules need to be optimized or that the system needs more resources.

2. Rule design

- **Use specific regular expressions** - When using `ContentRegularExpression`, write patterns that are as specific as possible to avoid false matches. Test regular expressions thoroughly before deploying them in production.
- **Combine multiple patterns efficiently** - Instead of creating separate rules for similar patterns, combine them into a single rule with multiple `ContentString` or `ContentRegularExpression` values. This reduces the number of times FSRM needs to scan each file.
- **Exclude unnecessary folders** - Use the `ExcludeNamespace` parameter in `Set-FSxFSRMClassification` to exclude temporary directories, and other locations that don't need classification.

3. Property management

- **Plan your property schema** - Design your classification properties before creating rules. Consider what properties you need for reporting, compliance, and file management policies.
- **Document property definitions** - Use the Description field to explain what each property means and how it should be used. This helps other administrators understand your classification schema.

4. Ongoing maintenance

- **Review classification results regularly** - Generate storage reports to verify that classification is working as expected and that files are receiving the correct property values.
- **Update rules as needed** - As your organization's data management requirements change, update classification rules to reflect new policies or compliance requirements.
- **Clean up unused properties** - Remove property definitions and rules that are no longer needed to keep your classification configuration manageable.

Classification management commands

You can access four families of FSx remote PowerShell commands for managing file classification:

1. **Property definition commands** - Create and manage classification property definitions that specify the types of metadata you can assign to files.
2. **Classification rule commands** - Create and manage automatic classification rules that assign property values based on file content or location.
3. **Management property commands** - Set and retrieve classification properties on folders rather than files.
4. **Classification process commands** - Start, stop, monitor, and configure the classification process.

List of File Classification FSx remote PowerShell commands

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console

on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

Property Definition Commands

New-FSxFSRMClassificationPropertyDefinition

`New-FSxFSRMClassificationPropertyDefinition`: Creates a classification property definition that can be used to classify files. Property definitions define the attributes that can be assigned to files through classification rules.

Parameters:

- `Name (string)` - Required. A name for the property definition.
- `DisplayName (string)` - Optional. A display name for the property definition.
- `Description (string)` - Optional. A description for the property definition.
- `Type (string)` - Required. The type of the classification property. You can specify the following values:
  - `OrderedList`: An ordered list of possible values
  - `MultiChoice`: Multiple choice selection from possible values
  - `SingleChoice`: Single choice from possible values
  - `String`: Single text string
  - `MultiString`: Multiple text strings
  - `Integer`: Numeric value
  - `YesNo`: Boolean value
  - `DateTime`: Date and time value
- `PossibleValueConfigurations (array)` - Optional. An array of configurations for OrderedList, MultiChoice, or SingleChoice property types. Each configuration has the following properties:
  - `Name (string)`: The name of the value (required)
  - `Description (string)`: A description of the value (optional)
- `Parameters (array)` - Optional. An array of strings in `"name=value"` format for additional configuration.

Examples:

1. Create a property list for PII data.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationPropertyDefinition -Name "ContainsPII" -Type OrderedList -
PossibleValueConfigurations @(
            @{ Name = "Yes" },
            @{ Name = "No" })
}
```

2. Create an ordered list property for Data Sensitivity.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationPropertyDefinition -Name "DataSensitivity" -Type
 OrderedList -PossibleValueConfigurations @(
            @{ Name = "Public" },
            @{ Name = "Internal" },
            @{ Name = "Confidential" },
            @{ Name = "Restricted" }
        )
}
```

Get-FSxFSRMClassificationPropertyDefinition

`Get-FSxFSRMClassificationPropertyDefinition`: Retrieves one or more classification property definitions from your file system.

Parameters:

- `Name (array)` - Optional. An array of names of property definitions to retrieve. If you don't specify a name, the command returns all property definitions on the file system.

Examples:

1. Retrieve all classification property definitions on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMClassificationPropertyDefinition
}
```

Set-FSxFSRMClassificationPropertyDefinition

Modifies the properties of an existing classification property definition.

Parameters

- `Name (array)` - Required. An array of property names to modify.
- `DisplayName (string)` - Optional. A new display name for the property definition.
- `Description (string)` - Optional. A new description for the property definition.
- `PossibleValueConfigurations (array)` - Optional. A new array of configurations for OrderedList, MultiChoice, or SingleChoice properties. Each configuration has the following properties:
  - `Name (string)`: The name of the value (required)
  - `Description (string)`: A description of the value (optional)
- `Parameters (array)` - Optional. A new array of strings in "name=value" format.
- `PassThru (boolean)` - Optional. If set to true, returns the modified property definition object.

Examples:

1. Update possible values with descriptions on an existing Property Definition.

```
$values = [System.Collections.ArrayList]@()
$null = $values.Add(@{
    Name = "High"
    Description = "High Risk Content"
})
$null = $values.Add(@{
    Name = "Medium"
    Description = "Medium Risk Content"
})

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $values -ScriptBlock {
    param($values)
    Set-FSxFSRMClassificationPropertyDefinition -Name "RiskLevel" -
PossibleValueConfigurations $Using:values -PassThru
}
```

Remove-FSxFSRMClassificationPropertyDefinition

Removes one or more classification property definitions from your file system. Only locally defined property definitions can be removed.

Parameters

- `Name (array)` - Required. An array of property names to remove.
- `PassThru (boolean)` - Optional. If set to true, returns the removed property definition object.

Examples:

1. Remove a single property definition.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMClassificationPropertyDefinition -Name "RiskLevel" -PassThru
}
```

Classification Rule Commands

New-FSxFSRMClassificationRule

Creates an automatic classification rule that assigns property values to files based on specified criteria. Each rule sets a value for a single property.

Parameters

- `Name (string)` - Required. A name for the classification rule.
- `Description (string)` - Optional. A description for the classification rule.
- `Property (string)` - Required. The name of the classification property to set. Must be an existing property definition name.
- `PropertyValue (string)` - Optional. The value to assign to the property. Must be valid for the specified classification mechanism.
- `Namespace (array)` - Required. An array of paths or folder types where the rule applies.

- `Disabled (boolean)` - Optional. If set to true, creates the rule in a disabled state.
- `ReevaluateProperty (string)` - Optional. Specifies when to re-evaluate files. You can specify the following values:
  - `Never`: Only evaluate files without existing property value
  - `Overwrite`: Re-evaluate when files change and overwrite existing value
  - `Aggregate`: Re-evaluate when files change and combine with existing value
- `Flags (array)` - Optional. Specifies special behaviors for the rule. You can specify the following values:
  - `ClearAutomaticallyClassifiedProperty`
  - `ClearManuallyClassifiedProperty`
  - `Deprecated`
- `ContentRegularExpression (array)` - Optional. An array of regular expressions to match file content.
- `ContentString (array)` - Optional. An array of case-insensitive strings to search for in file content.
- `ClassificationMechanism (string)` - Required. The mechanism to use for classifying files. You can specify the following values:
  - `Content Classifier`: Scans file content for specific strings or regular expression patterns. When you specify Content Classifier, you can use the ContentString, ContentStringCaseSensitive, or ContentRegularExpression parameters to define the content to search for.
  - `Folder Classifier`: Classifies files based on their folder location
- `Parameters (array)` - Optional. An array of `"name=value"` strings for additional configuration.

Examples:

1. Detect Social Security Numbers using a Regular Expression.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Detect_SSN" -Property "ContainsPII" -
PropertyValue "Yes" -Namespace "share" -ClassificationMechanism "Content Classifier" -
ContentRegularExpression "\b\d{3}-\d{2}-\d{4}\b"
}
```

2. Detect Credit Card Numbers using a regular Expression.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Detect_CreditCard" -Property "ContainsPII" -
PropertyValue "Yes" -Namespace "share" -ClassificationMechanism "Content Classifier" -
ContentRegularExpression "\b\d{4}[\s-]?\d{4}[\s-]?\d{4}[\s-]?\d{4}\b"
}
```

3. Classify every file under a folder with a 7-year retention period Property.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Contracts_Records_7Year" -Property
 "RetentionPeriod" -PropertyValue "7 years" -Namespace "share/Legal Documents" -
ClassificationMechanism "Folder Classifier"
}
```

Get-FSxFSRMClassificationRule

Retrieves one or more classification rules from your file system.

Parameters

- `Name (array)` - Optional. An array of names of classification rules to retrieve. If you don't specify a name, the command returns all rules on the file system.

Examples:

1. Retrieve all classification rules on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMClassificationRule
}
```

Set-FSxFSRMClassificationRule

Modifies the properties of existing classification rules.

Parameters

- `Name (array)` - Required. An array of names of classification rules to modify.
- `Description (string)` - Optional. A new description for the rule.
- `Property (string)` - Optional. The name of the classification property to set.
- `PropertyValue (string)` - Optional. A new value to assign to the property.
- `Namespace (array)` - Optional. A new array of paths or folder types where the rule applies.
- `Disabled (boolean)` - Optional. If set to true, disables the rule. If set to false, enables the rule.
- `ReevaluateProperty (string)` - Optional. Changes when to re-evaluate files. You can specify the following values:
  - `Never`: Only evaluate files without existing property value
  - `Overwrite`: Re-evaluate when files change and overwrite existing value
  - `Aggregate`: Re-evaluate when files change and combine with existing value
- `Flags (array)` - Optional. New special behaviors for the rule. You can specify the following values:
  - `ClearAutomaticallyClassifiedProperty`
  - `ClearManuallyClassifiedProperty`
  - `Deprecated`
- `ContentRegularExpression (array)` - Optional. A new array of regular expressions.
- `ContentString (array)` - Optional. A new array of case-insensitive search strings.
- `ContentStringCaseSensitive (array)` - Optional. A new array of case-sensitive search strings.
- `ClassificationMechanism (string)` - Optional. A new classification mechanism to use. You can specify the following values:
  - `Content Classifier`: Scans file content for specific strings or regular expression patterns. When you specify Content Classifier, you can use the ContentString, ContentStringCaseSensitive, or ContentRegularExpression parameters to define the content to search for.
  - `Folder Classifier`: Classifies files based on their folder location
- `Parameters (array)` - Optional. A new array of `"name=value"` configuration strings.
- `PassThru (boolean)` - Optional. If set to true, returns the modified rule object.

Examples:

1. Update rule properties and namespace of an existing Classification Rule.

```
$namespaces = @("share\finance", "share\accounting")

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $namespaces -ScriptBlock {
    param($namespaces)
    Set-FSxFSRMClassificationRule -Name "Detect_CreditCard" -Description "Updated PII
 detection" -Namespace $Using:namespaces -ReevaluateProperty "Overwrite"
}
```

Remove-FSxFSRMClassificationRule

Removes one or more classification rules from your file system.

Parameters

- `Name (array)` - Required. An array of names of classification rules to remove.
- `PassThru (boolean)` - Optional. If set to true, returns the removed rule object.

Examples:

1. Remove a single classification rule.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMClassificationRule -Name "Find Confidential Files" -PassThru
}
```

Management Property Commands

Get-FSxFSRMMgmtProperty

Retrieves management properties from specified folders. Management properties are classification properties that apply to folders rather than files.

Parameters

- `Namespace (string)` - Optional. A path to a folder.

- `Name (string)` - Optional. The name of a management property to retrieve. If you don't specify a name, the command retrieves all management properties.
- `Recurse (boolean)` - Optional. If set to true, retrieves management properties for all folders within the namespace. Requires the Namespace parameter.
- `Effective (boolean)` - Optional. If set to true, retrieves the management property for the nearest folder with the specified name. The search includes the specified namespace and its parent hierarchy. Requires the Name parameter.

Examples:

1. Retrieve all management properties on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMMgmtProperty
}
```

2. Retrieve management properties for a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMMgmtProperty -Namespace "share\department"
}
```

Remove-FSxFSRMMgmtProperty

Removes management properties from specified folders.

Parameters

- `Namespace (string)` - Optional. A path to a folder.
- `Name (string)` - Required. The name of the management property to remove.
- `Recurse (boolean)` - Optional. If set to true, removes management properties for all folders within the namespace. Requires the Namespace parameter.

Examples:

1. Remove all instances of a management property.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMMgmtProperty -Name "FolderUsage_MS"
}
```

2. Remove a management property from a specific folder.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMMgmtProperty -Name "FolderUsage_MS" -Namespace "share\department"
}
```

Set-FSxFSRMMgmtProperty

Changes the value of a management property for a specified namespace. Management properties are classification properties that apply to folders and don't have the Secure flag set.

Parameters

- `Namespace (string)` - Optional. The folder path.
- `Name (string)` - Required. The name of the management property to modify. Must be an existing classification property that applies to folders.
- `Value (string)` - Required. The new value to assign to the management property.

Examples:

1. Set a folder usage property.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMMgmtProperty -Namespace "share\department" -Name "FolderUsage_MS" -Value
 "User Files"
}
```

Classification Process Commands

Get-FSxFSRMClassification

Retrieves the status of the running file classification process. The status can be one of the following values:

- `Unknown`: The classification status cannot be determined
- `NotRunning`: No classification is currently running
- `Queued`: Classification is queued to start
- `Running`: Classification is currently in progress

Parameters

None

Examples:

1. Retrieve the current classification status.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMClassification
}
```

Start-FSxFSRMClassification

Initiates the file classification process, which applies classification rules to files and generates a classification report.

Parameters

- `Queue (boolean)` - Optional. If set to true, adds the classification task to a queue to run within the next 5 minutes. Any tasks queued during this period will run together. If set to false or not specified, classification starts immediately.
- `RunDuration (number)` - Optional. Specifies how many hours the classification process should run before being canceled. Valid values: `-1` to `2147483`. Special values:
  - `-1`: Run until canceled
  - `0`: Run to completion
  - If not specified, runs to completion.

Examples:

1. Start classification with no time limit.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMClassification -RunDuration 0
}
```

Stop-FSxFSRMClassification

Stops any running or queued classification job on your file system.

Parameters

None

Examples:

1. Stop a running classification.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Stop-FSxFSRMClassification
}
```

Wait-FSxFSRMClassification

Waits for the file classification process to complete. Use this command when you need to perform actions that depend on classification finishing, such as generating reports based on classified files.

Parameters

- `Timeout (number)` - Optional. Specifies how long to wait, in seconds, for the classification to complete. If the timeout expires before classification finishes, the command returns but classification continues running in the background. Valid values: `-1` to `2147483`. Special values:
  - `-1`: Wait indefinitely until classification completes (default)
  - `0`: Check the current status and return immediately without waiting

Examples:

1. Wait indefinitely for classification to complete.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Wait-FSxFSRMClassification
}
```

Set-FSxFSRMClassification

Modifies the configuration settings for file classification.

Parameters

- `ExcludeNamespace (array)` - Optional. An array of additional folders to exclude from classification.
- `ScheduleConfigurations (hashtable)` - Optional. A hashtable containing schedule configuration with the following properties:
  - `Time (datetime)`: DateTime object specifying when to run the task (required)
  - `RunDuration (number)`: Number of hours to run the task (optional)
  - `Weekly (array)`: Array of weekdays (optional)
  - `Monthly (array)`: Array of days of month, use -1 for last day (optional)
- `Continuous (boolean)` - Optional. If set to true, enables continuous background classification.
- `PassThru (boolean)` - Optional. If set to true, returns the modified classification configuration object.

Examples:

1. Enable continuous classification.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMClassification -Continuous $true
}
```

2. Set a weekly schedule to run classification.

```
$schedule = @{
    Time = ("12:00am")
    Weekly = @('Monday', 'Wednesday', 'Friday')
}
```

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $schedule -ScriptBlock {
    param($schedule)
    Set-FSxFSRMClassification -ScheduleConfigurations $schedule
}
```

3. Set a monthly schedule with custom exclusions.

```
$schedule = @{
    Time = ("12:00am")
    Monthly = @(1, 15, -1) # 1st, 15th, and last day
    RunDuration = 4
}
$excludeNamespaces = @("share\folder /s")

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList @($schedule, $excludeNamespaces) -ScriptBlock {
    param($schedule, $excludeNamespaces)
    Set-FSxFSRMClassification -ScheduleConfigurations $schedule -ExcludeNamespace
 $excludeNamespaces
}
```

Storage Reports

Storage reports provide detailed analysis of file system usage, helping you understand how storage is being consumed, identify files that can be archived or deleted, and monitor compliance with file management policies. You can generate multiple types of reports that analyze file ownership, file types, duplicate files, large files, file screening and quota usage.

Report types

You can create the following report types:

- `DuplicateFiles`

  Identifies files that have identical content based on file size and hash comparison. Use this report to find redundant files that consume unnecessary storage space. The report groups duplicate files together and shows the total space that could be recovered by removing duplicates.

- `FilesByFileGroup`

Groups files by their file group membership and shows storage consumption for each file group. Use this report to understand which types of files (documents, media, executables, etc.) consume the most storage space.

- `FilesByOwner`

  Groups files by owner and shows how much storage each user or group consumes. Use this report to identify users who are consuming the most storage space and to allocate storage costs or quotas appropriately.
- `FilesByProperty`

  Groups files by classification property values and shows file count and storage consumption for each property value. Use this report to analyze files based on their classification, such as data sensitivity level, department, or retention period. This report requires that files have been classified using Classification rules.
- `FileScreenAuditFiles`

  Lists File Screening violations where users attempted to save files that were blocked by active file screens. Use this report to monitor compliance with file screening policies and identify users who frequently attempt to save unauthorized file types.
- `FoldersByProperty`

  Groups folders by management property values and shows storage consumption for each property value. Use this report to analyze storage usage by folder purpose, such as user files, group shares, or application files.
- `LargeFiles`

  Lists files that exceed a specified size threshold. Use this report to identify files that consume significant storage space and may be candidates for archiving, compression, or deletion.
- `LeastRecentlyAccessed`

  Lists files that haven't been accessed for a specified number of days. Use this report to identify inactive files that can be archived or moved to lower-cost storage tiers.
- `MostRecentlyAccessed`

  Lists files that were accessed within a specified number of days.
- `QuotaUsage`

Shows quota usage statistics for folders with quotas configured. Use this report to monitor quota compliance and identify folders approaching their quota limits.

Report formats

You can generate reports in multiple formats to suit different use cases:

- `DHTML` - Dynamic HTML format with interactive features like sorting and filtering.
- `HTML` - Static HTML format suitable for archiving or emailing.
- `XML` - Structured data format for programmatic processing.
- `CSV` - Comma-separated values format for importing into spreadsheet applications.
- `Text` - Plain text format for simple viewing or processing.

You can specify multiple formats for a single report.

Interactive and scheduled reports

You can create two types of storage reports:

1. **Interactive reports** - Run immediately when created and execute only once. Use interactive reports for ad-hoc analysis or troubleshooting. Interactive reports do not have schedules and cannot be modified after creation. To run another interactive report, you must create a new report with a different name.
2. **Scheduled reports** - Run automatically according to a configured schedule. Use scheduled reports for regular monitoring and compliance reporting. You can schedule reports to run weekly or monthly at specific times. Scheduled reports can be modified to change their configuration, and you can also run them on-demand using the Start-FSxFSRMStorageReport command without waiting for the scheduled time.

Running reports

After you create a scheduled report, you can run it in several ways:

- **Automatic execution** - Scheduled reports run automatically at their configured schedule time.
- **Manual execution** - Use Start-FSxFSRMStorageReport to run a scheduled report on-demand without waiting for the scheduled time.

You can monitor report execution using Get-FSxFSRMStorageReport to check the status.

Accessing storage reports

After FSRM generates storage reports, the report files are saved to a default location on your file system. To access these reports, you need to map the administrative D$ share of your file system.

To access storage reports

1. Map the administrative D$ share using the following path format:

   ```
   \\file-system-dns-name\D$
   ```

   For example:

   ```
   \\amznfsxaa11bb22.corp.example.com\D$
   ```

2. Navigate to the StorageReports folder. This folder contains subfolders organized by report type and execution date.

> **Note**
>
> Accessing the administrative D$ share requires administrator credentials.

Storage report best practices

Follow these best practices to ensure efficient and effective storage reporting:

Performance considerations

Storage report generation is resource-intensive because FSRM must scan large amounts of files.

- **Limit report scope** - Use the `Namespace` parameter to limit reports to specific folders rather than scanning the entire file system. Scanning large directory structures is resource-intensive and can take hours to complete.
- **Schedule reports during off-peak hours** - Run scheduled reports during periods of low system activity, to minimize impact on performance. Avoid running reports during backup windows or other maintenance tasks.

- **Set reasonable thresholds** - Use threshold parameters to limit report output to actionable data. For example, set `LargeFileMinimum` to a value that identifies files worth investigating, not every file over 1MB.
- **Use RunDuration limits** - Set the `RunDuration` parameter to prevent reports from running too long and impacting system performance. If a report doesn't complete within the time limit, it will resume during the next scheduled run.
- **Monitor report performance** - Use Get-FSxFSRMStorageReport to check how long reports take to complete. If reports consistently take too long, consider narrowing their scope or running them less frequently.

Report design

- **Use descriptive names** - Give reports clear, descriptive names that indicate what they analyze and when they run, such as "Weekly Large Files - Finance Share" or "Monthly Duplicate Files - All Shares".
- **Combine related analysis** - When generating multiple report types for the same namespace, create a single report with multiple `ReportType` values rather than separate reports. This is more efficient because FSRM only needs to scan the directory structure once.
- **Filter by file patterns** - Use file pattern parameters to focus reports on specific file types. For example, when analyzing large files, you might create separate reports for video files, database files, and archive files to better understand storage consumption patterns.
- **Leverage classification properties** - Use `FilesByProperty` reports to analyze files based on their classification. This provides more meaningful insights.

Report management

- **Review reports regularly** - Schedule time to review report results and take action on findings. Reports are only valuable if you use them to make storage management decisions.
- **Archive old reports** - Report files accumulate over time and consume storage space. Establish a retention policy for report files and delete or archive old reports that are no longer needed.
- **Test reports before scheduling** - Create interactive reports to test report configurations and verify they produce the expected results before creating scheduled versions.

Storage report management commands

You can access two families of FSx remote PowerShell commands for managing storage reports:

1. **Report definition commands** - Create, retrieve, modify, and remove storage report configurations that specify what data to analyze, when to run reports, and what formats to generate.
2. **Report execution commands** - Start, stop, monitor, and wait for storage report generation. Use these commands to run reports on-demand or manage long-running report jobs.

List of Storage Report FSx remote PowerShell commands

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

Report Definition Commands

New-FSxFSRMStorageReport

New-FSxFSRMStorageReport: Creates a storage report that analyzes specified directories to generate one or more report types.

Parameters:

- `Name (string)` - Required. A name for the storage report.
- `Namespace (array)` - Required. An array of paths or folder types to analyze. You can specify paths in multiple formats:
  - Folder Path
  - Folder classification. For example, [FolderUsage_MS="User Files"]
- `ReportType (array)` - Required. An array of report types to generate. You can specify the following values:
  - `DuplicateFiles`: Identifies duplicate files based on file size and content

  - `FilesByFileGroup`: Groups files by file group membership
  - `FilesByOwner`: Groups files by owner
  - `FilesByProperty`: Groups files by classification property
  - `FileScreenAuditFiles`: Lists file screening violations
  - `FoldersByProperty`: Groups folders by management property
  - `LargeFiles`: Lists files above a specified size threshold
  - `LeastRecentlyAccessed`: Lists files that haven't been accessed recently
  - `MostRecentlyAccessed`: Lists files that were accessed recently
  - `QuotaUsage`: Shows quota usage statistics
- `ReportFormat (array)` - Optional. An array of output formats. You can specify the following values:
  - `DHTML`: Dynamic HTML format
  - `HTML`: Static HTML format
  - `XML`: XML format
  - `CSV`: Comma-separated values format
  - `Text`: Plain text format
- `Interactive (boolean)` - Optional. If set to true, generates an interactive report. Interactive reports cannot be modified after creation.
- `ScheduleConfigurations (hashtable)` - Required unless report is Interactive. A hashtable containing schedule configuration with the following properties:
  - `Time (datetime)`: DateTime object specifying when to run the task (required)
  - `RunDuration (number)`: Number of hours to run the task (optional)
  - `Weekly (array)`: Array of weekdays (optional)
  - `Monthly (array)`: Array of days of month, use `-1` for last day (optional)

Report-specific parameters:

- `FileScreenAuditDaysSince (number)` - Optional. For FileScreenAuditFiles reports, specifies how many days back to include audit events.
- `FileScreenAuditUser (array)` - Optional. For FileScreenAuditFiles reports, specifies an array of user accounts to include in the report. Only file screening violations by these users will be included.

- `FileGroupIncluded (array)` - Optional. For FilesByFileGroup reports, specifies which file groups to include.
- `FileOwnerFilePattern (string)` - Optional. For FilesByOwner reports, specifies a file pattern to filter results.
- `PropertyName (string)` - Optional. For FilesByProperty reports, specifies the classification property to group by.
- `FolderPropertyName (string)` - Optional. For FoldersByProperty reports, specifies the folder property to group by.
- `PropertyFilePattern (string)` - Optional. For FilesByProperty and FoldersByProperty, specifies a file pattern to filter results.
- `LargeFileMinimum (number)` - Optional. For LargeFiles reports, specifies the minimum file size in bytes.
- `LargeFilePattern (string)` - Optional. For LargeFiles reports, specifies a file pattern to filter results.
- `LeastAccessedMinimum (number)` - Optional. For LeastRecentlyAccessed reports, specifies the minimum number of days since last access.
- `LeastAccessedFilePattern (string)` - Optional. For LeastRecentlyAccessed reports, specifies a file pattern to filter results.
- `MostAccessedMaximum (number)` - Optional. For MostRecentlyAccessed reports, specifies the maximum number of days since last access.
- `MostAccessedFilePattern (string)` - Optional. For MostRecentlyAccessed reports, specifies a file pattern to filter results.
- `QuotaMinimumUsage (number)` - Optional. For QuotaUsage reports, specifies the minimum quota usage percentage to include.

Examples:

1. Create a monthly large files report.

```
$schedule = @{
    Time = ("3:00 AM")
    Monthly = @(1) # Run on first day
}

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $schedule -ScriptBlock {
    param($schedule)
    New-FSxFSRMStorageReport -Name "Monthly Large Files" -Namespace "share
\data" -ReportType "LargeFiles" -LargeFileMinimum 100MB -ReportFormat "HTML" -
ScheduleConfigurations $schedule
}
```

2. Create a weekly duplicate files report with multiple namespaces and formats.

```
$schedule = @{
    Time = ("12:00 AM")
    Weekly = @('Sunday')
    RunDuration = 4
}

$namespaces = @("share\docs", "[FolderUsage_MS=User Files]")
$reportFormats = @("HTML", "CSV")
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList @($schedule, $namespaces, $reportFormats) -ScriptBlock {
    param($schedule, $namespaces, $reportFormats)
    New-FSxFSRMStorageReport -Name "Weekly Duplicates" -Namespace $namespaces -
ReportType "DuplicateFiles" -ReportFormat $reportFormats -ScheduleConfigurations
 $schedule
}
```

3. Create an interactive report that runs immediately.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMStorageReport -Name "Find large files" -Namespace "share" -Interactive
 $true -ReportType "QuotaUsage"
}
```

Get-FSxFSRMStorageReport

Get-FSxFSRMStorageReport: Retrieves one or more storage reports from your file system. Returns details about report configurations and status.

Parameters:

- `Name (array)` - Optional. An array of report names to retrieve. If you don't specify a name, the command returns all storage reports on the file system.

Examples:

1. Retrieve all storage reports on the file system.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMStorageReport
}
```

Remove-FSxFSRMStorageReport

Remove-FSxFSRMStorageReport: Removes one or more storage reports from your file system. You cannot remove reports that are currently running.

Parameters:

- `Name (array)` - Required. An array of report names to remove.
- `PassThru (boolean)` - Optional. If set to true, returns the removed report object.

Examples:

1. Remove a single storage report.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Remove-FSxFSRMStorageReport -Name "Monthly Report" -PassThru
}
```

Set-FSxFSRMStorageReport

Parameters

- `Name (array)` - Required. An array of report names to modify.
- `Namespace (array)` - Optional. An array of paths or folder types to analyze. You can specify paths in multiple formats:
  - Folder Path
  - Folder classification. For example, [FolderUsage_MS="User Files"]
- `ReportType (array)` - Optional. An array of report types to generate. You can specify the following values:

  - `DuplicateFiles`: Identifies duplicate files based on file size and content
  - `FilesByFileGroup`: Groups files by file group membership
  - `FilesByOwner`: Groups files by owner
  - `FilesByProperty`: Groups files by classification property
  - `FileScreenAuditFiles`: Lists file screening violations
  - `FoldersByProperty`: Groups folders by management property
  - `LargeFiles`: Lists files above a specified size threshold
  - `LeastRecentlyAccessed`: Lists files that haven't been accessed recently
  - `MostRecentlyAccessed`: Lists files that were accessed recently
  - `QuotaUsage`: Shows quota usage statistics
- `ReportFormat (array)` - Optional. An array of output formats. You can specify the following values:
  - `DHTML`: Dynamic HTML format
  - `HTML`: Static HTML format
  - `XML`: XML format
  - `CSV`: Comma-separated values format
  - `Text`: Plain text format
- `ScheduleConfigurations (hashtable)` - Required unless report is Interactive. A hashtable containing schedule configuration with the following properties:
  - `Time (datetime)`: DateTime object specifying when to run the task (required)
  - `RunDuration (number)`: Number of hours to run the task (optional)
  - `Weekly (array)`: Array of weekdays (optional)
  - `Monthly (array)`: Array of days of month, use `-1` for last day (optional)
- `PassThru (boolean)` - Optional. If set to true, returns the modified report object.

Report-Specific Parameters

- `FileScreenAuditDaysSince (number)` - Optional. For FileScreenAuditFiles reports, specifies how many days back to include audit events.
- `FileScreenAuditUser (array)` - Optional. For FileScreenAuditFiles reports, specifies an array of user accounts to include in the report. Only file screening violations by these users will be included.

- `FileGroupIncluded (array)` - Optional. For FilesByFileGroup reports, specifies which file groups to include.
- `FileOwnerFilePattern (string)` - Optional. For FilesByOwner reports, specifies a file pattern to filter results.
- `PropertyName (string)` - Optional. For FilesByProperty reports, specifies the classification property to group by.
- `FolderPropertyName (string)` - Optional. For FoldersByProperty reports, specifies the folder property to group by.
- `PropertyFilePattern (string)` - Optional. For FilesByProperty and FoldersByProperty, specifies a file pattern to filter results.
- `LargeFileMinimum (number)` - Optional. For LargeFiles reports, specifies the minimum file size in bytes.
- `LargeFilePattern (string)` - Optional. For LargeFiles reports, specifies a file pattern to filter results.
- `LeastAccessedMinimum (number)` - Optional. For LeastRecentlyAccessed reports, specifies the minimum number of days since last access.
- `LeastAccessedFilePattern (string)` - Optional. For LeastRecentlyAccessed reports, specifies a file pattern to filter results.
- `MostAccessedMaximum (number)` - Optional. For MostRecentlyAccessed reports, specifies the maximum number of days since last access.
- `MostAccessedFilePattern (string)` - Optional. For MostRecentlyAccessed reports, specifies a file pattern to filter results.
- `QuotaMinimumUsage (number)` - Optional. For QuotaUsage reports, specifies the minimum quota usage percentage to include.

Examples:

1. Update schedule and format for an existing report.

```
$schedule = @{
    Time = ("3:00 AM")
    Monthly = @(1)
}
$reportFormats = @("HTML", "CSV")
```

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList @($schedule, $reportFormats) -ScriptBlock {
    param($schedule, $reportFormats)
    Set-FSxFSRMStorageReport -Name "Monthly Report" -ScheduleConfigurations $schedule -
ReportFormat $reportFormats -PassThru
}
```

Report Execution Commands

Start-FSxFSRMStorageReport

Parameters

- `Name (array)` - Required. An array of report names to start.
- `Queue (boolean)` - Optional. If set to true, adds the report to a queue to run within the next 5 minutes. Any reports queued during this period will run together. If set to false or not specified, the report starts immediately.
- `RunDuration (number)` - Optional. Specifies how many hours the report should run before being canceled. Valid values: `-1` to `2147483`. Special values:
  - `0`: Run to completion
  - `-1`: Run until canceled

  If not specified, runs to completion.

Examples

1. Start a storage report immediately.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMStorageReport -Name "Monthly Report"
}
```

2. Queue a storage report with a duration limit.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMStorageReport -Name "Quarterly Report" -Queue: $true -RunDuration 4
}
```

Stop-FSxFSRMStorageReport

Parameters

- `Name (array)` - Required. An array of report names to stop.

Examples:

1. Stop a single storage report.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Stop-FSxFSRMStorageReport -Name "Monthly Report"
}
```

Wait-FSxFSRMStorageReport

Parameters

- `Name (array)` - Required. An array of report names to wait for.
- `Timeout (number)` - Optional. Specifies how long to wait, in seconds, for the reports to complete. If the timeout expires before reports finish, the command returns but report generation continues running in the background. Valid values: `-1` to `2147483`. Special values:
  - `-1`: Wait indefinitely until reports complete (default)
  - `0`: Check the current status and return immediately without waiting

Examples:

1. Wait indefinitely for a storage report to complete.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Wait-FSxFSRMStorageReport -Name "Monthly Report"
}
```

File Management Tasks

Amazon FSx for Windows File Server does not support FSRM file management tasks. However, you can achieve common use cases such as data archiving and retention policies by using native PowerShell commands from client machines that have network access to your file system.

For example, you can use PowerShell scripts on client machines to:

- Move or archive files based on age or last access time
- Delete classified files that exceed a retention period
- Copy files to archive storage based on classification properties

You can access files properties using the `Get-FsrmClassification` command and take actions based on the values.

To access FSRM classification properties or other FSRM metadata from client-side PowerShell scripts, the client machine must also have FSRM installed.

FSRM Settings

FSRM settings provide system-wide configuration that allows you to customize behavior and streamline feature management. Use these settings to control how FSRM operates across your file system and to set default values that simplify creating and configuring features like storage reports and file screening.

Settings categories

FSRM settings are organized into three categories:

File screen auditing

File screen auditing records when users attempt to save files that are blocked by active file screens. This information is essential for monitoring compliance with file screening policies and identifying users who frequently attempt to save unauthorized file types.

- `ReportFileScreenAuditEnable` - This setting controls whether FSRM logs file screening violations at all. If disabled, FSRM does not record file screen violations, and `FileScreenAuditFiles` reports will have no data to display. You must enable this setting to use file screen audit reports.

- `ReportFileScreenAuditDaysSince` - This setting provides the default time range for file screen audit reports. When you create a `FileScreenAuditFiles` report without specifying how far back to look, FSRM uses this value. Setting an appropriate default (such as 30 days) ensures that reports focus on recent violations without including excessive historical data.
- `ReportFileScreenAuditUser` - This setting provides the default list of users to include in file screen audit reports. When you create a `FileScreenAuditFiles` report without specifying which users to include, FSRM uses this list. If empty, reports include all users by default. You can use this setting to focus reports on specific user groups or departments.

Default report filters

Default report filter settings provide values that are used when you create storage reports without specifying certain parameters. These defaults simplify report creation and ensure consistency across similar reports.

Each report type has associated default settings:

- Large file reports - `ReportLargeFileMinimum` sets the default minimum file size, and `ReportLargeFilePattern` sets the default file pattern filter.
- Least accessed file reports - `ReportLeastAccessedMinimum` sets the default number of days since last access, and `ReportLeastAccessedFilePattern` sets the default file pattern filter.
- Most accessed file reports - `ReportMostAccessedMaximum` sets the default maximum number of days since last access, and `ReportMostAccessedFilePattern` sets the default file pattern filter.
- Files by owner reports - `ReportFileOwnerFilePattern` sets the default file pattern filter, and `ReportFileOwnerUser` sets the default list of users to include.
- Files by property reports - `ReportPropertyName` sets the default classification property to analyze, and `ReportPropertyFilePattern` sets the default file pattern filter.
- Files by file group reports - `ReportFileGroupIncluded` sets the default list of file groups to include.
- Quota usage reports - `ReportQuotaMinimumUsage` sets the default minimum quota usage percentage.

When you create a report, you can override any of these defaults by specifying the parameter explicitly in the report configuration. The global defaults only apply when you don't specify a value.

Report limits

Report limit settings control the maximum number of items to include in storage reports. These limits serve two purposes:

1. Performance management - Limiting the number of items in reports prevents reports from taking too long to generate or consuming excessive system resources. Large reports that analyze millions of files can take hours to complete and impact system performance.
2. Report usability - Reports with thousands of entries are difficult to review and analyze. Report limits ensure that reports remain focused on the most relevant data.

You can set up granular control over report limits:

- Overall limits - `ReportLimitMaxFile` limits the total number of files in any report, regardless of type.
- Per-report-type limits - Settings like `ReportLimitMaxFileGroup`, `ReportLimitMaxOwner`, and `ReportLimitMaxPropertyValue` limit the number of groups, owners, or property values to include in specific report types.
- Per-group limits - Settings like `ReportLimitMaxFilesPerFileGroup`, `ReportLimitMaxFilesPerOwner`, and `ReportLimitMaxFilesPerPropertyValue` limit how many files to show within each group in the report.

When a report reaches a limit, FSRM includes the items that consume the most storage space or are most relevant to the report type and indicates in the report that the limit was reached.

FSRM settings commands

You can access commands for retrieving and modifying global settings. Use these commands to configure system-wide FSRM behavior.

List of FSRM Settings FSx remote PowerShell commands

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

Get-FSxFSRMSetting

`Get-FSxFSRMSetting`: Retrieves the current File Server Resource Manager settings on your file system. Returns only the settings that can be modified using Set-FSxFSRMSetting.

Parameters:

None

Examples:

1. Retrieve all current FSRM settings.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
Get-FSxFSRMSetting
}
```

Set-FSxFSRMSetting

`Set-FSxFSRMSetting`: Modifies global File Server Resource Manager settings on your file system. These settings provide default values for storage reports and control FSRM behavior.

Parameters:

File screen audit settings:

- `ReportFileScreenAuditEnable (boolean)` - Optional. Controls whether file screening audit events are included in FSRM reports.

- `ReportFileScreenAuditDaysSince (number)` - Optional. The default number of days to look back for file screening violations when generating FileScreenAuditFiles reports.
- `ReportFileScreenAuditUser (array)` - Optional. An array of the default list of user accounts to include in FileScreenAuditFiles reports.

Default report filter settings:

- `ReportFileGroupIncluded (array)` - Optional. An array of file group names to include in reports by default.
- `ReportFileOwnerFilePattern (string)` - Optional. The default file pattern for files by owner reports. Supports wildcards (* and ?).
- `ReportFileOwnerUser (array)` - Optional. An array of users in Domain\User format for files by owner reports.
- `ReportLargeFileMinimum (number)` - Optional. The default minimum file size in bytes for large file reports.
- `ReportLargeFilePattern (string)` - Optional. The default file pattern for large file reports. Supports wildcards (* and ?).
- `ReportLeastAccessedFilePattern (string)` - Optional. The default file pattern for least accessed file reports. Supports wildcards (* and ?).
- `ReportLeastAccessedMinimum (number)` - Optional. The default minimum number of days since last access for least accessed file reports.
- `ReportMostAccessedFilePattern (string)` - Optional. The default file pattern for most accessed file reports. Supports wildcards (* and ?).
- `ReportMostAccessedMaximum (number)` - Optional. The default maximum number of days since last access for most accessed file reports.
- `ReportPropertyFilePattern (string)` - Optional. The default file pattern for property reports. Supports wildcards (* and ?).
- `ReportPropertyName (string)` - Optional. The default property name for property reports.
- `ReportQuotaMinimumUsage (number)` - Optional. The default minimum quota usage percentage for quota usage reports.

Report limit settings:

- `ReportLimitMaxDuplicateGroup (number)` - Optional. The maximum number of duplicate file groups to include in duplicate file reports.
- `ReportLimitMaxFile (number)` - Optional. The maximum number of files to include in storage reports.
- `ReportLimitMaxFileGroup (number)` - Optional. The maximum number of file groups to include in reports.
- `ReportLimitMaxFileScreenEvent (number)` - Optional. The maximum number of file screen events to include in file screen audit reports.
- `ReportLimitMaxFilesPerDuplicateGroup (number)` - Optional. The maximum number of files per duplicate group in duplicate file reports.
- `ReportLimitMaxFilesPerFileGroup (number)` - Optional. The maximum number of files per file group in files by file group reports.
- `ReportLimitMaxFilesPerOwner (number)` - Optional. The maximum number of files per owner in files by owner reports.
- `ReportLimitMaxFilesPerPropertyValue (number)` - Optional. The maximum number of files per property value in files by property reports.
- `ReportLimitMaxOwner (number)` - Optional. The maximum number of owners to include in files by owner reports.
- `ReportLimitMaxPropertyValue (number)` - Optional. The maximum number of property values to include in files by property reports.
- `ReportLimitMaxQuota (number)` - Optional. The maximum number of quotas to include in quota usage reports.

Other settings:

- `PassThru (boolean)` - Optional. If set to true, returns the modified settings object.

Examples:

1. Configure default file screen auditing with a 30-day history.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMSetting -ReportFileScreenAuditDaysSince 30 -PassThru
}
```

2. Configure default large file report settings.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMSetting -ReportLargeFileMinimum 100MB -ReportLargeFilePattern "*.iso"
 -PassThru
}
```

Event Logs

When you enable FSRM on your file system, AWS FSx for Windows File Server generates event logs for file management activities and sends them to the destination you configured (AWS CloudWatch Logs or AWS Kinesis Data Firehose). These logs help you monitor FSRM operations, troubleshoot issues, and maintain audit trails of file management activities.

What FSRM logs

When you enable FSRM on your file system, AWS FSx for Windows File Server logs events and sends them to your configured destination. The following events will be logged:

- File screening violations - When users attempt to save files that are monitored by file screens that have event notification actions
- Quota threshold notifications - When quota usage reaches configured thresholds that have event notification actions
- FSRM service events – Confirmation of notification settings, service errors, and operational failures

Accessing FSRM logs

The location where you access FSRM logs depends on the destination you configured when enabling FSRM:

CloudWatch Logs

> View logs in the CloudWatch Logs console by navigating to the log group you specified. You can search, filter, and analyze logs using CloudWatch Logs Insights, and set up CloudWatch alarms to notify you of specific events.

Kinesis Data Firehose

Logs are delivered to the destination configured in your Kinesis Data Firehose delivery stream, such as Amazon S3, AWS Redshift, or AWS OpenSearch Service. You can process and analyze logs using the tools and services integrated with your delivery stream.

Common Use Cases

This topic provides step-by-step examples for common File Server Resource Manager tasks. These examples demonstrate how to use and implement FSRM features to solve typical file management challenges.

> **Note**
>
> All the examples in this page assume that you have defined the `$FSxWindowsRemotePowerShellEndpoint` variable with your file system's Windows Remote PowerShell endpoint. You can find this endpoint in the Amazon FSx console on your file system's details page, or by using the AWS CLI `describe-file-systems` command.

Setting a hard quota on a folder

This example shows how to create a hard quota that prevents users from storing more than 10 GB in a 'department' folder.

To set a quota on a folder:

1. Create a hard quota with a 10 GB limit:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMQuota -Folder "share\department" -Size 10GB -Description "10 GB hard
 limit for department folder"
}
```

2. (Optional) Modify the quota to add a threshold notification at 85% usage:

```
$thresholds = [System.Collections.ArrayList]@()
```

```
$threshold = @{
    ThresholdPercentage = 85
    Action = @(
        @{
            ActionType = "Event"
            EventType = "Warning"
            MessageBody = "Department folder has reached 85% of quota limit"
        }
    )
}

$null = $thresholds.Add($threshold)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList ($thresholds) -ScriptBlock {
    param($thresholds)
    Set-FSxFSRMQuota -Folder "share\department" -ThresholdConfigurations
 $Using:thresholds
}
```

3. Verify the quota was created:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMQuota -Folder "share\department"
}
```

Restricting specific file types using file groups

This example shows how to block users from saving audio and video files to a business documents folder using the default "`Audio and Video Files`" file group.

To restrict file types using file groups:

1. Create an active file screen that blocks audio and video files:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
New-FSxFSRMFileScreen -Folder "share\business-documents" -IncludeGroup "Audio and
 Video Files" -Description "Block media files in business documents folder"
}
```

2. (Optional) Update the file screen to add a notification when users attempt to save blocked files:

```
$notifications = [System.Collections.ArrayList]@()

$eventNotification = @{
    ActionType = "Event"
    EventType = "Warning"
    MessageBody = "User attempted to save blocked media file"
}
$null = $notifications.Add($eventNotification)

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $notifications -ScriptBlock {
    param($notifications)
    Set-FSxFSRMFileScreen -Folder "share\business-documents" -
NotificationConfigurations $Using:notifications
}
```

3. Verify the file screen was created:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMFileScreen -Folder "share\business-documents"
}
```

Identify and classify PII data

This example shows how to automatically identify files containing Social Security numbers and classify them as containing personally identifiable information (PII).

To identify and classify PII data:

1. Create a classification property for PII:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationPropertyDefinition -Name "ContainsPII" -Type
 OrderedList -PossibleValueConfigurations @(
        @{ Name = "Yes" },
        @{ Name = "No" })
}
```

2. Create a classification rule to detect Social Security numbers:

> **Note**
>
> The following Regular Expression will search files for text with the pattern XXX-XX-XXXX. For production use, consider using more sophisticated patterns or combining multiple detection methods.

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Detect_SSN" -Property "ContainsPII"
 -PropertyValue "Yes" -Namespace "share" -ClassificationMechanism "Content
 Classifier" -ContentRegularExpression "\b\d{3}-\d{2}-\d{4}\b"
}
```

3. Run classification:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMClassification
}
```

4. (Optional) Configure continuous classification to automatically classify new files:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMClassification -Continuous $true
}
```

5. Check for status (1 means completed):

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMClassification
}
```

6. After classification completes, you can view the classification properties assigned to files by right-clicking a file in Windows File Explorer, selecting **Properties**, and choosing the **Classification** tab. This tab displays all classification properties and their values for the file.

Creating a retention policy for files

This example shows how to classify files by retention period based on their folder location, which you can then use with client-side PowerShell scripts to archive or delete files.

To create a retention policy for files:

1. Create a classification property for retention period:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationPropertyDefinition -Name "RetentionPeriod" -Type
 String -Description "File retention period"
}
```

2. Create classification rules for different retention periods:

   - 7-year retention for legal documents under the folder **Legal Documents**:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Legal_7Year" -Property
 "RetentionPeriod" -PropertyValue "7 years" -Namespace "share/Legal Documents" -
ClassificationMechanism "Folder Classifier"
}
```

   - 3-year retention for financial records under the folder **Finance**:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
    New-FSxFSRMClassificationRule -Name "Finance_3Year" -Property
 "RetentionPeriod" -PropertyValue "3 years" -Namespace "share/Finance" -
ClassificationMechanism "Folder Classifier"
}
```

You can also classify by file content and search for strings like "Retention Period Seven Years". To achieve this, use the `ClassificationMechanism "Content Classifier"` and `ContentString "Retention seven years"`.

3. Run classification to apply retention properties:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMClassification
}
```

4. (Optional) Configure continuous classification to automatically classify new files:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Set-FSxFSRMClassification -Continuous $true
}
```

5. Check for status (1 means completed):

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Get-FSxFSRMClassification
}
```

6. After classification completes, you can view the classification properties assigned to files by right-clicking a file in Windows File Explorer, selecting **Properties**, and choosing the **Classification** tab. This tab displays all classification properties and their values for the file.

7. Once files are classified with retention periods, you can use client-side PowerShell scripts to archive or delete files based on their `RetentionPeriod` property and age. For example, you can scan the file system and compare file's age with their retention period classification. For more information, see File Management Tasks.

Setting up common storage reports

This section shows how to create two commonly used storage reports: a large files report and a files by owner report.

Large files report

This example creates a monthly report that identifies files larger than 200 MB.

To create a large files report:

1. Create a scheduled large files report:

```
$schedule = @{
    Time = "2:00 AM"
    Monthly = @(1) # Run on the 1st of each month
}

Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $schedule -ScriptBlock {
    param($schedule)
    New-FSxFSRMStorageReport -Name "Monthly Large Files Report" -Namespace "share"
 -ReportType "LargeFiles" -LargeFileMinimum 200MB -ReportFormat "HTML","CSV" -
ScheduleConfigurations $schedule
}
```

2. (Optional) Run the report immediately to test:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMStorageReport -Name "Monthly Large Files Report"
}
```

Files by owner report

This example creates a weekly report that shows storage consumption by user.

To create a files by owner report:

1. Create a scheduled files by owner report:

```
$schedule = @{
    Time = "3:00 AM"
    Weekly = @('Sunday')
}
```

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ArgumentList $schedule -ScriptBlock {
    param($schedule)
    New-FSxFSRMStorageReport -Name "Weekly Files by Owner Report" -
Namespace "share" -ReportType "FilesByOwner" -ReportFormat "HTML","CSV" -
ScheduleConfigurations $schedule
}
```

2. (Optional) Run the report immediately to test:

```
Invoke-Command -ComputerName $FSxWindowsRemotePowerShellEndpoint -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
    Start-FSxFSRMStorageReport -Name "Weekly Files by Owner Report"
}
```

Access the generated reports by mapping the administrative D$ share. For more information, visit Accessing storage reports.

Managing storage on FSx for Windows File Server

Your file system's storage configuration includes the amount of provisioned storage capacity, the storage type, and if the storage type is solid state drive (SSD), the amount of SSD IOPS. You can configure these resources, along with the file system's throughput capacity, when creating a file system and after it's created, to achieve the desired performance for your workload. Learn how to manage your file system's storage and storage-related performance using the AWS Management Console, AWS CLI, and the Amazon FSx CLI for remote management on PowerShell by exploring the following topics.

Topics

- Optimizing storage costs
- Managing storage capacity
- Managing your file system's storage type
- Managing SSD IOPS
- Reducing storage costs with Data Deduplication
- Managing storage quotas
- Increasing file system storage capacity
- Monitoring storage capacity increases

- Increasing the storage capacity of an FSx for Windows File Server file system dynamically
- Updating the storage type of a FSx for Windows file system
- Monitoring storage type updates
- Updating a file system's SSD IOPS
- Monitoring provisioned SSD IOPS updates
- Managing data deduplication
- Troubleshooting data deduplication

Optimizing storage costs

You can optimize your storage costs using the storage configuration options available in FSx for Windows.

Storage type options—FSx for Windows File Server provides two storage types, 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 about storage types and file system performance, see FSx for Windows File Server performance.

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 Reducing storage costs with Data Deduplication.

Managing storage capacity

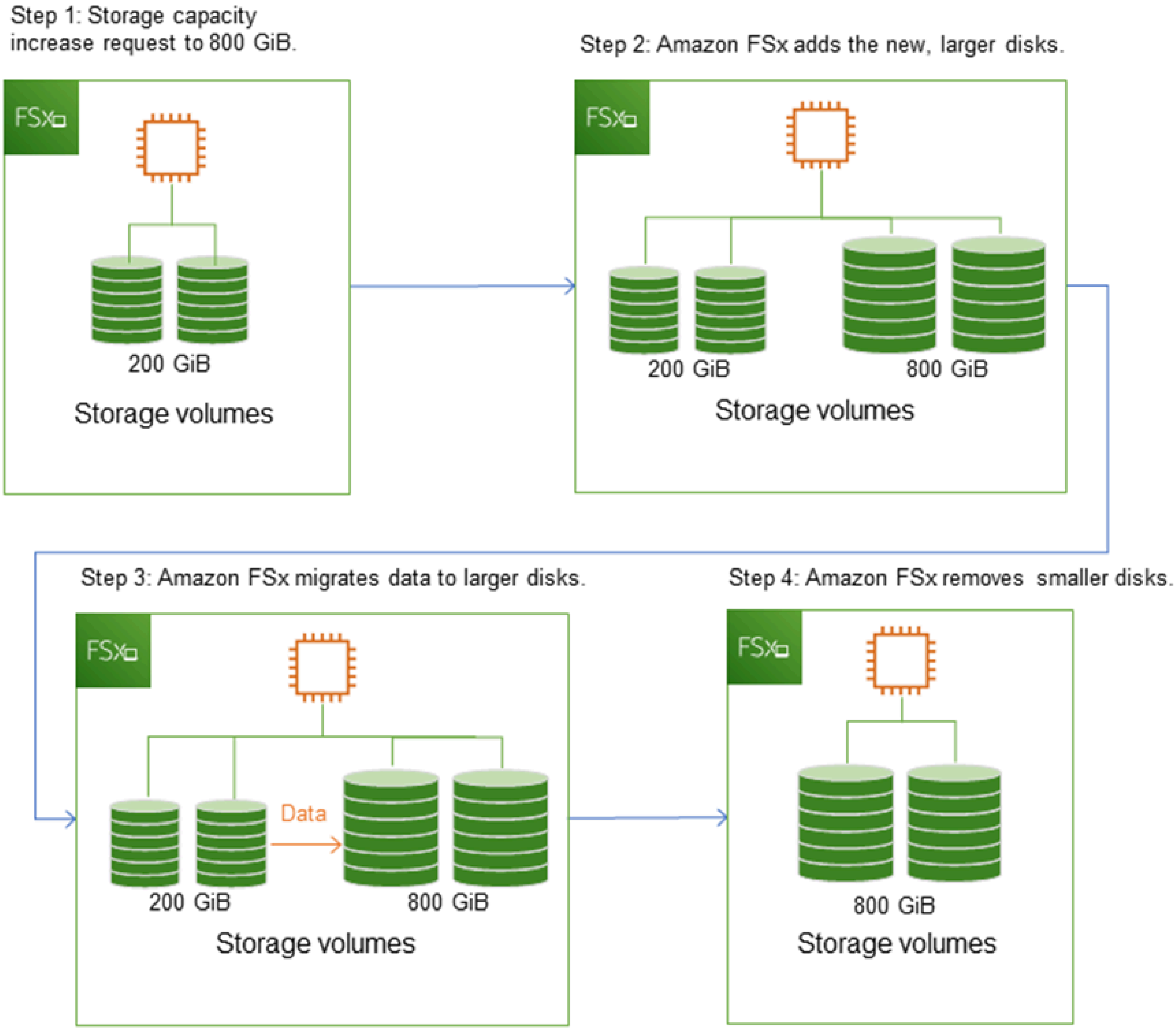
You can increase your FSx for Windows file system's storage capacity as your storage requirements change. You can do so using the Amazon FSx console, the Amazon FSx API, or the AWS Command Line Interface (AWS CLI). Factors to consider when planning a storage capacity increase include knowing when you need to increase storage capacity, understanding how Amazon FSx processes storage capacity increases, and tracking the progress of a storage increase request. You can only increase a file system's storage capacity; you cannot decrease storage capacity.

Note

You can't increase storage capacity for file systems created before June 23, 2019 or file systems restored from a backup belonging to a file system that was created before June 23, 2019.

When you increase the storage capacity of your Amazon FSx file system, Amazon FSx adds a new, larger set of disks to your file system behind the scenes. Amazon FSx then runs a storage optimization process in the background to transparently migrate data from the old disks to the new disks. Storage optimization can take between a few hours and several days, depending on the storage type and other factors, with minimal noticeable impact on the workload performance. During this optimization, backup usage is temporarily higher, because both the old and new storage volumes are included in the file system-level backups. Both sets of storage volumes are included to ensure that Amazon FSx can successfully take and restore from backups even during storage scaling activity. The backup usage reverts to its previous baseline level after the old storage volumes are no longer included in the backup history. When the new storage capacity becomes available, you are billed only for the new storage capacity.

The following illustration shows the four main steps of the process that Amazon FSx uses when increasing a file system's storage capacity.


Step 1: Storage capacity increase request to 800 GiB.
FSx
200 GiB
Storage volumes
Step 2: Amazon FSx adds the new, larger disks.
FSx
200 GiB
800 GiB
Storage volumes
Step 3: Amazon FSx migrates data to larger disks.
FSx
Data
200 GiB
800 GiB
Storage volumes
Step 4: Amazon FSx removes smaller disks.
FSx
800 GiB
Storage volumes

You can track the progress of storage optimization, SSD storage capacity increases, or SSD IOPS updates at any time using the Amazon FSx console, CLI, or API. For more information, see Monitoring storage capacity increases.

What to know about increasing a file system's storage capacity

Here are a few important items to consider when increasing storage capacity:

- **Increase only** – You can only *increase* the amount of storage capacity for a file system; you can't decrease storage capacity.
- **Minimum increase** – Each storage capacity increase must be a minimum of 10 percent of the file system's current storage capacity, up to the maximum allowed value of 65,536 GiB.
- **Minimum throughput capacity** – To increase storage capacity, a file system must have a minimum throughput capacity of 16 MBps. This is because the storage optimization step is a throughput-intensive process.
- **Time between increases** – You can't make further storage capacity increases on a file system until 6 hours after the last increase was requested, or until the storage optimization process has completed, whichever time is longer. Storage optimization can take from a few hours up to a few days to complete. To minimize the time it takes for storage optimization to complete, we recommend increasing your file system's throughput capacity before increasing storage capacity (the throughput capacity can be scaled back down after storage scaling completes), and increasing storage capacity when there is minimal traffic on the file system.

> **Note**
>
> Certain file system events can consume disk I/O performance resources For example: The optimization phase of storage capacity scaling can generate increased disk throughput, and potentially cause performance warnings. For more information, see Performance warnings and recommendations.

Knowing when to increase storage capacity

Increase your file system's storage capacity when it's running low on free storage capacity. Use the `FreeStorageCapacity` CloudWatch metric to monitor the amount of free storage available on the file system. You can create an Amazon CloudWatch alarm on this metric and get notified when it drops below a specific threshold. For more information, see Monitoring with Amazon CloudWatch.

We recommend maintaining at least 20% of free storage capacity at all times on your file system. Using all of your storage capacity can negatively impact your performance and might introduce data inconsistencies.

You can automatically increase your file system's storage capacity when the amount of free storage capacity falls below a defined threshold that you specify. Use the AWS-developed custom

CloudFormation template to deploy all of the components required to implement the automated solution. For more information, see Increasing storage capacity dynamically.

Storage capacity increases and file system performance

Most workloads experience minimal performance impact while Amazon FSx runs the storage optimization process in the background after the new storage capacity is available. However, file systems with HDD storage type and workloads involving large numbers of end users, high levels of I/O, or datasets that have large numbers of small files could temporarily experience reduction in the performance. For these cases, we recommend that you first increase your file system's throughput capacity before increasing storage capacity. For these types of workloads, we also recommend changing throughput capacity during idle periods when there is minimal load on your file system. This enables you to continue providing the same level of throughput to meet your application’s performance needs. For more information, see Managing throughput capacity.

Managing your file system's storage type

You can change your file system storage type from HDD to SSD using the AWS Management Console and AWS CLI. When you change the storage type to SSD, keep in mind that you can't update your file system configuration again until 6 hours after the last update was requested, or until the storage optimization process is complete—whichever time is longer. Storage optimization can take between a few hours and a few days to complete. To minimize this time, we recommend updating your storage type when there is minimal traffic on your file system. For more information, see Updating the storage type of a FSx for Windows file system.

You can't change your file system storage type from SSD to HDD. If you want to change a file system's storage type from SSD to HDD, you will need to restore a backup of the file system to a new file system that you configure to use HDD storage. For more information, see Restoring backups to new file system.

About storage types

You can configure your FSx for Windows File Server file system to use either the solid state drive (SSD) or the magnetic hard disk drive (HDD) storage type.

SSD storage is appropriate for most production workloads that have high performance requirements and latency-sensitivity. Examples of these workloads include databases, data analytics, media processing, and business applications. We also recommend SSD for use cases involving large numbers of end users, high levels of I/O, or datasets that have large numbers of small files. Lastly, we recommend using SSD storage if you plan to enable shadow copies. You can configure and scale SSD IOPS for file systems with SSD storage, but not HDD storage.

HDD storage is designed for a broad range of workloads—including home directories, user and departmental file shares, and content management systems. HDD storage comes at a lower cost relative to SSD storage, but with higher latencies and lower levels of disk throughput and disk IOPS per unit of storage. It might be suitable for general-purpose user shares and home directories with low I/O requirements, large content management systems (CMS) where data is retrieved infrequently, or datasets with small numbers of large files.

For more information, see Storage configuration & performance.

Managing SSD IOPS

For file systems configured with SSD storage, the amount of SSD IOPS determines the amount of disk I/O available when your file system has to read data from and write data to disk, as opposed to data that is in cache. You can select and scale the amount of SSD IOPS independently of storage capacity. The maximum SSD IOPS that you can provision is dependent on the amount of storage capacity and throughput capacity you select for your file system. If you attempt to increase your SSD IOPS above the limit that's supported by your throughput capacity, you might need to increase your throughput capacity to get that level of SSD IOPS. For more information, see FSx for Windows File Server performance and Managing throughput capacity.

Here are a few important items to know about updating a file system's provisioned SSD IOPS:

- Choosing an IOPS mode – there are two IOPS modes to choose from:
  - **Automatic** – choose this mode and Amazon FSx will automatically scale your SSD IOPS to maintain 3 SSD IOPS per GiB of storage capacity, up to 400,000 SSD IOPS per file system.
  - **User-provisioned** – choose this mode so that you can specify the number of SSD IOPS within the range of 96–400,000. Specify a number between 3–50 IOPS per GiB of storage capacity for all AWS Regions where Amazon FSx is available, or between 3–500 IOPS per GiB of storage capacity in US East (N. Virginia), US West (Oregon), US East (Ohio), Europe (Ireland), Asia Pacific (Tokyo), and Asia Pacific (Singapore). When you choose the user-provisiohed mode, and the amount of SSD IOPS you specify is not at least 3 IOPS per GiB, the request fails. For higher levels of provisioned SSD IOPS, you pay for the average IOPS above 3 IOPS per GiB per file system.
- **Storage capacity updates** – If you increase your file system's storage capacity, and the amount requires by default an amount of SSD IOPS that is greater than your current user-provisioned

SSD IOPS level, Amazon FSx automatically switches your file system to Automatic mode and your file system will have a minimum of 3 SSD IOPS per GiB of storage capacity.

- **Throughput capacity updates** – If you increase your throughput capacity, and the maximum SSD IOPS supported by your new throughput capacity is higher than your user-provisioned SSD IOPS level, Amazon FSx automatically switches your file system to Automatic mode.
- **Frequency of SSD IOPS increases** – You can't make further SSD IOPS increases, throughput capacity increases, or storage type updates on a file system until 6 hours after the last increase was requested, or until the storage optimization process has completed—whichever time is longer. Storage optimization can take from a few hours up to a few days to complete. To minimize the time it takes for storage optimization to complete, we recommend scaling SSD IOPS when there is minimal traffic on the file system.

> **Note**
>
> Note that throughput capacity levels of 4,608 MBps and higher are supported only in the following AWS Regions: US East (N. Virginia), US West (Oregon), US East (Ohio), Europe (Ireland), Asia Pacific (Tokyo), and Asia Pacific (Singapore).

For more information about how update the amount of provisioned SSD IOPS for your FSx for Windows File Server file system, see Updating a file system's SSD IOPS.

Reducing storage costs with Data Deduplication

Data Deduplication, often referred to as Dedup for short, helps storage administrators reduce costs that are associated with duplicated data. With FSx for Windows File Server, you can use Microsoft Data Deduplication to identify and eliminate redundant data. Large datasets often have redundant data, which increases the data storage costs. For example:

- User file shares may have many copies of the same or similar files.
- Software development shares can have many binaries that remain unchanged from build to build.

You can reduce your data storage costs by enabling data deduplication for your file system. *Data deduplication* reduces or eliminates redundant data by storing duplicated portions of the dataset only once. When you enable Data Deduplication, Data compression is enabled by default, compressing the data after deduplication for additional savings. Data Deduplication optimizes redundancies without compromising data fidelity or integrity. Data deduplication runs as a background process that continually and automatically scans and optimizes your file system, and it is transparent to your users and connected clients.

The storage savings that you can achieve with data deduplication depends on the nature of your dataset, including how much duplication exists across files. Typical savings average 50–60 percent for general-purpose file shares. Within shares, savings range from 30–50 percent for user documents to 70–80 percent for software development datasets. You can measure potential deduplication savings using the `Measure-FSxDedupFileMetadata` remote PowerShell command described below.

You can also customize data deduplication to meet your specific storage needs. For example, you can configure deduplication to run only on certain file types, or you can create a custom job schedule. Because deduplication jobs can consume file server resources, we recommend monitoring the status of your deduplication jobs using the `Get-FSxDedupStatus`.

For information about configuring data deduplication on your file system, see Managing data deduplication.

For information on resolving issues related to data deduplication, see

Use the following information to help troubleshoot some common issues when configuring and using data deduplication.

Topics

- Data deduplication is not working
- Deduplication values are unexpectedly set to 0
- Space is not freed up on file system after deleting files

Data deduplication is not working

To see the current status of data deduplication, run the `Get-FSxDedupStatus` PowerShell command to view the completion status for the most recent deduplication jobs. If one or more jobs is failing, you may not see an increase in free storage capacity on your file system.

The most common reason for deduplication jobs failing is insufficient memory.

- Microsoft recommends optimally having 1 GB of memory per 1 TB of logical data (or at a minimum 350 MB per 1 TB of logical data). Use the Amazon FSx performance table to determine the memory associated with your file system's throughput capacity and ensure the memory resources are sufficient for the size of your data. If it is not, you need to increase the file system's throughput capacity to the level that meets the memory requirements of 1 GB per 1 TB of logical data.
- Deduplication jobs are configured with the Windows recommended default of 25% memory allocation, which means that for a file system with 32 GB of memory, 8 GB will be available for deduplication. The memory allocation is configurable (using the `Set-FSxDedupSchedule` command with parameter `–Memory`). Be aware that using a higher memory allocation for dedup may impact file system performance.
- You can modify the configuration of deduplication jobs to reduce the amount of memory required. For example, you can constrain the optimization to run on specific file types or folders, or set a minimum file size and age for optimization. We also recommend configuring deduplication jobs to run during idle periods when there is minimal load on your file system.

You may also see errors if deduplication jobs have insufficient time to complete. You may need to change the maximum duration of jobs, as described in Modifying a data deduplication schedule.

If deduplication jobs have been failing for a long period of time, and there have been changes to the data on the file system during this period, subsequent deduplication jobs may require more resources to complete successfully for the first time.

Deduplication values are unexpectedly set to 0

The values for `SavedSpace` and `OptimizedFilesSavingsRate` are unexpectedly 0 for a file system on which you have configured data deduplication.

This can occur during the storage optimization process when you increase the file system's storage capacity. When you increase a file system's storage capacity, Amazon FSx cancels existing data deduplication jobs during the storage optimization process, which migrates data from the old disks to the new, larger disks. Amazon FSx resumes data deduplication on the file system once the storage optimization job completes. For more information about increasing storage capacity and storage optimization, see Managing storage capacity.

Space is not freed up on file system after deleting files

The expected behavior of data deduplication is that if the data that was deleted was something that dedup had saved space on, then the space is not actually freed up on your file system until the garbage collection job runs.

A practice you may find helpful is to set the schedule to run the garbage collection job right after you delete a large number of files. After the garbage collection job finishes, you can set the garbage collection schedule back to its original settings. This ensures you can quickly see the space from your deletions immediately.

Use the following procedure to set the garbage collection job to run in 5 minutes.

1. To verify that data deduplication is enabled, use the `Get-FSxDedupStatus` command. For more information on the command and its expected output, see Viewing the amount of saved space.

2. Use the following to set the schedule to run the garbage collection job 5 minutes from now.

```
$FiveMinutesFromNowUTC = ((get-date).AddMinutes(5)).ToUniversalTime()
$DayOfWeek = $FiveMinutesFromNowUTC.DayOfWeek
$Time = $FiveMinutesFromNowUTC.ToString("HH:mm")

Invoke-Command -ComputerName ${RPS_ENDPOINT} -ConfigurationName FSxRemoteAdmin -
ScriptBlock {
    Set-FSxDedupSchedule -Name "WeeklyGarbageCollection" -Days $Using:DayOfWeek -
Start $Using:Time -DurationHours 9
}
```

3. After the garbage collection job has run and the space has been freed up, set the schedule back to its original settings.

.

For more information about data deduplication, see the Microsoft Understanding Data Deduplication documentation.

⚠ **Warning**

It is not recommended to run certain Robocopy commands with data deduplication because these commands can impact the data integrity of the Chunk Store. For more information, see the Microsoft Data Deduplication interoperability documentation.

Best practices when using data deduplication

Here are some best practices for using Data Deduplication:

- **Schedule Data Deduplication jobs to run when your file system is idle**: The default schedule includes a weekly `GarbageCollection` job at 2:45 UTC on Saturdays. It can take multiple hours to complete if you have a large amount of data churn on your file system. If this time isn't ideal for your workload, schedule this job to run at a time when you expect low traffic on your file system.
- **Configure sufficient throughput capacity for Data Deduplication to complete**: Higher throughput capacities provide higher levels of memory. Microsoft recommends having 1 GB of memory per 1 TB of logical data to run Data Deduplication. Use the Amazon FSx performance table to determine the memory that's associated with your file system's throughput capacity and ensure that the memory resources are sufficient for the size of your data.
- **Customize Data Deduplication settings to meet your specific storage needs and reduce performance requirements**: You can constrain the optimization to run on specific file types or folders, or set a minimum file size and age for optimization. To learn more, see Reducing storage costs with Data Deduplication.

Managing storage quotas

You can configure user storage quotas on your file systems to limit how much data storage that users can consume. After you set quotas, you can track quota status to monitor usage and see when users surpass their quotas.

You can also enforce quotas by stopping users who reach their quotas from writing to the storage space. When you enforce quotas, a user that exceeds their quota receives an "insufficient disk space" error message.

You can set these thresholds for quota settings:

- Warning – used to track whether a user or group is approaching their quota limit, relevant for tracking only.
- Limit – the storage quota limit for a user or group.

You can configure default quotas that are applied to new users who access a file system and quotas that apply to specific users or groups. You can also view a report of how much storage each user or group is consuming and whether they're surpassing their quotas.

Storage consumption at a user level is tracked based on file ownership. Storage consumption is calculated using logical file size, not the actual physical storage space that files occupy. User storage quotas are tracked at the time when data is written to a file.

Updating quotas for multiple users requires either running the update command once for each user, or organizing the users into a group and updating the quota for that group.

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 Using the Amazon FSx CLI for PowerShell.

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

User storage quotas command	Description
Enable-FSxUserQuotas	Starts tracking or enforcing user storage quotas, or both.
Disable-FSxUserQuotas	Stops tracking and enforcement for user storage quotas.
Get-FSxUserQuotaSettings	Retrieves the current user-storage quota settings for the file system.
Get-FSxUserQuotaEntries	Retrieves the current user-storage quota entries for individual users and groups on the file system.
Set-FSxUserQuotas	Set the user storage quota for an individual user or group. Quota values are specified in bytes.

The online help for each command provides a reference of all command options. To access this help, run the command with **-?**, for example **Enable-FSxUserQuotas -?**.

Increasing file system storage capacity

You can increase your FSx for Windows File Server file system's storage capacity as your storage requirements change. Use the Amazon FSx console, the AWS CLI, or the Amazon FSx API to increase a file system's storage capacity as described in the following procedures. For more information, see Managing storage capacity.

To increase storage capacity for a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems** and choose the Windows file system that you want to increase storage capacity for.
3. For **Actions**, choose **Update storage**. Or, in the **Summary** panel, choose **Update** next to the file system's **Storage capacity**.

   The **Update storage capacity** window appears.
4. For **Input type**, choose **Percentage** to enter the new storage capacity as a percentage change from the current value, or choose **Absolute** to enter the new value in GiB.
5. Enter the **Desired storage capacity**.

   > **Note**
   >
   > The desired capacity value must be at least 10 percent larger than the current value, up to the maximum value of 65,536 GiB.
6. Choose **Update** to initiate the storage capacity update.
7. You can monitor the update progress on the **File systems** detail page, in the **Updates** tab.

To increase storage capacity for a file system (CLI)

To increase the storage capacity for an FSx for Windows File Server file system, use the AWS CLI command update-file-system. Set the following parameters:

- `--file-system-id` to the ID of the file system you are updating.
- `--storage-capacity` to a value that is at least 10 percent greater than the current value.

You can monitor the progress of the update by using the AWS CLI command describe-file-systems. Look for the `administrative-actions` in the output.

For more information, see AdministrativeAction.

Monitoring storage capacity increases

After increasing your file system's storage capacity, you can monitor the progress of the storage capacity increase using the Amazon FSx console, the API, or the AWS CLI as described in the following procedures.

Monitoring increases in the console

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

For storage capacity updates, you can view the following information.

Update type

Possible values are **Storage capacity**.

Target value

The desired value to update the file system's storage capacity to.

Status

The current status of the update. For storage capacity updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Updated optimizing** – Amazon FSx has increased the file system's storage capacity. The storage optimization process is now moving the file system data to the new larger disks.
- **Completed** – The storage capacity increase completed successfully.
- **Failed** – The storage capacity increase failed. Choose the question mark (**?**) to see details on why the storage update failed.

Progress %

Displays the progress of the storage optimization process as percent complete.

Request time

The time that Amazon FSx received the update action request.

Monitoring increases with the AWS CLI and API

You can view and monitor file system storage capacity increase requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you increase a file system's storage capacity, two `AdministrativeActions` are generated: a `FILE_SYSTEM_UPDATE` and a `STORAGE_OPTIMIZATION` action.

The following example shows an excerpt of the response of a **describe-file-systems** CLI command. The file system has a storage capacity of 300 GB, and there is a pending administrative action to increase the storage capacity to 1000 GB.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 300,
            "AdministrativeActions": [
                {
                     "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                     "RequestTime": 1581694764.757,
                     "Status": "PENDING",
                     "TargetFileSystemValues": {
                         "StorageCapacity": 1000
                     }
                },
                {
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
                    "Status": "PENDING",
                }
            ]
```

Amazon FSx processes the FILE_SYSTEM_UPDATE action first, adding the new larger storage disks to the file system. When the new storage is available to the file system, the FILE_SYSTEM_UPDATE status changes to UPDATED_OPTIMIZING. The storage capacity shows the new larger value, and Amazon FSx begins processing the STORAGE_OPTIMIZATION administrative action. This is shown in the following excerpt of the response of a **describe-file-systems** CLI command.

The ProgressPercent property displays the progress of the storage optimization process. After the storage optimization process completes successfully, the status of the FILE_SYSTEM_UPDATE action changes to COMPLETED, and the STORAGE_OPTIMIZATION action no longer appears.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 1000,
            "AdministrativeActions": [
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "RequestTime": 1581694764.757,
                    "Status": "UPDATED_OPTIMIZING",
                    "TargetFileSystemValues": {
                        "StorageCapacity": 1000
                }
                },
                {
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
                    "Status": "IN_PROGRESS",
                    "ProgressPercent": 50,
                }
            ]
```

If the storage capacity increase fails, the status of the FILE_SYSTEM_UPDATE action changes to FAILED. The FailureDetails property provides information about the failure, shown in the following example.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 300,
            "AdministrativeActions": [
                {
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "FailureDetails": {
                        "Message": "string"
                    },
                    "RequestTime": 1581694764.757,
                    "Status": "FAILED",
                    "TargetFileSystemValues":
                        "StorageCapacity": 1000
                }
            ]
```

For information about troubleshooting failed actions, see Storage or throughput capacity updates fail.

Increasing the storage capacity of an FSx for Windows File Server file system dynamically

As an alternative to manually increasing your FSx for Windows File Server file system's storage capacity as the amount of data stored increases, you can use a CloudFormation template to increase storage automatically. The solution presented in the this section dynamically increases a file system's storage capacity when the amount of free storage capacity falls below a defined threshold that you specify.

This AWS CloudFormation template automatically deploys all of the components that are required to define the free storage capacity threshold, the Amazon CloudWatch alarm based on this threshold, and the AWS Lambda function that increases the file system’s storage capacity.

The solution takes in the following parameters:

- The file system ID
- The free storage capacity threshold (numerical value)
- Unit of measurement (percentage [default] or GiB)

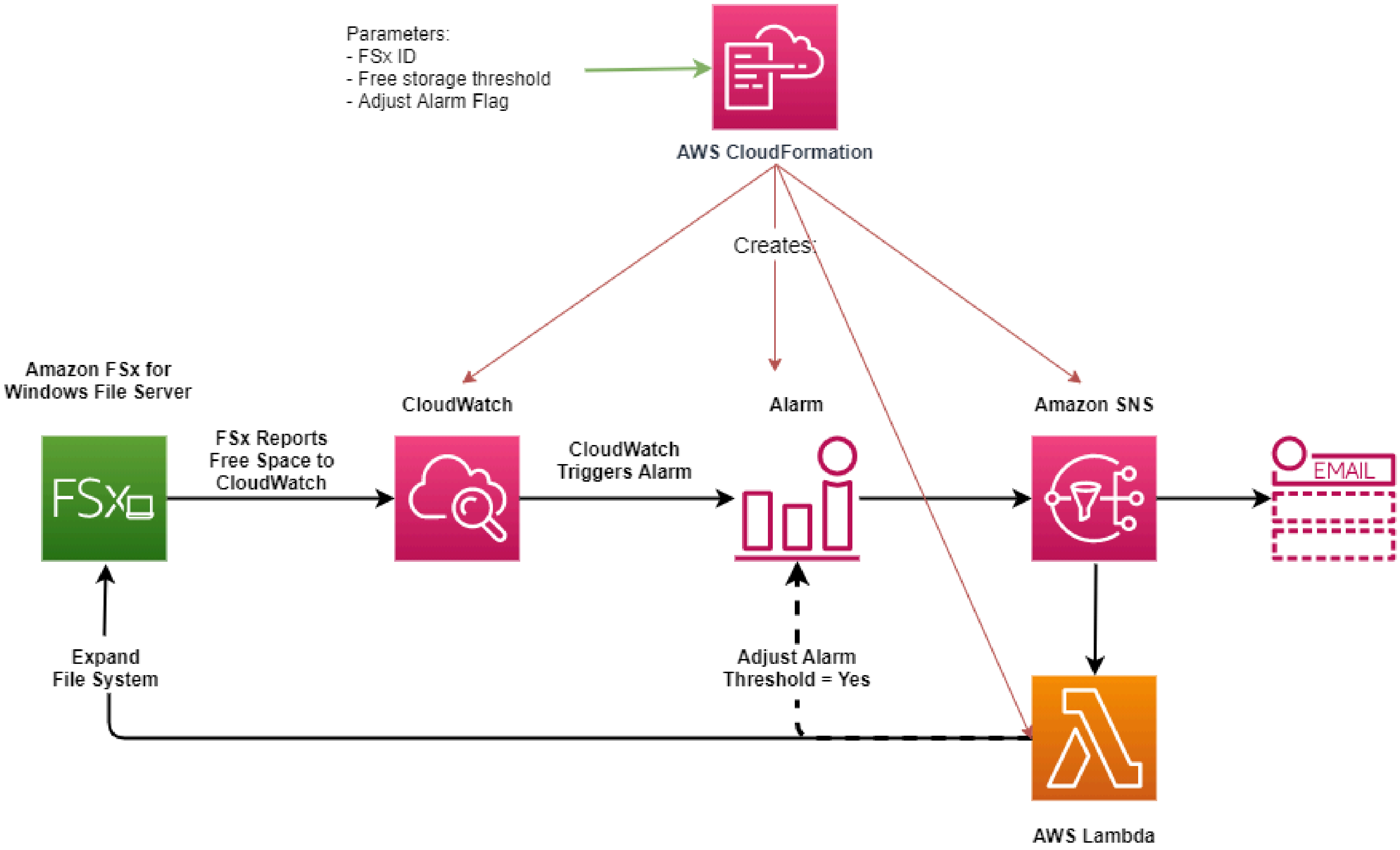
- The percentage by which to increase the storage capacity (%)
- The email address for the SNS subscription
- Adjust alarm threshold (Yes/No)

Topics

- Architecture overview
- CloudFormation template
- Automated deployment with CloudFormation

Architecture overview

Deploying this solution builds the following resources in the AWS Cloud.



The diagram illustrates the following steps:

1. The CloudFormation template deploys a CloudWatch alarm, an AWS Lambda function, an Amazon Simple Notification Service (Amazon SNS) queue, and all required AWS Identity and

Access Management (IAM) roles. The IAM role gives the Lambda function permission to invoke the Amazon FSx API operations.

2. CloudWatch triggers an alarm when the file system's free storage capacity goes below the specified threshold, and sends a message to the Amazon SNS queue.
3. The solution then triggers the Lambda function that is subscribed to this Amazon SNS topic.
4. The Lambda function calculates the new file system storage capacity based on the specified percent increase value and sets the new file system storage capacity.
5. The Lambda function can optionally adjust the free storage capacity threshold so that it is equal to a specified percentage of the file system's new storage capacity.
6. The original CloudWatch alarm state and results of the Lambda function operations are sent to the Amazon SNS queue.

To receive notifications about the actions that are performed as a response to the CloudWatch alarm, you must confirm the Amazon SNS topic subscription by following the link provided in the **Subscription Confirmation** email.

CloudFormation template

This solution uses CloudFormation to automate deploying the components that are used to automatically increase the storage capacity of an FSx for Windows File Server file system. To use this solution, download the IncreaseFSxSize CloudFormation template.

The template uses the **Parameters** described as follows. Review the template parameters and their default values, and modify them for the needs of your file system.

FileSystemId

No default value. The ID of the file system for which you want to automatically increase the storage capacity.

LowFreeDataStorageCapacityThreshold

No default value. Specifies the initial free storage capacity threshold at which to trigger an alarm and automatically increase the file system's storage capacity, specified in GiB or as a percentage (%) of the file system's current storage capacity. When expressed as a percentage, the CloudFormation template re-calculates to GiB to match the CloudWatch alarm settings.

LowFreeDataStorageCapacityThresholdUnit

Default is **%**. Specifies the units for the `LowFreeDataStorageCapacityThreshold`, either in GiB or as a percentage of the current storage capacity.

AlarmModificationNotification

Default is **Yes**. If set to Yes, the initial `LowFreeDataStorageCapacityThreshold`, is increased proportionally to the value of `PercentIncrease` for subsequent alarm thresholds.

For example, when `PercentIncrease` is set to 20, and AlarmModificationNotification is set to Yes, the available free space threshold (`LowFreeDataStorageCapacityThreshold`) specified in GiB is increased by 20% for subsequent storage capacity increase events.

EmailAddress

No default value. Specifies the email address to use for the SNS subscription and receives storage capacity threshold alerts.

PercentIncrease

No default value. Specifies the amount by which to increase the storage capacity, expressed as a percentage of the current storage capacity.

Automated deployment with CloudFormation

The following procedure configures and deploys an CloudFormation stack to automatically increase the storage capacity of an FSx for Windows File Server file system. It takes about 5 minutes to deploy.

> **Note**
>
> Implementing this solution incurs billing for the associated AWS services. For more information, see the pricing details pages for those services.

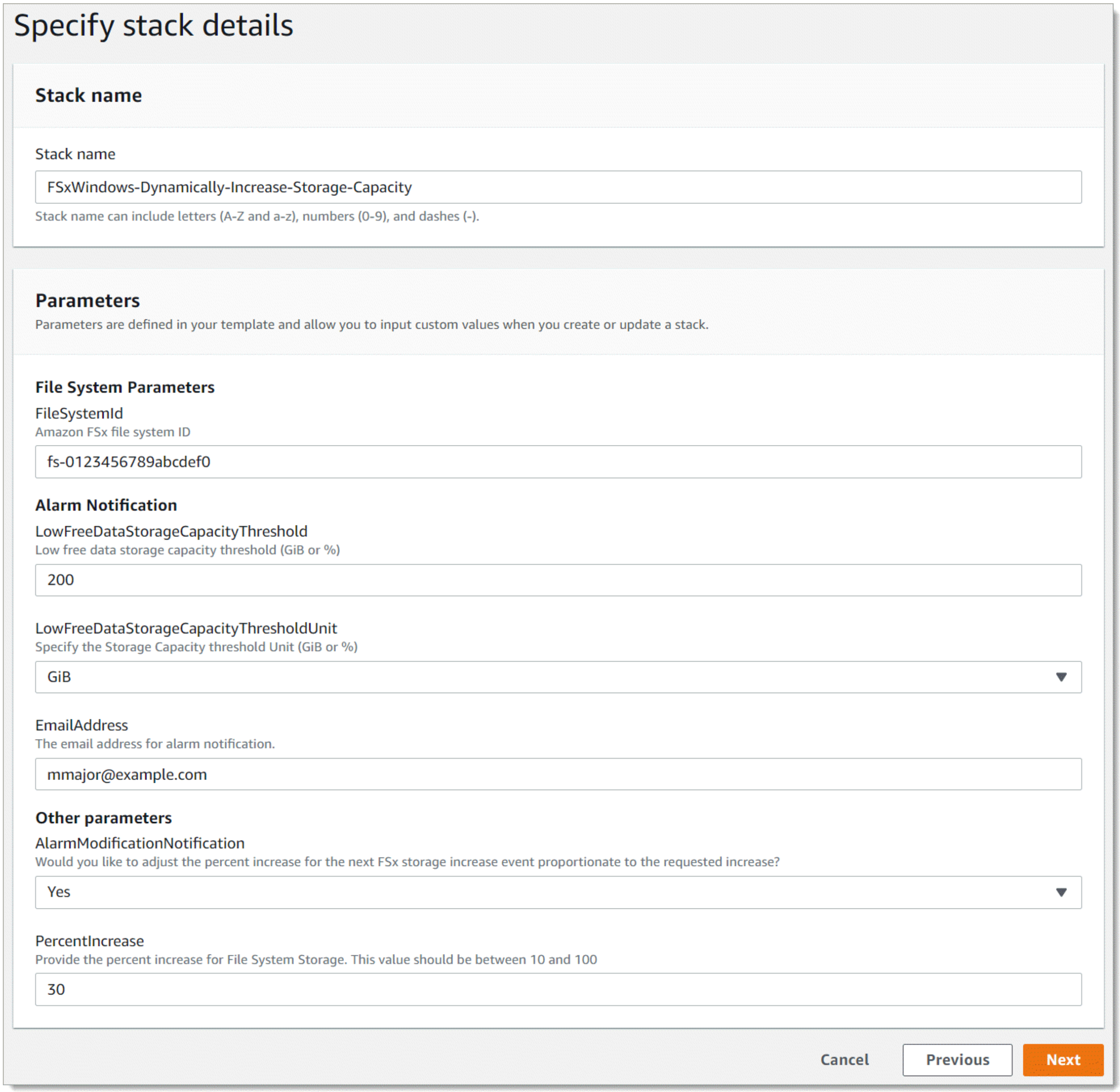
Before you start, you must have the ID of the Amazon FSx file system running in an Amazon Virtual Private Cloud (Amazon VPC) in your AWS account. For more information about creating Amazon FSx resources, see Getting started with Amazon FSx for Windows File Server.

To launch the automatic storage capacity increase solution stack

1. Download the IncreaseFSxSize CloudFormation template. For more information about creating a CloudFormation stack, see Creating a stack on the AWS CloudFormation console in the *AWS CloudFormation User Guide*.

   > **Note**
   >
   > Amazon FSx is currently only available in specific AWS Regions. You must launch this solution in an AWS Region where Amazon FSx is available. For more information, see Amazon FSx endpoints and quotas in the *AWS General Reference*.

2. In **Specify stack details**, enter the values for your automatic storage capacity increase solution.



Specify stack details

Stack name

Stack name

FSxWindows-Dynamically-Increase-Storage-Capacity

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

File System Parameters

FileSystemId

Amazon FSx file system ID

fs-0123456789abcdef0

Alarm Notification

LowFreeDataStorageCapacityThreshold

Low free data storage capacity threshold (GiB or %)

200

LowFreeDataStorageCapacityThresholdUnit

Specify the Storage Capacity threshold Unit (GiB or %)

GiB ▼

EmailAddress

The email address for alarm notification.

mmajor@example.com

Other parameters

AlarmModificationNotification

Would you like to adjust the percent increase for the next FSx storage increase event proportionate to the requested increase?

Yes ▼

PercentIncrease

Provide the percent increase for File System Storage. This value should be between 10 and 100

30

3. Enter a **Stack name**.
4. For **Parameters**, review the parameters for the template and modify them for the needs of your file system. Then choose **Next**.
5. Enter any **Options** settings that you want for your custom solution, and then choose **Next**.
6. For **Review**, review and confirm the solution settings. You must select the check box acknowledging that the template creates IAM resources.
7. Choose **Create** to deploy the stack.

You can view the status of the stack in the CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in about 5 minutes.

Updating the stack

After the stack is created, you can update it by using the same template and providing new values for the parameters. For more information, see Updating stacks directly in the *AWS CloudFormation User Guide*.

Updating the storage type of a FSx for Windows file system

You can change the storage type of a file system that uses HDD storage to use SSD storage. You can use the the Amazon FSx console, the AWS CLI, or the Amazon FSx API to change a file system's storage type, as shown in the following procedures. For more information, see Managing your file system's storage type.

To update a file system's storage type (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems** and choose the Windows file system that you want to update the storage type for.
3. Under **Actions**, choose **Update storage type**. Or, in the **Summary** panel, select the **Update** button next to **HDD**. The **Update storage type** window appears.
4. For **Desired storage type**, choose **SSD**. Choose **Update** to initiate the storage type update.

   You can monitor the progress of the storage type update using the console and the CLI.

To update a file system's storage type (CLI)

To update storage type for an FSx for Windows File Server file system, use the AWS CLI command update-file-system. Set the following parameters:

- `--file-system-id` to the ID of the file system that you want to update.
- `--storage-type` to SSD. You can't switch from SSD storage type to HDD storage type.

You can monitor the progress of the update by using the AWS CLI command describe-file-systems. Look for the `administrative-actions` in the output.

For more information, see AdministrativeAction.

Monitoring storage type updates

After you update your file system's storage type from HDD to SSD storage, you can monitor the progress of the storage type update using the Amazon FSx console, the AWS CLI, or the API, as described in the following procedures.

Monitoring file system updates in the console

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

For storage type updates, you can view the following information.

Update type

Possible value is **Storage type**.

Target value

SSD

Status

The current status of the update. For storage type updates, the possible values are as follows:

- **Pending** – Amazon FSx received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Updated optimizing** – The SSD storage performance is available for write operations. The update enters an **Updated optimizing** state, which typically lasts a few hours, during which read operations will have performance levels between HDD and SSD. Once your update action is complete, your new SSD performance is available for both reads and writes.
- **Completed** – The storage type update completed successfully.
- **Failed** – The storage type update failed. Choose the question mark (**?**) to see details.

Progress %

Displays the progress of the storage optimization process by the percentage that's complete.

Request time

The time that Amazon FSx received the update action request.

Monitoring updates with the AWS CLI and API

You can view and monitor file system storage type update requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you increase a file system's SSD IOPS, two `AdministrativeActions` are generated: a `FILE_SYSTEM_UPDATE` and a `STORAGE_TYPE_OPTIMIZATION` action.

Updating a file system's SSD IOPS

For file systems configured with SSD storage, the level of provisioned SSD IOPS determines the amount of disk I/O available when your file system has to read data from and write data to disk, as opposed to reading or writing data that is in cache. You can update SSD IOPS for a file system using the Amazon FSx console, the AWS CLI, or the Amazon FSx API, as described in the following procedures. For more information about managing SSD IOPS, see Managing SSD IOPS.

To update SSD IOPS for a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems** and choose the Windows file system that you want to update SSD IOPS for.
3. Under **Actions**, choose **Update SSD IOPS**. Or, in the **Summary** panel, select the **Update** button next to **Provisioned SSD IOPS**. The **Update IOPS provisioning** window opens.
4. For **Mode**, choose **Automatic** or **User-provisioned**. If you choose **Automatic**, Amazon FSx automatically provisions 3 SSD IOPS per GiB of storage capacity for your file system. If you choose **User-provisioned**, enter any whole number in the range of 96–400,000.
5. Choose **Update** to initiate the provisioned SSD IOPS update.
6. You can monitor the update progress on the **File systems** detail page, on the **Updates** tab.

To update SSD IOPS for a file system (CLI)

To update SSD IOPS for an FSx for Windows File Server file system, use the `--windows-configuration DiskIopsConfiguration` property. This property has two parameters, `Iops` and `Mode`:

- If you want to specify the number of SSD IOPS, use `Iops=`*`number_of_IOPS`*, up to a maximum of 400,000 in supported AWS Regions and `Mode=USER_PROVISIONED`.

- If you want Amazon FSx to increase your SSD IOPS automatically, use `Mode=AUTOMATIC` and don't use the `Iops` parameter. Amazon FSx automatically maintains 3 SSD IOPS per GiB of storage capacity on your file system, up to a maximum of 400,000 in supported AWS Regions.

You can monitor the progress of the update by using the AWS CLI command describe-file-systems. Look for the `administrative-actions` in the output.

For more information, see AdministrativeAction.

Monitoring provisioned SSD IOPS updates

After you update the amount of provisioned SSD IOPS for your file system, you can monitor the progress of the SSD IOPS update using the Amazon FSx console, the AWS CLI, and the API, as described in the following procedures.

Monitoring updates in the console

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

For provisioned SSD IOPS updates, you can view the following information.

Update type

Possible values are **IOPS Mode** and **SSD IOPS**.

Target value

The desired value to update the file system's IOPS mode and SSD IOPS to.

Status

The current status of the update. For SSD IOPS updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Updated optimizing** – The new IOPS level is available for your workload's write operations. Your update enters an **Updated optimizing** state, which typically lasts a few hours, during which your workload's read operations have IOPS performance between the previous level and the new level. After your update action is complete, your new IOPS level is available for both reads and writes.

- **Completed** – The SSD IOPS update completed successfully.
- **Failed –** The SSD IOPS update failed. Choose the question mark **(?)** to see details on why the storage update failed.

Progress %

Displays the progress of the storage optimization process as percent complete.

Request time

The time that Amazon FSx received the update action request.

Monitoring updates with the AWS CLI and API

You can view and monitor file system SSD IOPS update requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you increase a file system's SSD IOPS, two `AdministrativeActions` are generated: a `FILE_SYSTEM_UPDATE` and an `IOPS_OPTIMIZATION` action.

Managing data deduplication

You can manage your file system's data deduplication settings using the Amazon FSx CLI for remote management on PowerShell. For more information about using the Amazon FSx CLI remote management on PowerShell, see Using the Amazon FSx CLI for PowerShell.

Following are commands that you can use for data deduplication.

Data deduplication command	Description
Enable-FSxDedup	Enables data deduplication on the file share. Data compression after deduplication is enabled by default when you enable data deduplication.
Disable-FSxDedup	Disables data deduplication on the file share.
Get-FSxDedupConfiguration	Retrieves deduplication configuration information, including Minimum file size and age for optimization, compression settings, and Excluded file types and folders.

Data deduplication command	Description
Set-FSxDedupConfiguration	Changes the deduplication configuration settings, including minimum file size and age for optimization, compression settings, and excluded file types and folders.
Get-FSxDedupStatus	Retrieve the deduplication status, and include read-only properties that describe optimization savings and status on the file system, times, and completion status for the last dedup jobs on the file system.
Get-FSxDedupMetadata	Retrieves deduplication optimization metadata.
Update-FSxDedupStatus	Computes and retrieves updated data deduplication savings information.
Measure-FSxDedupFileMetadata	Measures and retrieves the potential storage space that you can reclaim on your file system if you delete a group of folders. Files often have chunks that are shared across other folders, and the deduplication engine calculates which chunks are unique and would be deleted.
Get-FSxDedupSchedule	Retrieves deduplication schedules that are currently defined.
New-FSxDedupSchedule	Create and customize a data deduplication schedule.
Set-FSxDedupSchedule	Change configuration settings for existing data deduplication schedules.
Remove-FSxDedupSchedule	Delete a deduplication schedule.
Get-FSxDedupJob	Get status and information for all currently running or queued deduplication jobs.
Stop-FSxDedupJob	Cancel one or more specified data deduplication jobs.

The online help for each command provides a reference of all command options. To access this help, run the command with **-?**, for example **Enable-FSxDedup -?**.

Enabling data deduplication

You enable data deduplication on an Amazon FSx for Windows File Server file share using the `Enable-FSxDedup` command, as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

When you enable data deduplication, a default schedule and configuration are created. You can create, modify, and remove schedules and configurations using the commands below.

You can use the `Disable-FSxDedup` command to disable data deduplication entirely on your file system.

Creating a data deduplication schedule

Although the default schedule works well in most cases, you can create a new deduplication schedule by using the `New-FsxDedupSchedule` command, shown as follows. Data deduplication schedules use UTC time.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
New-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days Mon,Wed,Sat -
Start 08:00 -DurationHours 7
}
```

This command creates a schedule named `CustomOptimization` that runs on days Monday, Wednesday, and Saturday, starting the job at 8:00 am (UTC) each day, with a maximum duration of 7 hours, after which the job stops if it is still running.

Note that creating new, custom deduplication job schedules does not override or remove the existing default schedule. Before creating a custom deduplication job, you may want to disable the default job if you don’t need it.

You can disable the default deduplication schedule by using the `Set-FsxDedupSchedule` command, shown as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com
 -ConfigurationName FSxRemoteAdmin -ScriptBlock {Set-FSxDedupSchedule -Name
 “BackgroundOptimization” -Enabled $false}
```

You can remove a deduplication schedule by using the `Remove-FSxDedupSchedule -Name "ScheduleName"` command. Note that the default `BackgroundOptimization` deduplication schedule cannot be modified or removed and will need to be disabled instead.

Modifying a data deduplication schedule

You can modify an existing deduplication schedule by using the `Set-FsxDedupSchedule` command, shown as follows.

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days
 Mon,Tues,Wed,Sat -Start 09:00 -DurationHours 9
}
```

This command modifies the existing `CustomOptimization` schedule to run on days Monday to Wednesday and Saturday, starting the job at 9:00 am (UTC) each day, with a maximum duration of 9 hours, after which the job stops if it is still running.

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

Viewing the amount of saved space

To view the amount of disk space you are saving from running data deduplication, use the `Get-FSxDedupStatus` command, as follows.

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

OptimizedFilesCount OptimizedFilesSize SavedSpace OptimizedFilesSavingsRate
------------------- ------------------ ---------- -------------------------
              12587           31163594   25944826                        83
```

Note

The values shown in the command response for following parameters are not reliable, and you should not use these values: Capacity, FreeSpace, UsedSpace, UnoptimizedSize, and SavingsRate.

Troubleshooting data deduplication

Use the following information to help troubleshoot some common issues when configuring and using data deduplication.

Topics

- Data deduplication is not working
- Deduplication values are unexpectedly set to 0
- Space is not freed up on file system after deleting files

Data deduplication is not working

To see the current status of data deduplication, run the `Get-FSxDedupStatus` PowerShell command to view the completion status for the most recent deduplication jobs. If one or more jobs is failing, you may not see an increase in free storage capacity on your file system.

The most common reason for deduplication jobs failing is insufficient memory.

- Microsoft recommends optimally having 1 GB of memory per 1 TB of logical data (or at a minimum 350 MB per 1 TB of logical data). Use the Amazon FSx performance table to determine the memory associated with your file system's throughput capacity and ensure the memory resources are sufficient for the size of your data. If it is not, you need to increase the file system's throughput capacity to the level that meets the memory requirements of 1 GB per 1 TB of logical data.
- Deduplication jobs are configured with the Windows recommended default of 25% memory allocation, which means that for a file system with 32 GB of memory, 8 GB will be available for deduplication. The memory allocation is configurable (using the `Set-FSxDedupSchedule` command with parameter `–Memory`). Be aware that using a higher memory allocation for dedup may impact file system performance.

- You can modify the configuration of deduplication jobs to reduce the amount of memory required. For example, you can constrain the optimization to run on specific file types or folders, or set a minimum file size and age for optimization. We also recommend configuring deduplication jobs to run during idle periods when there is minimal load on your file system.

You may also see errors if deduplication jobs have insufficient time to complete. You may need to change the maximum duration of jobs, as described in Modifying a data deduplication schedule.

If deduplication jobs have been failing for a long period of time, and there have been changes to the data on the file system during this period, subsequent deduplication jobs may require more resources to complete successfully for the first time.

Deduplication values are unexpectedly set to 0

The values for `SavedSpace` and `OptimizedFilesSavingsRate` are unexpectedly 0 for a file system on which you have configured data deduplication.

This can occur during the storage optimization process when you increase the file system's storage capacity. When you increase a file system's storage capacity, Amazon FSx cancels existing data deduplication jobs during the storage optimization process, which migrates data from the old disks to the new, larger disks. Amazon FSx resumes data deduplication on the file system once the storage optimization job completes. For more information about increasing storage capacity and storage optimization, see Managing storage capacity.

Space is not freed up on file system after deleting files

The expected behavior of data deduplication is that if the data that was deleted was something that dedup had saved space on, then the space is not actually freed up on your file system until the garbage collection job runs.

A practice you may find helpful is to set the schedule to run the garbage collection job right after you delete a large number of files. After the garbage collection job finishes, you can set the garbage collection schedule back to its original settings. This ensures you can quickly see the space from your deletions immediately.

Use the following procedure to set the garbage collection job to run in 5 minutes.

1. To verify that data deduplication is enabled, use the `Get-FSxDedupStatus` command. For more information on the command and its expected output, see Viewing the amount of saved space.

2. Use the following to set the schedule to run the garbage collection job 5 minutes from now.

```
$FiveMinutesFromNowUTC = ((get-date).AddMinutes(5)).ToUniversalTime()
$DayOfWeek = $FiveMinutesFromNowUTC.DayOfWeek
$Time = $FiveMinutesFromNowUTC.ToString("HH:mm")

Invoke-Command -ComputerName ${RPS_ENDPOINT} -ConfigurationName FSxRemoteAdmin -
ScriptBlock {
    Set-FSxDedupSchedule -Name "WeeklyGarbageCollection" -Days $Using:DayOfWeek -
Start $Using:Time -DurationHours 9
}
```

3. After the garbage collection job has run and the space has been freed up, set the schedule back to its original settings.

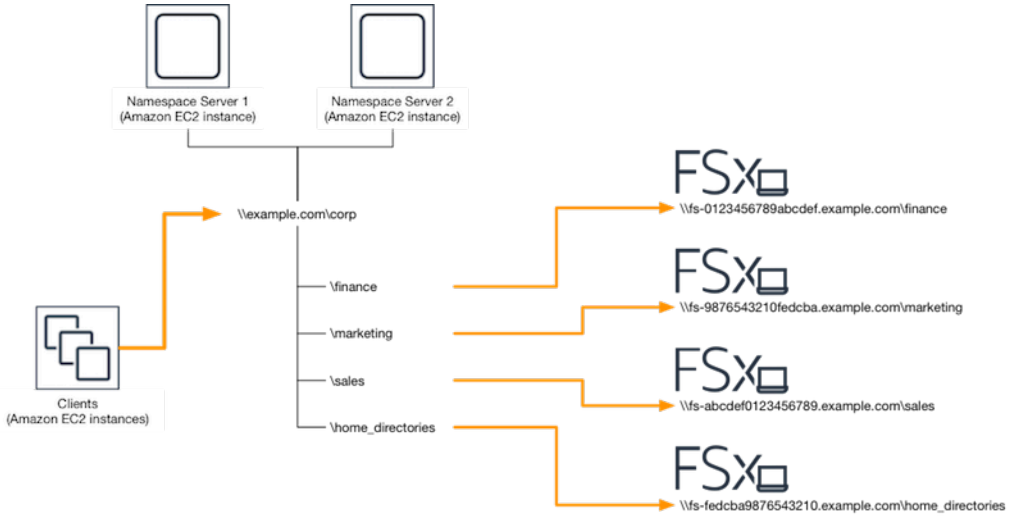
Using DFS Namespaces

DFS Namespaces is a Windows Server role service that you use to group shared folders located on different servers into one or more logically structured namespaces. This makes it possible to give users a virtual view of shared folders, where a single path leads to files located on multiple file systems, as shown in the following diagram. In addition to organizing and unifying access to your file shares across multiple file systems,

Group multiple FSx for Windows File Server file systems with DFS Namespaces

You can use Microsoft's Distributed File System (DFS) Namespaces to group file shares on multiple FSx for Windows File Server file systems into one common folder structure, or namespace. Using DFS Namespaces, you can scale file storage beyond the maximum storage capacity of single file system (64 TiB) for large file datasets—up to hundreds of petabytes. This section shows you how to set up DFS namespaces on multiple FSx for Windows File Server file systems.

DFS Namespaces is a Windows Server role service that you use to group shared folders located on different servers into one or more logically structured namespaces. This makes it possible to give users a virtual view of shared folders, where a single path leads to files located on multiple file systems, as shown in the following diagram. In addition to organizing and unifying access to your file shares across multiple file systems,



For a step-by-step procedure for grouping FSx for Windows file systems using DFS Namespaces, see Group multiple file systems under a single namespace.

Improving 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 the solution provided in Sharding data using DFS Namespaces for scale-out performance to distribute read/write access to your data uniformly across your data multiple FSx for Windows File Server file systems.

Group multiple file systems under a single namespace

In this procedure, you will create a single domain-based namespace (`example.com\corp`) on two namespace servers, in order to consolidate file shares stored on multiple FSx for Windows file systems (finance, marketing, sales, home_directories). You will also set up four file shares under the namespace, each transparently redirecting users to shares hosted on separate FSx for Windows file systems. This enables your users to access file shares using a common namespace instead of having to specify the DNS names for each of the file systems hosting the file shares.

> **Note**
>
> Amazon FSx cannot be added to the root of the DFS share path.

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 CloudFormation template. For more information on creating an 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*.
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.

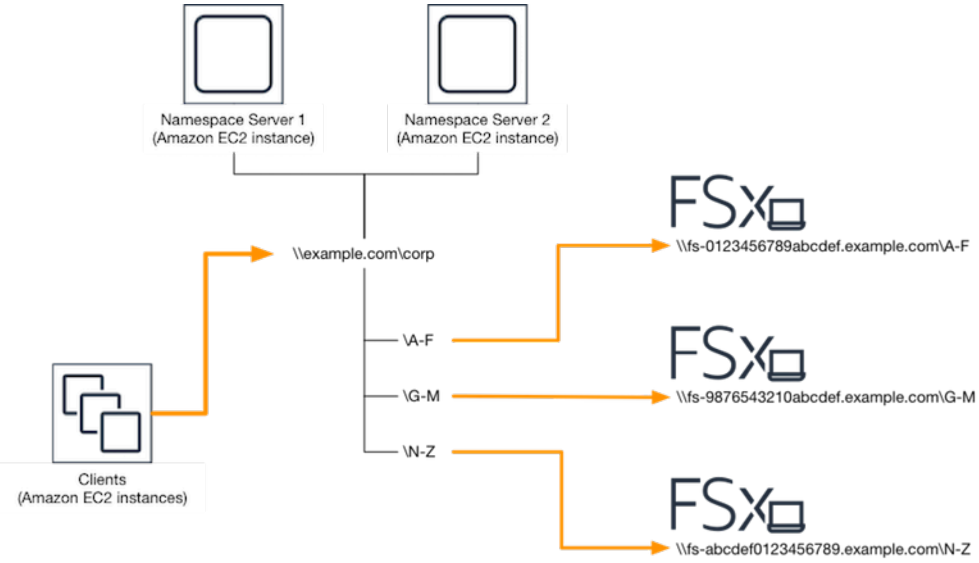
    b. From the **Create Share** dialog, choose **Browse**.

    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.

Sharding data using DFS Namespaces for scale-out performance

The following procedure guides you through creating a DFS solution on Amazon FSx for scale-out performance. In this example, the data stored in the *`corp`* namespace is sharded alphabetically. Data files ‘A-F’, ‘G-M’ and ‘N-Z’ are all stored on different file shares. Based on the type of data, I/O size, and I/O access pattern, you should decide how to best shard your data across multiple file shares. Choose a sharding convention that distributes I/O evenly across all the file shares you plan on using. Keep in mind that each namespace supports up to 50,000 file shares and hundreds of petabytes of storage capacity in aggregate.



To set up DFS Namespaces for scale-out performance

1. If you don't already have DFS Namespace servers running, you can launch a pair of highly available DFS Namespace servers using the setup-DFSN-servers.template CloudFormation template. For more information on creating an 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*.
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.

Managing throughput capacity

You can increase and decrease your file system's throughput capacity to help manage its performance at any time. Throughput capacity is one of the dimensions that determines the speed at which the file server hosting your FSx for Windows File Server file system can serve data. Higher levels of throughput capacity also come with higher levels of I/O operations per second (IOPS) and a larger amount of cache memory on the file server. For more information, see FSx for Windows File Server performance.

Topics

- How throughput scaling works
- Knowing when to modify throughput capacity
- Modifying throughput capacity
- Monitoring throughput capacity updates

How throughput scaling works

When you modify your file system's throughput capacity, Amazon FSx switches out the file system's file server to one with more or less throughput behind the scenes. For Multi-AZ file systems, switching to a new file server triggers an automatic failover and failback while Amazon FSx switches out the preferred and secondary file servers. Single-AZ file systems will be unavailable for a few minutes while the file server is switched during throughput capacity scaling. You are billed for the new amount of throughput capacity once it becomes available to your file system.

Note

During a maintenance operation on the back end, system modifications (including throughput capacity modifications) may be delayed. Maintenance operations can cause system modifications to queue up to be processed.

For Multi-AZ file systems, throughput capacity scaling results in an automatic failover and failback while Amazon FSx switches out the preferred and secondary file servers. During file server replacements, which happen during throughput capacity scaling as well as file system maintenance and an unplanned service disruption, any ongoing traffic to the file system will be served by the remaining file server. When the replaced file server is back online, FSx for Windows will run a resynchronization job to ensure that data is synced back to the newly replaced file server.

FSx for Windows is designed to minimize the impact of this resynchronization activity on application and users. However, the resynchronization process involves synchronizing data in large blocks. This means that a large block of data can require synchronization even if only a small portion is updated. Consequently, the amount of resynchronization depends not only on the amount of data churn, but also the nature of the data churn on the file system. If your workload is write-heavy and IOPS-heavy, the data synchronization process may take longer and require additional performance resources.

Your file system will continue to be available during this time, but in order to reduce the duration of data synchronization, we recommend modifying throughput capacity during idle periods when there is minimal load on your file system. We also recommend ensuring that your file system has sufficient throughput capacity to run the synchronization job in addition to your workload, in order to reduce the duration of data synchronization. Lastly, we recommend testing the impact of failovers while your file system has a lighter load.

Knowing when to modify throughput capacity

Amazon FSx integrates with Amazon CloudWatch, enabling you to monitor your file system's ongoing throughput usage levels. The performance (throughput and IOPS) that you can drive through your file system depends on your specific workload’s characteristics, along with your file system’s throughput capacity, storage capacity, and storage type. You can use CloudWatch metrics to determine which of these dimensions to change to improve performance. For more information, see Monitoring with Amazon CloudWatch.

FSx for Windows File Server provides performance alerts based on values of CloudWatch metrics for your file system in the Monitoring & performance dashboard in the File system details page on the Amazon FSx console. This includes throughput capacity, and other file system metrics that can benefit from throughput capacity increases. For more information, see Performance warnings and recommendations.

Configure your file system with sufficient throughput capacity to meet not only the expected traffic of your workload, but also additional performance resources that are needed to support the features you enable on your file system. For example, if you’re running data deduplication, the throughput capacity that you select must provide enough memory to run deduplication based on the storage that you have. If you’re using shadow copies, increase throughput capacity to a value that's at least three times the value that's expected to be driven by your workload to avoid Windows Server deleting your shadow copies. For more information, see Impact of throughput capacity on performance.

Modifying throughput capacity

You can increase or decrease your file system's throughput capacity using the Amazon FSx console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx API, as described in the following procedures.

To modify a file system's throughput capacity (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to increase the throughput capacity for.
3. For **Actions**, choose **Update throughput**.

   Or, in the **Summary** panel, choose **Update** next to the file system's **Throughput capacity**.

   The **Update throughput capacity** window appears.
4. Choose the new value for **Throughput capacity** from the list.
5. Choose **Update** to initiate the throughput capacity update.

> **Note**
>
> Multi-AZ file systems fail over and fail back when updating throughput scaling, and are fully available. Single-AZ file systems experience a very brief period of unavailability during the update.

6. You can monitor the update progress on the **File systems** detail page, in the **Updates** tab.

   You can monitor the progress of the update by using the Amazon FSx console, the AWS CLI, and the API. For more information, see Monitoring throughput capacity updates.

To modify a file system's throughput capacity (CLI)

To increase or decrease a file system's throughput capacity, use the AWS CLI command update-file-system. Set the following parameters:

- `--file-system-id` to the ID of the file system that you are updating.
- `ThroughputCapacity` to the desired value; valid values are 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4608, 6144, 9216, 12288 MBps.

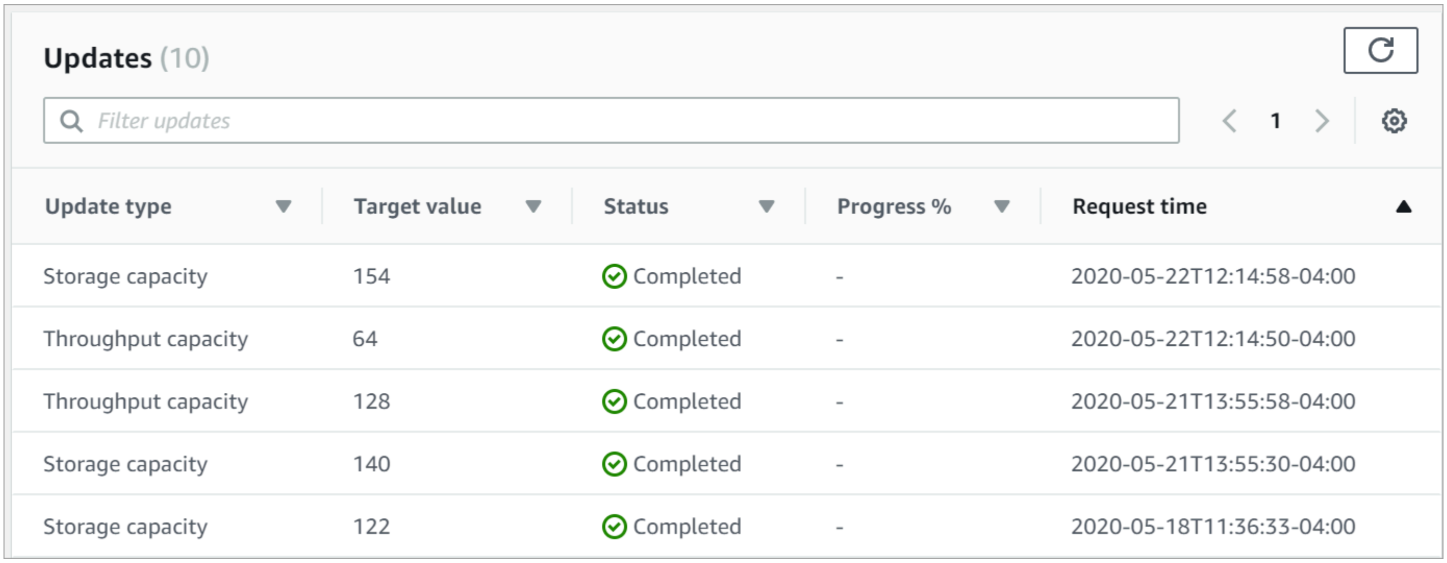
You can monitor the progress of the update by using the Amazon FSx console, the AWS CLI, and the API. For more information, see Monitoring throughput capacity updates.

Monitoring throughput capacity updates

You can monitor the progress of a throughput capacity modification using the Amazon FSx console, the API, and the AWS CLI.

Monitoring throughput capacity changes in the console

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



Updates (10)

Update type	Target value	Status	Progress %	Request time
Storage capacity	154	✓ Completed	-	2020-05-22T12:14:58-04:00
Throughput capacity	64	✓ Completed	-	2020-05-22T12:14:50-04:00
Throughput capacity	128	✓ Completed	-	2020-05-21T13:55:58-04:00
Storage capacity	140	✓ Completed	-	2020-05-21T13:55:30-04:00
Storage capacity	122	✓ Completed	-	2020-05-18T11:36:33-04:00

For throughput capacity update actions, you can view the following information.

Update type

Possible value is **Throughput capacity**.

Target value

The desired value to change the file system's throughput capacity to.

Status

The current status of the update. For throughput capacity updates, the possible values are as follows:

- **Pending** – Amazon FSx has received the update request, but has not started processing it.
- **In progress** – Amazon FSx is processing the update request.
- **Updated optimizing** – Amazon FSx has updated the file system's network I/O, CPU, and memory resources. The new disk I/O performance level is available for write operations. Your read operations will see disk I/O performance between the previous level and the new level until your file system is no longer in the this state.
- **Completed** – The throughput capacity update completed successfully.
- **Failed** – The throughput capacity update failed. Choose the question mark (**?**) to see details on why the throughput update failed.

Request time

The time that Amazon FSx received the update request.

Monitoring changes with the AWS CLI and API

You can view and monitor file system throughput capacity modification requests using the describe-file-systems CLI command and the DescribeFileSystems API action. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you modify a file system's throughput capacity, a `FILE_SYSTEM_UPDATE` administrative action is generated.

The following example shows the response excerpt of a `describe-file-systems` CLI command. The file system has a throughput capacity of 8 MBps, and the target throughput capacity of 256 MBps.

```
.
.
.
    "ThroughputCapacity": 8,
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1581694764.757,
        "Status": "PENDING",
        "TargetFileSystemValues": {
          "WindowsConfiguration": {
            "ThroughputCapacity": 256
          }
        }
    }
]
```

When Amazon FSx completes processing the action successfully, the status changes to `COMPLETED`. The new throughput capacity is then available to the file system, and shows in the `ThroughputCapacity` property. This is shown in the following response excerpt of a **describe-file-systems** CLI command.

```
.
.
.
    "ThroughputCapacity": 256,
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1581694764.757,
        "Status": "COMPLETED",
        "TargetFileSystemValues": {
          "WindowsConfiguration": {
            "ThroughputCapacity": 256
          }
        }
    }
]
```

If the throughput capacity modification fails, the status changes to `FAILED`, and the `FailureDetails` property provides information about the failure. For information about troubleshooting failed actions, see Storage or throughput capacity updates fail.

Managing network type

When you create an FSx for Windows file system, you must specify a network type, which must be one of the following options:

- `IPv4` allows your file system to communicate using only Internet Protocol version 4 (IPv4).
- `Dual-stack` allows your file system to communicate using both Internet Protocol version 6 (IPv6) and IPv4.

You can change the network type of an existing FSx for Windows file system at any time using the Amazon FSx Management Console, AWS CLI, AWS API, or one of the AWS SDKs. For example, if your subnets support both IPv4 and IPv6 addressing, you can update your existing file system from IPv4-only to dual-stack mode, You can also update your dual-stack file system to IPv4-only.

Using dual-stack mode

You should use dual-stack mode if you need to access and manage your Amazon FSx file systems natively from IPv6 clients. By configuring your Amazon FSx file system to use dual-stack addressing, you can access your file data from IPv6 clients, as well as IPv4 clients, in the same Amazon VPC, in another AWS account's VPC, or in your on-premises network. For example, with an Amazon FSx file system configured to use dual-stack, you can have existing IPv4 clients and new IPv6 clients accessing your file data stored on your file system.

By default, Amazon FSx and Amazon VPC use the IPv4 addressing protocol. So as a prerequisite to using IPv6, you must first assign an Amazon-provided IPv6 Classless Inter-Domain Range (CIDR) block to your VPC and subnets before you can use IPv6 with your Amazon FSx file systems. For information on enabling IPv6 for your VPC, see Add IPv6 support for your VPC in the *Amazon Virtual Private Cloud User Guide*.

Changing network type

You can modify a file system's network type using the Amazon FSx console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx API.

To change a file system's network type (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the FSx for Windows file system that you want to change the network type for.
3. For **Actions**, choose **Update network type**. Or, in the **Network & security** panel, choose **Manage** next to the file system's **Network type**.

   The **Update network type** window appears.
4. For **Desired network type**, choose either **IPv4** or **Dual-stack**.

   - If you choose `IPv4`, no further configuration is required.
   - If you choose `Dual-stack`, specify the IPv6 address range that your file system endpoints will use:
     - **Unallocated IPv6 address range from your VPC** – Amazon FSx chooses an available /118 IP address range from one of the VPC's IPv6 CIDR ranges to use as the endpoint IPv6 address range for the file system.
5. Choose **Update**.

To modify a file system's network type (CLI)

- To modify a file system's network type, use the update-file-system CLI command (or the equivalent UpdateFileSystem API operation), as shown in the following example.

  ```
  aws fsx update-file-system \
      --file-system-id fs-0123456789abcdef0 \
      --network-type DUAL
  ```

Tagging your Amazon FSx resources

To help you manage your file systems and other FSx for Windows File Server resources, you can assign your own metadata to each resource in the form of tags. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Topics

- Tag basics
- Tagging your resources
- Tag restrictions
- Permissions required to tag resources

Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's FSx for Windows File Server file systems that helps you track each instance's owner and stack level.

We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper Tagging Best Practices.

Tags don't have any semantic meaning to Amazon FSx and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

If you're using the FSx for Windows File Server API, the AWS CLI, or an AWS SDK, you can use the `TagResource` API action to apply tags to existing resources. Additionally, some resource-creating actions enable you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, we roll back the resource creation process. This ensures that resources are either created with tags or not created at all, and that no resources are left untagged at any time. By tagging resources at the time of creation, you can eliminate the need to run custom tagging scripts after resource creation. For more information about enabling users to tag resources on creation, see Grant permission to tag resources during creation.

Tagging your resources

You can tag FSx for Windows File Server resources that exist in your account. If you're using the Amazon FSx console, you can apply tags to resources by using the Tags tab on the relevant resource screen. When you create resources, you can apply the Name key with a value, and you can apply tags of your choice when creating a new file system. The console may organize resources according to the Name tag, but this tag doesn't have any semantic meaning to the FSx for Windows File Server service.

You can apply tag-based resource-level permissions in your IAM policies to the FSx for Windows File Server API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `TagResource` and `UntagResource` FSx for Windows File Server API actions in your IAM policies to control which tag keys and values are set on your existing resources.

For more information about tagging your resources for billing, see Using cost allocation tags in the *AWS Billing User Guide*.

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8

- Maximum value length – 256 Unicode characters in UTF-8
- The allowed characters for FSx for Windows File Server tags are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.

You can't delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete a file system that you tagged with a tag key called `DeleteMe`, you must use the `DeleteFileSystem` action with the file system resource identifier, such as fs-1234567890abcdef0.

When you tag public or shared resources, the tags you assign are available only to your AWS account; no other AWS account will have access to those tags. For tag-based access control to shared resources, each AWS account must assign its own set of tags to control access to the resource.

Permissions required to tag resources

For more information about the permissions required to tag Amazon FSx resources at creation, see Grant permission to tag resources during creation. For more information about using tags to restrict access to Amazon FSx resources in IAM policies, see Using tags to control access to your Amazon FSx resources.

Update a file system using the AWS CLI

There are three elements that you can update using the procedures in this walkthrough. All other elements of your file system that you can update, you can do so from the console. These procedures assume you have the AWS CLI installed and configured on your local computer. For more information, see Install and Configure in the *AWS Command Line Interface User Guide*.

- **AutomaticBackupRetentionDays** – the number of days that you want to retain automatic backups for your file system.
- **DailyAutomaticBackupStartTime** – the time of the day in Coordinated Universal Time (UTC) that you want the daily automatic backup window to start. The window is 30 minutes starting from this specified time. This window can't overlap with the weekly maintenance backup window.

- **WeeklyMaintenanceStartTime** – the time of the week that you want the maintenance window to start. Day 1 is Monday, 2 is Tuesday, and so on. The window is 30 minutes starting from this specified time. This window can't overlap with the daily automatic backup window.

The following procedures outlines how to update your file system with the AWS CLI.

To update how long automatic backups are retained for your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the number of days that you want to retain your automatic backups for.

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

To update the daily backup window of your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the time with when you want to begin the window.

```
aws fsx update-file-system --file-system-id fs-0123456789abcdef0 --windows-
configuration DailyAutomaticBackupStartTime=01:00
```

To update the weekly maintenance window of your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the date and time with when you want to begin the window.

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

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

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

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

Topics

- Protecting your data with backups
- Protecting your data with shadow copies
- Scheduled replication using AWS DataSync

Protecting your data with backups

You can protect the data on your FSx for Windows File Server file system by taking regular file system backups. Amazon FSx provides you with multiple options for backing up your file systems. You can use automatic daily backups to take a backup everyday. You can take a user-initiated backup of your file system at any time. You can also use AWS Backup as part of a centralized backup solution for your AWS resources. These backup solutions can help you to meet your data retention, business, and compliance needs.

We recommend using the automatic daily backups that are enabled by default for your file system, and using AWS Backup for a centralized backup solution across AWS services. AWS Backup enables you to configure additional backup plans with different frequencies (for example, multiple times a day, daily, or weekly) and retention periods.

With Amazon FSx, backups are file-system-consistent, highly durable, and incremental. Each backup contains all of the information that is necessary to create a new file system, effectively restoring a point-in-time snapshot of the file system. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in Microsoft Windows. To ensure high durability, Amazon FSx stores backups in Amazon Simple Storage Service (Amazon S3).

Amazon FSx backups are incremental, whether they are generated using the automatic daily backup or the user-initiated backup feature. This means that only the data on the file system that has changed after your most recent backup is saved. This minimizes the time required to create the backup and saves on storage costs by not duplicating data.

At some point during the backup process, storage I/O may be suspended briefly, typically for a few seconds. Because the VSS service needs to flush any cached writes to disk before resuming I/O, the duration of the pause may be longer if your workload has a large amount of write operations per second (`DataWriteOperations`). Most end users and applications will experience this I/O suspension as a brief I/O pause. Your applications may have different sensitivity to timeout settings depending on how they are configured.

Creating regular backups for your file system is a best practice that complements the replication that Amazon FSx for Windows File Server performs for your file system. Amazon FSx backups help support your backup retention and compliance needs. Working with Amazon FSx backups is easy, whether it's creating backups, copying a backup, restoring a file system from a backup, or deleting a backup. Note that in order to view usage for a single file system backup, you will need to enable tags for that specific backup and enable tag-based billing reporting.

Topics

- Working with automatic daily backups
- Working with user-initiated backups
- Using AWS Backup with Amazon FSx
- Copying backups
- Restoring backups to new file system
- Creating user-initiated backups
- Deleting backups
- Size of backups
- Copying backups within the same account
- Restoring a backup to a new file system

Working with automatic daily backups

By default, Amazon FSx takes an automatic daily backup of your file system. These automatic daily backups occur during the daily backup window that was established when you created the file system. When you choose your daily backup window, we recommend that you choose a convenient time of the day that is outside of the normal operating hours for the applications that use the file system. We also recommend choosing a backup window outside of the maintenance window because automated backups may not occur if there is ongoing file system maintenance.

Automatic daily backups are kept for a certain period of time, known as a retention period. When you create a file system in the Amazon FSx console, the default automatic daily backup retention period is 30 days. The default retention period is different in the Amazon FSx API and CLI. You can set the retention period to be between 0–90 days. Setting the retention period to 0 (zero) days turns off automatic daily backups. Automatic daily backups are deleted when the file system is deleted.

Note

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

You can use the AWS CLI or one of the AWS SDKs to change the backup window and backup retention period for your file systems. Use the `UpdateFileSystem` API operation or the `update-file-system` CLI command. For more information, see Update a file system using the AWS CLI.

Important

Lowering the retention period for automatic daily backups will result in the permanent deletion of backups outside of the new retention window. Ensure that you no longer need these older backups before proceeding.

Working with user-initiated backups

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

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

To learn how to take user-initiated backups of your file systems, see Creating user-initiated backups.

Using AWS Backup with Amazon FSx

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

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

AWS Backup uses the built-in backup functionality of Amazon FSx. Backups taken from the AWS Backup console have the same level of file system consistency and performance, and the same restore options as backups taken through the Amazon FSx console. Backups taken from AWS Backup are incremental relative to any other Amazon FSx backups you take, either user-initiated or automatic.

If you use AWS Backup to manage these backups, you gain additional functionality, such as unlimited retention options and the ability to create scheduled backups as frequently as every hour. In addition, AWS Backup retains your immutable backups even after the source file system is deleted. This protects against accidental or malicious deletion.

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

Copying backups

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

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

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

Cross-account backup copies are valuable for meeting regulatory compliance requirements to copy backups to an isolated account. They also provide an additional layer of data protection to help prevent accidental or malicious deletion of backups, loss of credentials, or compromise of AWS KMS keys. Cross-account backups support *fan-in* (copy backups from multiple primary accounts to one isolated backup copy account) and *fan-out* (copy backups from one primary account to multiple isolated backup copy accounts).

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

Backup copy limitations

The following are some limitations when you copy backups:

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

Permissions for cross-Region backup copies

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

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

JSON

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

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

Full and incremental copies

When you copy a backup to a different destination AWS Region or destination AWS account from the source backup, the first copy is a full backup copy, even if you use the same KMS key to encrypt both source and destination copies of the backup.

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

To learn how to copy backups of your file systems, see Copying backups within the same account.

Restoring backups to new file system

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.

To ensure that users can continue to access the restored file system, make sure that the Active Directory domain associated with the restored file system is the same as that of the original file system, or is trusted by the Active Directory domain of the original file system. For more information about Active Directory, see Working with Microsoft Active Directory.

To learn how to restore a backup to a new FSx for Windows file system, see Restoring a backup to a new file system.

Note

You can only restore a file system backup to a new file system with the same deployment type and storage capacity as the original. You can increase the new file system's storage capacity after it becomes available. For more information, see Managing storage capacity.

You can change any of the following file system settings when restoring a backup to a new file system:

- Storage type
- Throughput capacity
- VPC
- Availability Zone
- Subnet
- VPC security groups
- Active Directory Configuration
- AWS KMS encryption key
- Daily automatic backup start time
- Weekly maintenance window

Creating user-initiated backups

In addition to automatic daily file system backups, you can create a user-initiated file system backup at anytime, using the Amazon FSx console as described in the following procedure.

To create a user-initiated file system backup

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

You have now created your file system backup. You can find a table of all your backups in the Amazon FSx console by choosing **Backups** in the left side navigation. Your new user-initiated backup has the type `USER_INITIATED`, and its status is `CREATING` until it becomes `AVAILABLE`. For more information, see Working with user-initiated backups.

Deleting backups

You can delete any user-initiated and automatic daily backups of your file system using the Amazon FSx console, CLI, or API, described in the following procedures. For deleting backups taken by AWS Backup, which have type of **AWS Backup**, you must use the the AWS Backup console, CLI, or API. 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 (console)

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

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

Size of backups

Backups size is determined using the used storage in the file system, rather than the total provisioned storage capacity. The size of your backups will depend on the used storage capacity as well as the amount of data churn on your file system. Depending on how your data is distributed across the file system’s storage volumes and how often it changes, your total backup usage may be greater or less than your used storage capacity. When you delete a backup, only the data unique to that backup is removed.

In order to provide backups that are file-system-consistent, durable, and incremental, Amazon FSx backs up data at the block level. The data on the file system's storage volumes may be stored across multiple blocks depending on the pattern that they were written or over-written in. As a result, the total size of backup usage may not match the exact size of the files and directories on the file system. Your overall backup usage and cost can be found in the AWS Billing Dashboard or AWS Cost Management Console.

Use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of combined resources, organize your billing information according to resources with the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Using Cost Allocation Tags in the *AWS Billing User Guide*.

> **Note**
>
> When you increase storage capacity, the process of migrating data from the old set of storage disks to the new, larger set of storage disks can result in a temporary increase in backup usage until backups associated with the old set of storage disks are deleted. If your file system's storage was only partially used before you increase storage capacity, the size of data that needs to be migrated to the new disks may be larger than the size of data that exists on the original storage disks. This may cause an increase in backup usage up to the new storage capacity level. You should consider the impact of increasing storage capacity on your backup planning.

Copying backups within the same account

You can use the AWS Management Console and AWS CLI to manually copy backups within the same AWS account to another AWS Region (cross-Region copies) or within the same AWS Region (in-Region copies) using the following procedures.

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

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the navigation pane, choose **Backups**.
3. In the **Backups** table, choose the backup that you want to copy, and then choose **Copy backup**.
4. In the **Settings** section, do the following:

   - In the **Destination Region** list, choose a destination AWS Region to copy the backup to. The destination can be in another AWS Region (cross-Region copy) or within the same AWS Region (in-Region copy).
   - (Optional) Select **Copy Tags** to copy tags from the source backup to the destination backup. If you select **Copy Tags** and also add tags at step 6, all the tags are merged.

5. For **Encryption**, choose the AWS KMS encryption key to encrypt the copied backup.
6. For **Tags - optional**, enter a key and value to add tags for your copied backup. If you add tags here and also selected **Copy Tags** at step 4, all the tags are merged.
7. Choose **Copy backup**.

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

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

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

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

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

  The response shows the description of the copied backup.

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

Restoring a backup to a new file system

You can restore a file system backup to create new file system using the AWS Management Console, CLI, and API, as described in the following procedure.

To restore a file system from a backup

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

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

- The file system deployment type is **Multi-AZ** or **Single-AZ 2**.
- The storage capacity is at least 2,000 GiB.

4. Complete the wizard as you do when you create a new file system.
5. Choose **Review and create**.
6. Review the settings you chose for your Amazon FSx file system, and then choose **Create file system**.

   Amazon FSx is creating a new file system, and once its status changes to `AVAILABLE`, you can use the file system as normal.

Protecting your data 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, users can quickly recover deleted or changed files that are stored on the network, and compare file versions. Storage administrators can easily schedule shadow copies to be taken periodically using Windows PowerShell commands.

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

> **Note**
>
> Shadow copies are *not* enabled on FSx for Windows File Server by default. To protect the data on your file system using shadow copies, you must enable shadow copies and set up a shadow copy schedule on your file system. For more information, see Configuring shadow copies to use the default storage and schedule.

⚠ **Warning**

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

Topics

- Best practices when using shadow copies
- Setting up shadow copies
- Configuring shadow copies to use the default storage and schedule
- Setting the maximum amount of shadow copy storage
- Viewing shadow copy storage
- Creating a custom shadow copy schedule
- Viewing the shadow copy schedule
- Creating a shadow copy
- Viewing existing shadow copies
- Deleting shadow copies
- Deleting a shadow copy schedule
- Deleting shadow copy storage, schedule, and all shadow copies
- Troubleshooting shadow copies

Best practices when using shadow copies

You can enable shadow copies for your file system to allow end-users to view and restore individual files or folders from an earlier snapshot in Windows File Explorer. Amazon FSx uses the shadow copies feature as provided by Microsoft Windows Server. Use these best practices for shadow copies:

- **Ensure your file system has sufficient performance resources**: Microsoft Windows uses a copy-on-write method to record changes since the most recent shadow copy point, and this copy-on-write activity can result in up to three I/O operations for every file write operation.
- **Use SSD storage and increase throughput capacity**: Because Windows requires a high level of I/O performance to maintain shadow copies, we recommend using SSD storage and increasing throughput capacity up to a value as high as three times that of your expected workload. This

helps to ensure that your file system has enough resources to avoid issues like the unwanted deletion of shadow copies.

- **Maintain only the number of shadow copies that you need**: If you have a large number of shadow copies—for example, more than 64 of the most recent shadow copies—or shadow copies that occupy a large amount of storage (TB-scale) on a single file system, processes such as failover and failback might take some extra time. This is due to the need for FSx for Windows to run consistency checks on the shadow copy storage. You might also experience higher latency of I/O operations due to the need for FSx for Windows to perform copy-on-write activity while maintaining the shadow copies. To minimize availability and performance impact from shadow copies, delete unused shadow copies manually or configure scripts to delete old shadow copies on your file system automatically.

> **Note**
>
> During failover events for Multi-AZ file systems, FSx for Windows runs a consistency check that requires scanning the shadow copy storage on your file system before the new active file server comes online. The duration of the consistency check is related to the number of shadow copies on your file system as well as the storage consumed. To prevent delayed failover and failback events, we recommend maintaining fewer than 64 shadow copies on your file system and following the steps below to regularly monitor and delete your oldest shadow copies.

Setting up shadow copies

You enable and schedule periodic shadow copies on your file system using Windows PowerShell commands defined by Amazon FSx. The following are three main settings when configuring shadow copies on your FSx for Windows File Server file system:

- Setting the maximum amount of storage that shadow copies can consume on your file system
- (Optional) Setting the maximum number of shadow copies that can be stored on your file system. The default value is 20.
- (Optional) Setting a schedule that defines the times and intervals at which to take shadow copies, such as daily, weekly, and monthly

You can store a maximum of 500 shadow copies per file system at any point in time; however, we recommend maintaining fewer than 64 shadow copies at any time to ensure availability and performance. When you reach this limit, the next shadow copy that you take replaces the oldest shadow copy. Similarly, when the maximum shadow copy storage amount is reached, one or more of the oldest shadow copies are deleted to make sufficient storage space for the next shadow copy.

For information about how to quickly enable and schedule periodic shadow copies by using default Amazon FSx settings, see Configuring shadow copies to use the default storage and schedule.

Considerations for allocating shadow copy storage

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

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

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

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

Considerations when setting the maximum number of shadow copies

When you enable shadow copies, you can specify the maximum number of shadow copies stored on the file system. The default limit is 20, and to minimize availability and performance impact from shadow copies, Microsoft recommends configuring the maximum number of shadow copies to less than 64. Because Windows requires a high level of I/O performance to maintain shadow copies, we recommend using SSD storage and increasing throughput capacity up to a value as high as three times that of your expected workload. This helps to ensure that your file system has enough resources to avoid issues like the unwanted deletion of shadow copies.

You can set the maximum number of shadow copies up to 500. However, if you have a large number of shadow copies or shadow copies that occupy a large amount of storage (TB-scale) on a single file system, processes such as failover and failback may take longer than expected. This is because Windows needs to run consistency checks on the shadow copy storage. You may also experience higher latency of I/O operations due to the need for Windows to perform copy-on-write activity while maintaining the shadow copies.

File system recommendations for shadow copies

Following are file system recommendations for using shadow copies.

- Make sure you provision sufficient performance capacity for your workload needs on your file system. Amazon FSx delivers the Shadow Copies feature as provided by Microsoft Windows Server. By design, Microsoft Windows uses a copy-on-write method for recording the changes since the most recent shadow copy point, and this copy-on-write activity can result in up to three I/O operations for every file write operation. If Windows is unable to keep up with the incoming rate of I/O operations per second, it can cause all shadow copies to be deleted because it can no longer maintain the shadow copies via copy-on-write. Therefore, it is important that you provision sufficient I/O performance capacity for your workload needs on your file system (both the throughput capacity dimension that determines the file server I/O performance, and the storage type and capacity that determine the storage I/O performance).
- We generally recommend that you use file systems configured with SSD storage rather than HDD storage when you enable shadow copies, given that Windows consumes a higher I/O performance to maintain shadow copies, and given that HDD storage provides lower performance capacity for I/O operations.
- Your file system should have at least 320 MB of free space, in addition to the maximum shadow copy storage amount configured (`MaxSpace`). For example, if you allocated 5 GB `MaxSpace` to shadow copies, your file system should always have at least 320 MB free space in addition to the 5 GB `MaxSpace`.

⚠ **Warning**

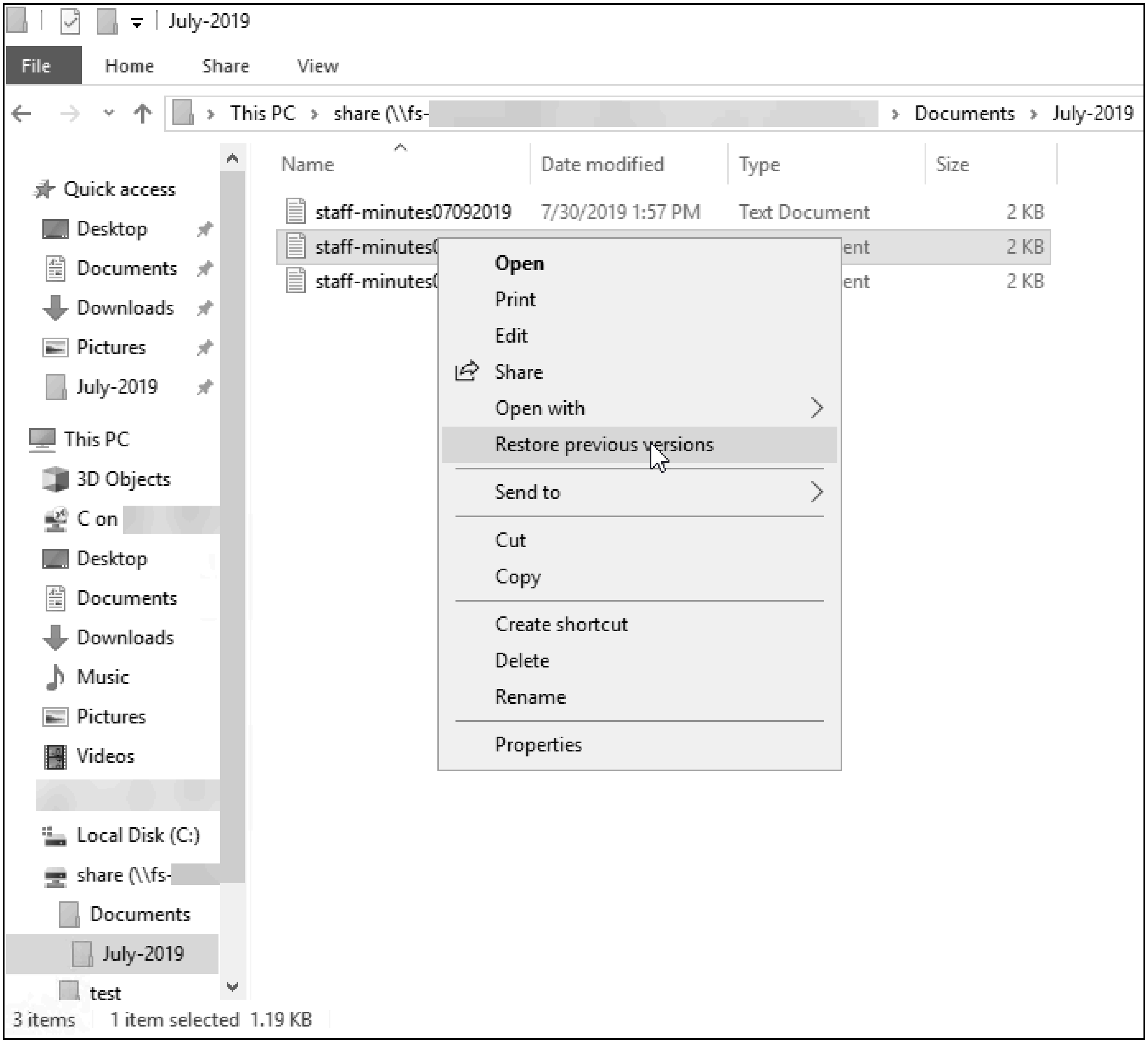
When configuring your shadow copy schedule, make sure that you don't schedule shadow copies when migrating data or when data deduplication jobs are scheduled to run.

You should schedule shadow copies when you expect your file system to be idle. For information about configuring a custom shadow copy schedule, see Creating a custom shadow copy schedule.

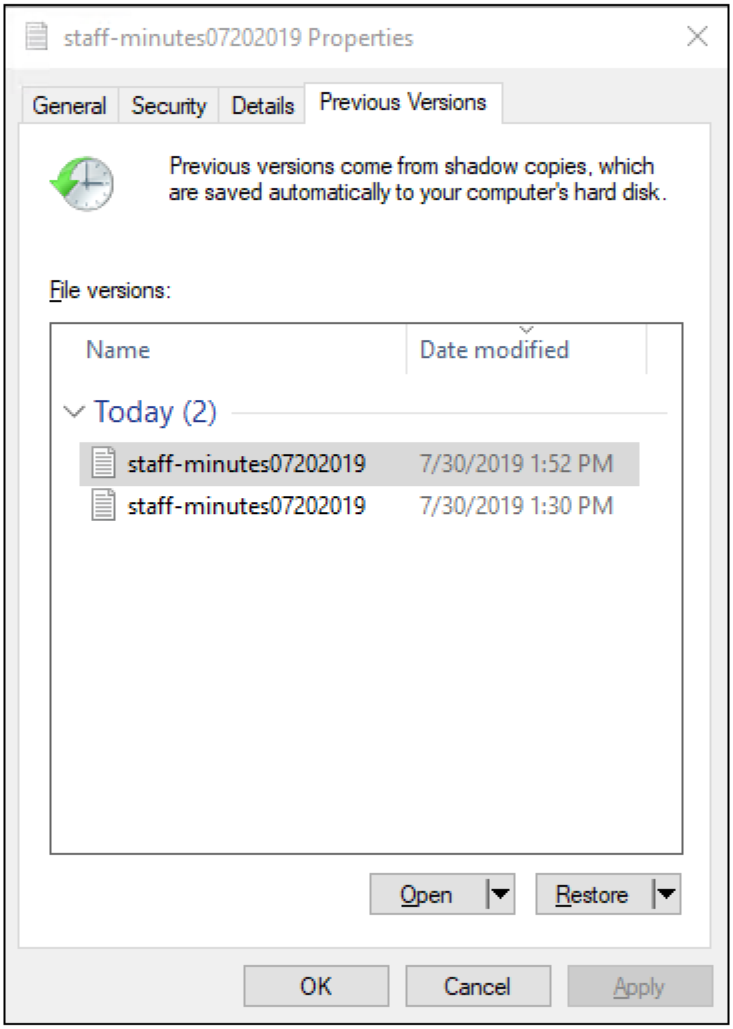
Restoring individual files and folders

After you configure shadow copies on your Amazon FSx file system, your users can quickly restore previous versions of individual files or folders, and recover deleted files.

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.



staff-minutes07202019 Properties

Previous versions come from shadow copies, which are saved automatically to your computer's hard disk.

File versions:

Name	Date modified
Today (2)	
staff-minutes07202019	7/30/2019 1:52 PM
staff-minutes07202019	7/30/2019 1:30 PM

Configuring shadow copies to use the default storage and schedule

You can quickly set up shadow copies on your file system by using the default shadow copy storage setting and schedule. The default shadow copy storage setting lets shadow copies consume a maximum of 10 percent of your file system storage capacity. If you increase your file system's storage capacity, the amount of the currently allocated shadow copy storage is not similarly increased.

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

To set 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*.

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 MaxShadowCopyNumber
   -------------- ---------    -------- -------------------
                0         0 10737418240                  20
   ```

To set the default shadow copy schedule

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*.

3. Set the default shadow copy schedule by using the following command.

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

The response displays the default schedule that is now set.

```
FSx Shadow Copy Schedule

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

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

Setting the maximum amount of shadow copy storage

You define the maximum amount of storage that shadow copies can consume on a file system using the `Set-FsxShadowStorage` custom PowerShell command. You can specify the maximum size that shadow copies can grow to by using either the `-Maxsize` or the `-Default` parameters. Using `Default` sets the maximum to 10% of the file system's storage capacity. You cannot specify the `-Maxsize` and `-Default` parameters in the same command.

Using `-Maxsize`, you can define shadow copy storage as follows:

- In bytes: `Set-FsxShadowStorage -Maxsize 2500000000`
- In kilobytes, megabytes, gigabytes, or other units: `Set-FsxShadowStorage -Maxsize (2500MB)` or `Set-FsxShadowStorage -Maxsize (2.5GB)`
- As a percentage of the overall storage: `Set-FsxShadowStorage -Maxsize "20%"`
- As unbounded: `Set-FsxShadowStorage -Maxsize "UNBOUNDED"`

Use `-Default` to set shadow storage to use up to 10 percent of the file system: `Set-FsxShadowStorage -Default`. To learn more about using the default option, see Configuring shadow copies to use the default storage and schedule.

To set the amount of shadow copy storage on an FSx for Windows File Server file system

1. Connect to a compute instance that has network connectivity with your file system as a user that is a member of the file system administrators group. In AWS Managed Microsoft AD, that group is **AWS Delegated FSx Administrators**. In your self-managed Microsoft AD, that

group is **Domain Admins** or the custom group that you specified for administration when you created your file system. For more information, see Connecting to Your Windows Instance in the *Amazon EC2 User Guide*.

2. Open a Windows PowerShell window on the compute instance.
3. Use the following command to open a remote PowerShell session on your Amazon FSx file system. Replace *`FSxFileSystem-Remote-PowerShell-Endpoint`* with the Windows Remote PowerShell endpoint of file system that you want to administer. You can find the Windows Remote PowerShell endpoint in the Amazon FSx console, in the **Network & Security** section of the file system details screen, or in the response of the `DescribeFileSystem` API operation.

```
PS C:\Users\delegateadmin> enter-pssession -computername FSxFileSystem-Remote-
PowerShell-Endpoint -configurationname fsxremoteadmin
```

4. Verify that shadow copy storage is not already configured on the file system using the following command.

```
[fs-1234567890abcef12]: PS>Get-FsxShadowStorage
No Fsx Shadow Storage Configured
```

5. Set the amount of shadow storage to 10 percent of the volume and the maximum number of shadow copes to 20 using the `-Default` option.

```
[fs-1234567890abcef12]: PS>Set-FsxShadowStorage -Default
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace    MaxSpace MaxShadowCopyNumber
-------------- ---------    -------- -------------------
             0         0 32530536858                  20
```

You can limit the maximum number of shadow copies allowed on your file system by using the `Set-FSxShadowStorage` command with the `-MaxShadowCopyNumber` parameter and specifying a value from 1-500. By default, the maximum number of shadow copies is set to 20, as recommended for by Microsoft for active workloads.

Viewing shadow copy storage

You can view the amount of storage currently consumed by shadow copies on your file system using the `Get-FsxShadowStorage` command in a remote PowerShell session on your file system. For instructions on launching a remote PowerShell session on your file system, see Using the Amazon FSx CLI for PowerShell.

```
[fs-1234567890abcef12]: PS>PS>Get-fsxshadowstorage
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace    MaxSpace MaxShadowCopyNumber
-------------- ---------    -------- -------------------
             0         0 10737418240                  20
```

The output shows the shadow storage configuration, as follows:

- `AllocatedSpace` – The amount of storage on the file system in bytes currently allocated to shadow copies. Initially, this value is 0.
- `UsedSpace` – The amount of storage, in bytes, currently used by shadow copies. Initially, this value is 0.
- `MaxSpace` – The maximum amount of storage, in bytes, to which shadow storage can grow. This is the value that you set for shadow copy storage using the `Set-FsxShadowStorage` command.
- `MaxShadowCopyNumber` – The maximum number of shadow copies that the file system can have, from 1-500.

When the `UsedSpace` amount reaches the maximum shadow copy storage amount configured (`MaxSpace`) or the number of shadow copies reaches the maximum shadow copy number configured (`MaxShadowCopyNumber`), the next shadow copy that you take replaces the oldest shadow copy. If you don't want to lose your oldest shadow copies, monitor your shadow copy storage to make sure that you have sufficient storage space for new shadow copies. If you need more space, you can delete existing shadow copies or increase the maximum amount of shadow copy storage.

> **Note**
>
> When shadow copies are automatically or manually created, they use the amount of shadow copy storage that you configured as a storage limit. Shadow copies grow in size over time and utilize the available storage space shown by the CloudWatch

FreeStorageCapacity metric up to the maximum shadow copy storage amount configured (MaxSpace).

Creating a custom shadow copy schedule

Shadow copy schedules use scheduled task triggers in Microsoft Windows to specify when shadow copies are automatically taken. A shadow copy schedule can have multiple triggers, providing you with a lot of scheduling flexibility. Only one shadow copy schedule can exist at a time. Before you can create a shadow copy schedule, you must first set the amount of shadow copy storage.

When you run the Set-FsxShadowCopySchedule command on a file system, you overwrite any existing shadow copy schedule. If your client computer is in the UTC time zone, you can also 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 Configuring shadow copies to use the default storage and schedule.

To create a custom shadow copy schedule

1. Create a set of Windows scheduled task triggers to define when shadow copies are taken in the shadow copy schedule. Use the new-scheduledTaskTrigger command in a PowerShell on your local machine to set multiple triggers.

   This following example creates a custom shadow copy schedule that takes shadow copies every Monday–Friday, at 6:00 AM and at 6:00 PM UTC. By default, times are in UTC, unless you specify a time zone in the Windows scheduled task triggers you create.

   ```
   PS C:\Users\delegateadmin> $trigger1 = new-scheduledTaskTrigger -weekly -DaysOfWeek
    Monday,Tuesday,Wednesday,Thursday,Friday -at 06:00
   PS C:\Users\delegateadmin> $trigger2 = new-scheduledTaskTrigger -weekly -DaysOfWeek
    Monday,Tuesday,Wednesday,Thursday,Friday -at 18:00
   ```

2. Use invoke-command to run the scriptblock command. Doing so writes a script that sets the shadow copy schedule with the new-scheduledTaskTrigger value that you just created. Replace *FSxFileSystem-Remote-PowerShell-Endpoint* with the Windows

Remote PowerShell endpoint of file system that you want to administer. You can find the Windows Remote PowerShell endpoint in the Amazon FSx console, in the **Network & Security** section of the file system details screen, or in the response of the `DescribeFileSystem` API operation.

```
PS C:\Users\delegateadmin> invoke-command -ComputerName FSxFileSystem-Remote-
PowerShell-Endpoint -ConfigurationName FSxRemoteAdmin -scriptblock {
```

3. Enter the following line at the >> prompt to set your shadow copy schedule using the `set-fsxshadowcopyschedule` command.

```
>> set-fsxshadowcopyschedule -scheduledtasktriggers $Using:trigger1,$Using:trigger2
 -Confirm:$false }
```

The response displays the shadow copy schedule that you configured on the file system.

```
FSx Shadow Copy Schedule



Start Time:     : 2019-07-16T06:00:00+00:00
Days of Week    : Monday,Tuesday,Wednesday,Thursday,Friday
WeeksInterval   : 1
PSComputerName  : fs-0123456789abcdef1
RunspaceId      : 12345678-90ab-cdef-1234-567890abcde1

Start Time:     : 2019-07-16T18:00:00+00:00
Days of Week    : Monday,Tuesday,Wednesday,Thursday,Friday
WeeksInterval   : 1
PSComputerName  : fs-0123456789abcdef1
RunspaceId      : 12345678-90ab-cdef-1234-567890abcdef
```

Viewing the 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 Using the Amazon FSx CLI for PowerShell.

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

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

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 Using the Amazon FSx CLI for PowerShell.

```
[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 Using the Amazon FSx CLI for PowerShell.

```
[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 Using the Amazon FSx CLI for PowerShell.

Specify which shadow copies to delete by using one of the following required options:

- `-Oldest` deletes the oldest shadow copy

- `-All` deletes all existing shadow copies
- `-ShadowCopyId` deletes a specific shadow copy by ID.

You can use only one option with the command. An error occurs if you don't specify which shadow copy to delete, if you specify multiple shadow copy IDs, or if you specify an invalid shadow copy ID.

To delete the oldest shadow copy on your file system, enter the following command in a remote PowerShell session on your file system.

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopies -Oldest
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-FSxShadowCopies" on target "Removing oldest shadow
 copy".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y": Y
Shadow Copy {ABCDEF12-3456-7890-ABCD-EF1234567890} deleted
```

To delete a specific shadow copy on your file system, enter the following command in a remote PowerShell session on your file system.

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopies -ShadowCopyId "{ABCDEF12-3456-7890-
ABCD-EF1234567890}"
Are you sure you want to perform this action?
Performing the operation "Remove-FSxShadowCopies" on target "Removing shadow copy
 {ABCDEF12-3456-7890-ABCD-EF1234567890}".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y":>Y
Shadow Copy \\AMZNFSXABCDE123\root\cimv2:Wind32_ShadowCopy.ID{ABCDEF12-3456-7890-ABCD-
EF1234567890}".ID deleted.
```

To delete a certain number of the oldest shadow copies on your file system, update your `-MaxShadowCopyNumber` parameter to the desired number of shadow copies that you would like to have remaining. However, this change will only take effect after the next shadow copy snapshot is taken, when the system will automatically delete the excess shadow copies. Use the following command in a remote PowerShell session on your file system.

```
[fs-1234567890abcef12]: PS>Get-fsxshadowstorage
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace MaxSpace    MaxShadowCopyNumber
```

```
-------------- --------- ----------- -------------------
     556679168  21659648 10737418240                  50

[fs-1234567890abcef12]: PS>Set-FsxShadowStorage  -MaxShadowCopyNumber 5
Validation
You have 50 shadow copies. Older versions of shadow copies will be deleted, keeping 5
 latest shadow copies on your file system.
Do you want to continue?
[Y] Yes  [N] No  [?] Help (default is "N"): y
FSx Shadow Storage Configuration

AllocatedSpace UsedSpace    MaxSpace MaxShadowCopyNumber
-------------- ---------    -------- -------------------
     556679168  21659648 10737418240                   5
```

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 Using the Amazon FSx CLI for PowerShell.

```
[fs-0123456789abcdef1]PS>Remove-FsxShadowCopySchedule

Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-FsxShadowCopySchedule" on target "Removing FSx Shadow
 Copy Schedule".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (Default is "Y"): Y
[fs-0123456789abcdef1]PS>
```

Deleting shadow copy storage, schedule, and all shadow copies

You can delete your shadow copy configuration, including all existing shadow copies and 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 Using the Amazon FSx CLI for PowerShell.

```
[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.
```

Troubleshooting shadow copies

There are a number of potential causes when shadow copies are missing or inaccessible, as described in the following section.

Topics

- Oldest shadow copies are missing
- All of my shadow copies are missing
- Cannot create Amazon FSx backups or access shadow copies on a recently restored or updated file system

Oldest shadow copies are missing

The oldest shadow copies are deleted in either of these situations:

- If you have 500 shadow copies, the next shadow copy replaces the oldest shadow copy, regardless of the remaining allocated storage volume space for shadow copies.
- If the maximum shadow copy storage amount configured is reached, the next shadow copy replaces one or more of the oldest shadow copies, even if you have fewer than 500 shadow copies.

Both results are expected behavior. If you have insufficient storage allocated for shadow copies, consider increasing the storage you have allocated.

All of my shadow copies are missing

Having insufficient I/O performance capacity on your file system (for example, because you're using HDD storage, because the HDD storage has run out of burst capacity, or because the throughput capacity is insufficient) can cause all shadow copies to be deleted by Windows Server because it is unable to maintain the shadow copies with the available I/O performance capacity. Consider the following recommendations to help prevent this problem:

- If you're using HDD storage, use the Amazon FSx console or Amazon FSx API to switch to using SSD storage. For more information, see Managing your file system's storage type.
- Increase the file system's throughput capacity to a value three times your expected workload.
- Make sure that your file system has at least 320 MB of free space, in addition to the maximum shadow copy storage amount configured.
- Schedule shadow copies when you expect your file system to be idle.

For more information, see File system recommendations for shadow copies.

Cannot create Amazon FSx backups or access shadow copies on a recently restored or updated file system

This is expected behavior. Amazon FSx rebuilds the shadow-copy state on a recently restored file system and does not allow access to shadow copies or backups while the rebuilding is still in progress.

Scheduled replication using AWS DataSync

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

Using FSx for Windows File Server with Microsoft SQL Server

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

> **Note**
>
> Currently, Amazon FSx doesn't support the Microsoft SQL Server IFI (Instant File Initialization) feature.

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

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

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

Using Amazon FSx for Active SQL Server Data Files

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

Create a Continuously Available Share

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

Configure SMB timeout settings

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

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

Using Amazon FSx as an SMB File Share Witness

Windows Server Failover cluster deployments commonly deploy an SMB file share witness to maintain quorum of the cluster's resources. Witness file shares require only a small amount of storage for quorum information. Amazon FSx file systems can be used as an SMB file share witness for Windows Server Failover Cluster deployments.

Migrating existing file storage to Amazon FSx

Amazon FSx for Windows File Server has the features, performance, and compatibility to help you easily lift and shift enterprise applications to the Amazon Web Services Cloud. The process to migrate your on-premises Microsoft Windows File Server storage to FSx for Windows File Server has the following four major steps:

1. Migrate your files to FSx for Windows File Server. For more information, see Migrating existing file storage to FSx for Windows File Server.
2. Migrate your file share configuration to FSx for Windows File Server. For more information, see Migrating your on-premises file share configurations to Amazon FSx.
3. Associate your existing DNS name as a DNS alias for your Amazon FSx file system. For more information, see Associating a DNS alias with Amazon FSx.
4. Cut over to FSx for Windows File Server. For more information, see Cutting over operations to Amazon FSx for Windows File Server.

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

Topics

- Migrating existing file storage to FSx for Windows File Server
- Migrating your on-premises file share configurations to Amazon FSx
- Migrating your on-premises DNS configuration to FSx for Windows File Server
- Cutting over operations to Amazon FSx for Windows File Server

Migrating existing file storage to FSx for Windows File Server

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

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

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

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

To move data from an Amazon FSx file system using SSD storage to HDD storage, you can take the following steps. (Note that HDD file systems have a minimum 2TB storage capacity, and you cannot change storage capacity when restoring from a backup.)

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

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

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

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

You can use DataSync to transfer files between two FSx for Windows File Server file systems, and also move data to a file system in a different AWS Region or AWS account. You can use DataSync with FSx for Windows File Server file systems for other tasks. For example, you can perform one-time data migrations, periodically ingest data for distributed workloads, and schedule replication for data protection and recovery.

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

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

Prerequisites

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

If you are performing a large data migration, or a migration involving many small files, we recommend using an Amazon FSx File System with SSD storage type. This is because DataSync tasks involve scans of file metadata which can exhaust the disk IOPS limits of HDD file systems, leading to long-running migrations and file system performance impact. For more information, see: Best practices for migrating existing file storage to FSx for Windows File Server.

If your dataset consists of mostly small files, with file counts in the millions, or if you have more available network bandwidth than a single DataSync task can consume, you can also accelerate your data transfers with scale out architecture. For more information, see: How to accelerate your data transfers with AWS DataSync scale out architectures.

You can monitor the disk I/O utilization of your file system using FSx performance metrics.

Basic steps for migrating files using DataSync

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

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

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

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

Migrating between two Amazon FSx file systems

You can use DataSync to migrate data between two Amazon FSx file systems. This can be helpful if you need to move your workload from an existing file system to a new file system with a different configuration, such as from a Single-AZ to a Multi-AZ configuration. You can also use DataSync to split your workload between two file systems.

Here is a sample overview of the migration process:

1. Create DataSync locations for the source and destination file systems. Note that the source and destination must belong to the same Active Directory (AD) domain, or have an AD trust relationship between their domains.
2. Create and configure a DataSync task to transfer data from the source to the destination. You can run the task as a one-time instance, or set the task to run automatically on a schedule that you configure.
3. After the task completes successfully, the data in your destination file system is an exact copy of your source. Note that you will need to temporarily pause any write activity or file updates on your source file system to complete the task. You can then cut over to your destination file system and delete the source file system.

Before migrating from your production file system, you can test the migration process on a file system that's restored from a recent backup. This enables you to estimate how long the data transfer process takes, and to troubleshoot DataSync errors in advance.

To minimize your cutover time, you can run DataSync tasks in advance, moving the majority of your data from your source file system to your destination file system. After stopping traffic to your source file system, you can run one final task transfer to sync any data that’s been newly updated since you stopped traffic, and then cut over to your destination file system.

You can configure DataSync tasks to only run in certain directories, or to include or exclude certain paths. This can be useful if you’re running multiple tasks in parallel, or if you want to migrate a subset of your data.

You can create a DNS alias on your destination file system that's the same as the DNS name of your source file system. This enables your end-users and applications to continue accessing file data using the DNS name of your source file system. For more information about how to set up a DNS alias, see: Accessing data using DNS aliases.

When performing this type of migration, we recommend the following:

- Schedule your migration to avoid any file system backups, your weekly maintenance window, and `Data Deduplication` jobs. Specifically, we recommend disabling the `Data Deduplication GarbageCollection` job if it coincides with your planned migration.
- Use an SSD storage type for both your source and destination file systems. You can switch between HDD and SSD storage types by restoring from backup. For more information see: Migrating existing file storage to FSx for Windows File Server.
- Configure your source and destination file systems with sufficient throughput capacity for the amount of data that you need to transfer. During DataSync task processes, monitor the performance utilization of both the source and the destination file systems. For more information, see: Monitoring with Amazon CloudWatch.
- Set up DataSync monitoring to help you understand the progress of ongoing tasks. You can also send DataSync logs to the Amazon CloudWatch Logs group to assist you with debugging your tasks if you encounter any errors.

Migrating existing files to FSx for Windows File Server using Robocopy

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

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

Prerequisites for file migration with Robocopy

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

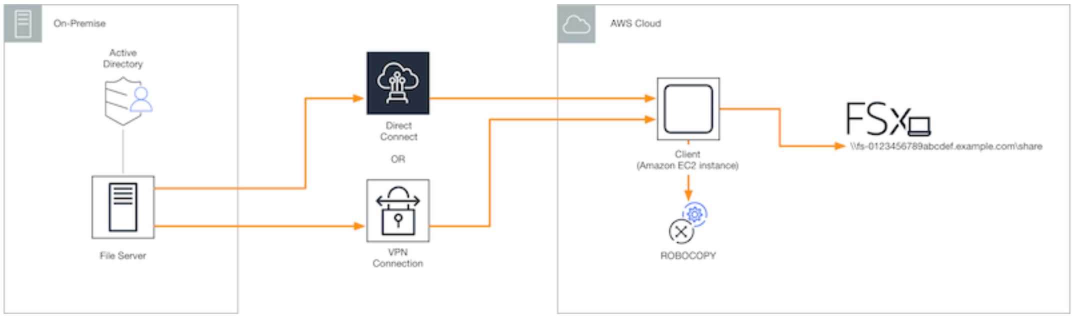
- Establish network connectivity (by using Direct Connect or VPN) between your on-premises Active Directory and the VPC where you want to create the Amazon FSx file system.
- Create a service account on your Active Directory with delegated permissions to join computers to the domain. For more information,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

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

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

Note

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



Migrating files using Robocopy

You can migrate your existing files from your on-premises file systems to FSx for Windows File Server file systems by using the following procedure.

To migrate existing files to Amazon FSx using Robocopy

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

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

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

The command completed successfully.
```

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

   For more information, see the table of source and destination file system accessibility requirements in the Prerequisites for file migration with Robocopy.

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

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

   The command completed successfully.
   ```

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

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

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

   The example command preceding uses the following elements and options:

   - Y – Refers to the source share located in the on-premises Active Directory forest mydata.com.
   - Z – Refers to the target share \\amznfsxabcdef1.mydata.com\share on Amazon FSx.
   - /copy – Specifies the following file properties to be copied:
     - D – data
     - A – attributes

    - T – timestamps
    - S – NTFS ACLs
    - O – owner information
    - U – auditing information.
- /secfix – Fixes file security on all files, even skipped ones.
- /e – Copies subdirectories, including empty ones.
- /b – Uses the backup and restore privilege in Windows to copy files even if their NTFS ACLs deny permissions to the current user.
- /MT:8 – Specifies how many threads to use for performing multithreaded copies.

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

Migrating your on-premises file share configurations to Amazon FSx

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

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

To migrate existing file shares to FSx for Windows File Server

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

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

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

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

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

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

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

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

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

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

   ```
   $FSxAcceptedParameters = ("ContinuouslyAvailable", "Description",
    "ConcurrentUserLimit", "CATimeout", "FolderEnumerationMode", "CachingMode",
    "FullAccess", "ChangeAccess", "ReadAccess", "NoAccess", "SecurityDescriptor",
    "Path", "Name", "EncryptData")
   ```

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

Migrating your on-premises DNS configuration to FSx for Windows File Server

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

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

A DNS alias name has to meet the following requirements:

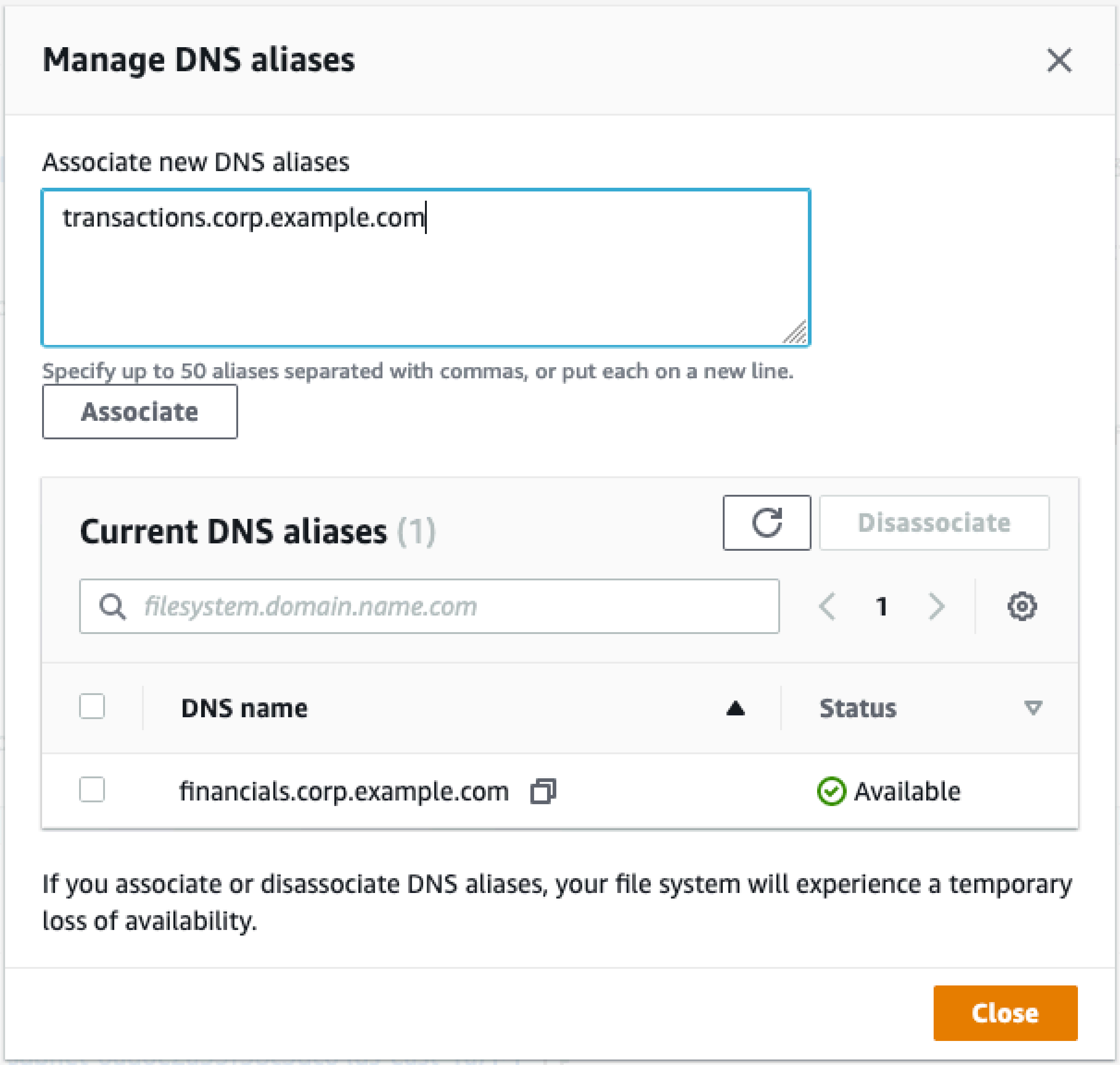
- Must be formatted as a fully qualified domain name (FQDN), for example, `accounting.example.com`.
- Can contain alphanumeric characters and the hyphen (-).
- Cannot start or end with a hyphen.
- Can start with a numeric.

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

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

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

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



Manage DNS aliases

Associate new DNS aliases

transactions.corp.example.com

Specify up to 50 aliases separated with commas, or put each on a new line.

Current DNS aliases (1)

	DNS name	Status
□	financials.corp.example.com	Available

If you associate or disassociate DNS aliases, your file system will experience a temporary loss of availability.

4. In the **Associate new aliases** box, enter the DNS aliases that you want to associate.

5. Choose **Associate** to add the aliases to the file system.

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

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

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

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

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

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

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

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

Cutting over operations to Amazon FSx for Windows File Server

After you have migrated your on-premises file storage, file share configuration, and DNS configuration, the next step is cutting over your operations to the FSx for Windows File Server file systems. To cut over to your FSx for Windows File Server file system, you perform the following steps:

- Prepare for the cut over.
  - Temporarily disconnect SMB clients from the original file system.
  - Perform a final file and file share configuration sync.

- Configure service principal names (SPNs) for your Amazon FSx file system.
- Update DNS CNAME records to point to your Amazon FSx file system.

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

Topics

- Preparing for the cutover to Amazon FSx
- Configure SPNs for Kerberos authentication
- Update the DNS CNAME records for the Amazon FSx file system

Preparing for the cutover to Amazon FSx

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

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

Configure SPNs for Kerberos authentication

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

There are two required SPNs for Kerberos authentication.

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

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

```
HOST/finance
```

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

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

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

To install the required PowerShell Active Directory module

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

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

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

1. Find any existing SPNs by using the following commands. Replace *alias_fqdn* with the DNS alias that you associated with the file system in Migrating your on-premises DNS configuration to FSx for Windows File Server.

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

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

   - Replace *alias_fqdn* with the full DNS alias that you associated with the file system in Migrating your on-premises DNS configuration to FSx for Windows File Server.
   - Replace *file_system_DNS_name* with the original file system's DNS name .

   ```
   ## Delete SPNs for original file system's AD computer object
   ```

```
$Alias = "alias_fqdn"
$FileSystemDnsName = "file_system_dns_name"
$FileSystemHost = (Resolve-DnsName ${FileSystemDnsName} | Where Type -eq 'A')
[0].Name.Split(".")[0]
$FSxAdComputer = (Get-AdComputer -Identity ${FileSystemHost})

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

3. Repeat these steps for each DNS alias that you associated with the file system in Migrating your on-premises DNS configuration to FSx for Windows File Server.

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

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

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

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

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

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

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

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

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

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

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

3. Repeat the previous steps for each DNS alias that you've associated with the file system in Migrating your on-premises DNS configuration to FSx for Windows File Server.

> **Note**
>
> You can enforce Kerberos authentication and encryption in transit with clients connecting to your file system using DNS aliases by setting the following Group Policy Objects (GPOs) in your Active Directory:
>
> - Restrict NTLM: Outgoing NTLM traffic to remote servers
> - Restrict NTLM: Add remote server exceptions for NTLM authentication

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

Update the DNS CNAME records for the Amazon FSx file system

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

To install the required PowerShell cmdlets

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

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

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

To update an existing a DNS CNAME record

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

   To run the script:

   - Replace *alias_fqdn* with the DNS alias that you associated with the file system.

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

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

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

2. Repeat the previous step for each DNS alias that you associated with the file system in Migrating your on-premises DNS configuration to FSx for Windows File Server.

Monitoring FSx for Windows File Server file systems

Monitoring is an important part of maintaining the reliability, availability, and performance of FSx for Windows File Server and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a failure if one occurs. However, before you start monitoring FSx for Windows File Server, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

For more information about logging and monitoring in FSx for Windows File Server, see the following topics.

Topics

- Automated and manual monitoring
- Monitoring with Amazon CloudWatch
- Logging Amazon FSx for Windows File Server API calls using AWS CloudTrail

Automated and manual monitoring

AWS provides various tools that you can use to monitor FSx for Windows File Server. You can configure some of these tools to do the monitoring for you, whereas some of the tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

Automated monitoring tools

You can use the following automated monitoring tools to watch FSx for Windows File Server and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring with Amazon CloudWatch.
- **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources. For more information, see What Is Amazon CloudWatch Logs? in the *Amazon CloudWatch Logs User Guide*.
- **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see Working with CloudTrail Log Files in the *AWS CloudTrail User Guide*.

Manual monitoring tools

Another important part of monitoring FSx for Windows File Server involves manually monitoring those items that the Amazon CloudWatch alarms don't cover. The FSx for Windows File Server, CloudWatch, and other AWS console dashboards provide an at-a-glance view of the state of your AWS environment.

Amazon FSx **Monitoring & performance** dashboard shows:

- Current warnings and CloudWatch alarms
- A summary of file system activity
- File system storage capacity and utilization
- File server and storage volume performance
- CloudWatch alarms

Amazon CloudWatch Dashboard shows:

- Current alarms and status
- Graphs of alarms and resources
- Service health status

In addition, you can use CloudWatch to do the following:

- Create customized dashboards to monitor the services you use.
- Graph metric data to troubleshoot issues and discover trends.
- Search and browse all your AWS resource metrics.
- Create and edit alarms to be notified of problems.

For more information about the Amazon FSx **Monitoring & performance** dashboard, see Using file system metrics.

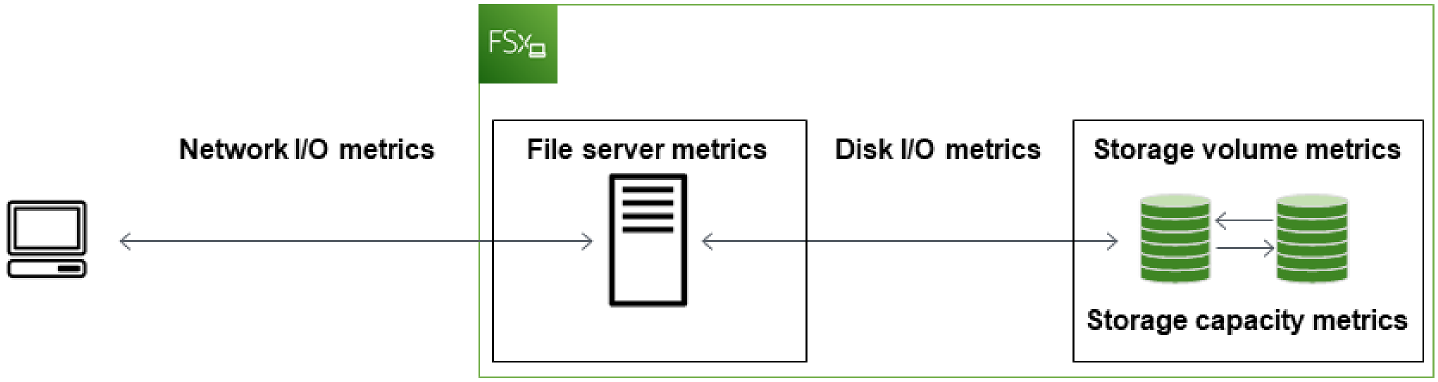
Monitoring with Amazon CloudWatch

Amazon CloudWatch collects and processes raw data from your FSx for Windows File Server file system into readable, near real-time metrics. These statistics are retained for a period of 15 months, giving you to access historical information to help gain perspectives on how your workflow or file system is performing.

FSx for Windows File Server publishes CloudWatch metrics in the following domains:

- Network I/O metrics measure activity between clients accessing the file system and the file server.
- File server metrics measure network throughput utilization, file server CPU and memory, and file server disk throughput and IOPS utilization.
- Disk I/O metrics measure activity between the file server and the storage volumes.
- Storage volume metrics measure disk throughput utilization for HDD storage volumes, and IOPS utilization for SSD storage volumes.
- Storage capacity metrics measure storage usage, including storage savings due to Data Deduplication.

The following diagram illustrates an FSx for Windows File Server file system, its components, and the metric domains.



By default, Amazon FSx for Windows File Server sends metric data to CloudWatch at 1-minute periods, with the following exceptions that are emitted in 5-minute intervals:

- `FileServerDiskThroughputBalance`
- `FileServerDiskIopsBalance`

For more information about CloudWatch, see What is Amazon CloudWatch? in the *Amazon CloudWatch User Guide*.

Metrics might not be published for Single-AZ file systems during file system maintenance or infrastructure component replacement, and for Multi-AZ file systems during failover and failback between the primary and secondary file servers.

Some Amazon FSx CloudWatch metrics are reported as raw *Bytes*. Bytes are not rounded to either a decimal or binary multiple of the unit.

Topics

- CloudWatch metrics and dimensions
- Using file system metrics
- Performance warnings and recommendations
- Accessing file system metrics
- Creating CloudWatch alarms

CloudWatch metrics and dimensions

FSx for Windows File Server publishes the following metrics into the `AWS/FSx` namespace in Amazon CloudWatch for all file systems:

- `DataReadBytes`
- `DataWriteBytes`
- `DataReadOperations`
- `DataWriteOperations`
- `MetadataOperations`
- `FreeStorageCapacity`

FSx for Windows File Server publishes the metrics described in the following sections into the `AWS/FSx` namespace in Amazon CloudWatch for file systems configured with a throughput capacity of at least 32 MBps.

Network I/O metrics

The `AWS/FSx` namespace includes the following network I/O metrics.

Metric	Description
`DataReadBytes`	The number of bytes for read operations for clients accessing the file system. Units: Bytes Valid statistics: `Sum`
`DataWriteBytes`	The number of bytes for write operations for clients accessing the file system. Units: Bytes Valid statistics: `Sum`
`DataReadOperations`	The number of read operations for clients accessing the file system. Units: Count Valid statistics: `Sum`
`DataWriteOperations`	The number of write operations for clients accessing the file system.

Metric	Description
	Units: Count Valid statistics: Sum
MetadataOperations	The number of metadata operations for clients accessing the file system. Units: Count Valid statistics: Sum
ClientConnections	The number of active connections between clients and the file server. Units: Count

File server metrics

The AWS/FSx namespace includes the following file server metrics.

Metric	Description
NetworkThroughputUtilization	The network throughput for clients accessing the file system, as a percentage of the provisioned limit. Units: Percent
CPUUtilization	The percentage utilization of your file server's CPU resources. Units: Percent
MemoryUtilization	The percentage utilization of your file server's memory resources. Units: Percent

Metric	Description
FileServerDiskThroughputUtilization	The disk throughput between your file server and its storage volumes, as a percentage of the provisioned limit determined by throughput capacity. Units: Percent
FileServerDiskThroughputBalance	The percentage of available burst credits for disk throughput between your file server and its storage volumes. Valid for file systems provisioned with throughput capacity of 256 MBps or less. Units: Percent
FileServerDiskIopsUtilization	The disk IOPS between your file server and storage volumes, as a percentage of the provisioned limit determined by throughput capacity. Units: Percent
FileServerDiskIopsBalance	The percentage of available burst credits for disk IOPS between your file server and its storage volumes. Valid for file systems provisioned with throughput capacity of 256 MBps or less. Units: Percent

Disk I/O metrics

The AWS/FSx namespace includes the following disk I/O metrics.

Metric	Description
DiskReadBytes	The number of bytes for read operations that access storage volumes. Units: Bytes Valid statistics: Sum

Metric	Description
DiskWriteBytes	The number of bytes for write operations that access storage volumes. Units: Bytes Valid statistics: Sum
DiskReadOperations	The number of read operations for the file server accessing storage volumes. Units: Count Valid statistics: Sum
DiskWriteOperations	The number of write operations for the file server accessing storage volumes. Units: Count Valid statistics: Sum

FSx for Windows storage volume metrics

The AWS/FSx namespace includes the following storage volume metrics.

Metric	Description
DiskThroughputUtilization	(HDD only) The disk throughput between your file server and its storage volumes, as a percentage of the provisioned limit determined by the storage volumes. Units: Percent
DiskThroughputBalance	(HDD only) The percentage of available burst credits for disk throughput and disk IOPS for the storage volumes. Units: Percent

Metric	Description
`DiskIopsUtilization`	(SSD only) The disk IOPS between your file server and storage volumes, as a percentage of the provisioned IOPS limit determined by the storage volumes. Units: Percent

Storage capacity metrics

The `AWS/FSx` namespace includes the following storage capacity metrics.

Metric	Description
`FreeStorageCapacity`	The amount of available storage capacity. Units: Bytes Valid statistics: `Average`, `Minimum`
`StorageCapacityUtilization`	Used physical storage capacity as a percentage of total storage capacity. Units: Percent
`DeduplicationSavedStorage`	The amount of storage space saved by data deduplication, if it is enabled. Units: Bytes

Namespace and dimensions for FSx for Windows File Server metrics

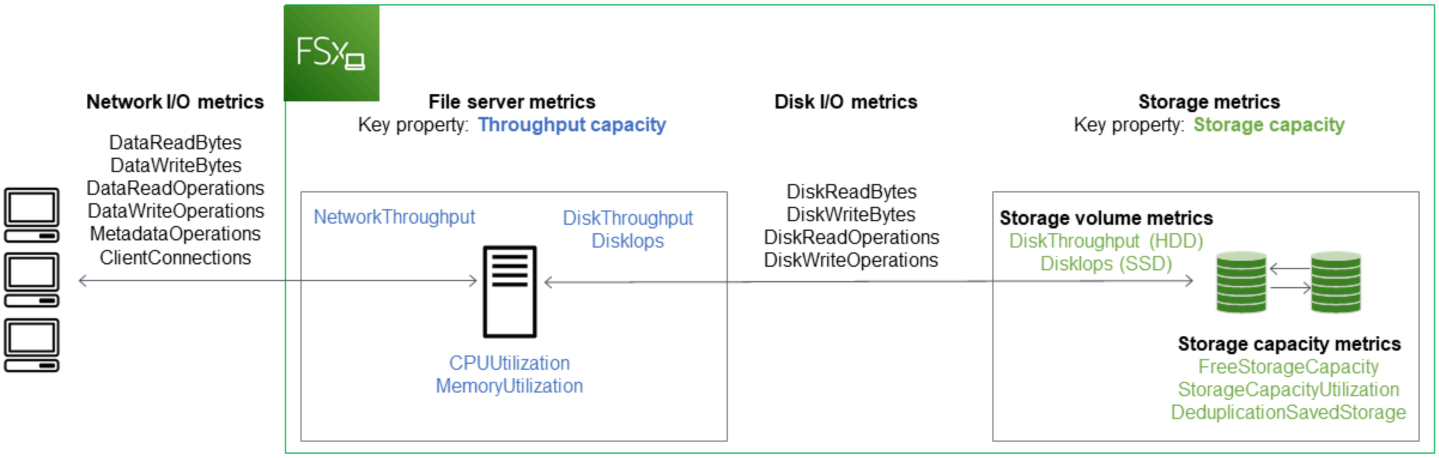
FSx for Windows File Server metrics use the `FSx` namespace and provide metrics for a single dimension, `FileSystemId`. You can find a file system's ID using the describe-file-systems AWS CLI command or the DescribeFileSystems API command. A file system ID takes the form of *`fs-0123456789abcdef0`*.

Using file system metrics

There are two primary architectural components of each Amazon FSx file system:

- The **file server** that serves data to clients accessing the file system.
- The **storage volumes** that host the data in your file system.

FSx for Windows File Server reports metrics in CloudWatch that track the performance and resource utilization for your file system's file server and storage volumes. The following diagram illustrates an Amazon FSx file system with its architectural components, and the performance and resource CloudWatch metrics available for monitoring. The key property shown for a set of metrics is the file system property that determines the capacity for those metrics. Adjusting that property modifies the file system's performance for that set of metrics.



Use the **Monitoring & performance** panel in the Amazon FSx console to view the FSx for Windows File Server CloudWatch metrics described in the following table.

Monitoring & performance panel	How do I...	Chart	Relevant metrics
Summary	...determine my file system's total IOPS?	Total IOPS	`SUM(DataReadOperations + DataWriteOperations`

Monitor g & perform ce panel	How do I...	Chart	Relevant metrics
			+ `MetadataOperations`)/Period (in seconds)
	...determine my file system's total throughput?	Total throughput	SUM(`DataReadBytes` + `DataWriteBytes`)/ Period (in seconds)
	...determine the amount of available storage capacity on my file system?	Available storage capacity	`FreeStorageCapacity`
	...determine the number of connections established between clients and the file server?	Client connections	`ClientConnections`
Storage	...determine the amount of used physical disk space as a percentage of the file system's total storage capacity?	Storage capacity utilization	`StorageCapacityUtilization`
	...determine the amount of physical disk space saved by data deduplication?	Storage saved from Data Deduplication	`DeduplicationSavedStorage`
Performance - File server	...determine the network throughput for clients accessing the file system, as a percentage of the file system's provisioned throughput?	Network throughput utilization	`NetworkThroughputUtilization` [1]

Monitoring & performance panel	How do I…	Chart	Relevant metrics
	…determine the disk throughput between file server and its storage volumes, as a percentage of the provisioned limit determined by Throughput Capacity?	Disk throughput utilization	`FileServerDiskThroughputUtilization` [1]
	…determine the percentage of available burst credits for disk throughput between the file server and its storage volumes?	Disk throughput burst balance	`FileServerDiskThroughputBalance`
	…determine the amount of disk IOPS between the file server and storage volumes, as a percentage of the provisioned limit determined by Throughput Capacity?	Disk IOPS utilization	`FileServerDiskIopsUtilization`
	…determine the percentage of available burst credits for disk IOPS between the file server and storage volumes?	Disk IOPS burst balance	`FileServerDiskIopsBalance`
	…determine the file server's CPU utilization percentage?	CPU utilization	`CPUUtilization`
	…determine the file server's memory utilization percentage?	Memory utilization	`MemoryUtilization`
Performance – Storage volumes	…determine the throughput for operations that access storage volumes, as a percentage of the provisioned limit determined by HDD Storage Capacity?	Disk throughput utilization (HDD)	`DiskThroughputUtilization`

Monitoring & performance panel	How do I…	Chart	Relevant metrics
	…determine the percentage of available throughput and IOPS burst credits for operations that access HDD storage volumes?	Disk throughput burst balance (HDD)	`DiskThroughputBalance` [2]
	…determine the IOPS for operations that access storage volumes, as a percentage of the provisioned limit determined by HDD Storage Capacity?	Disk IOPS utilization (HDD)	SUM(`DiskReadOperations` + `DiskWriteOperations`) / `Period` (in seconds) / (12 * provisioned HDD storage capacity in TiB)
	…determine the IOPS for operations that access storage volumes, as a percentage of the provisioned limit determined by SSD Storage Capacity?	Disk IOPS utilization (SSD)	`DiskIopsUtilization`

Note

[1] We recommend that you maintain an average throughput capacity utilization under 50% to ensure that you have enough spare throughput capacity for unexpected spikes in your workload, as well as for any background Windows storage operations (such as storage synchronization, deduplication, or shadow copies).

[2] HDD storage volumes can experience significant performance variations depending on the workload. Sudden spikes in IOPS or throughput can lead to disk performance degradation. For more information, see HDD burst performance.

Performance warnings and recommendations

FSx for Windows provides you with performance warnings for file systems configured with a throughput capacity of at least 32 MBps. Amazon FSx displays a warning for a set of the CloudWatch metrics whenever one of these metrics has approached or crossed a predetermined threshold for multiple consecutive data points. These warnings provide you with actionable recommendations that you can use to optimize your file system's performance.

Warnings are accessible in several areas of the **Monitoring & performance** dashboard. All active or recent Amazon FSx performance warnings and any CloudWatch alarms configured for the file system that are in an ALARM state appear in the **Monitoring & performance** panel in the **Summary** section. The warning also appears in the section of the dashboard that the metric graph is displayed.

You can create CloudWatch alarms for any of the Amazon FSx metrics. For more information, see Creating CloudWatch alarms.

Use performance warnings to improve file system performance

Amazon FSx provides actionable recommendations that you can use to optimize your file system's performance. These recommendations describe how you can address a potential performance bottle neck. You can take the recommended action if you expect the activity to continue, or if it's causing an impact to your file system's performance. Depending on which metric has triggered a warning, you can resolve it by increasing either the file system's throughput capacity or storage capacity, as described in the following table.

If there's a warning for this metric	Do this
Network throughput – utilization	Increase throughput capacity
File server > Disk IOPS – utilization	
File server > Disk throughput – utilization	
File server > Disk IOPS - burst balance	
File server > Disk throughput – burst balance	
Storage capacity utilization	Increase storage capacity

If there's a warning for this metric	Do this
Storage volume > Disk throughput – utilization (HDD)	Increase storage capacity or switch to SDD storage type
Storage volume > Disk throughput – burst balance (HDD)	
Storage volume > Disk IOPS – utilization (SSD)	Increase SSD IOPS

Note

Certain file system events can consume disk I/O performance resources and potentially trigger performance warnings. For example:

- The optimization phase of storage capacity scaling can generate increased disk throughput, as described in Storage capacity increases and file system performance
- For Multi-AZ file systems, events such as throughput capacity scaling, hardware replacement, or Availability Zone disruption result in automatic failover and failback events. Any data changes that occur during this time need to be synchronized between the primary and secondary file servers, and Windows Server runs a data synchronization job that can consume disk I/O resources. For more information, see Managing throughput capacity.

For more information file system performance, see FSx for Windows File Server performance.

Accessing file system metrics

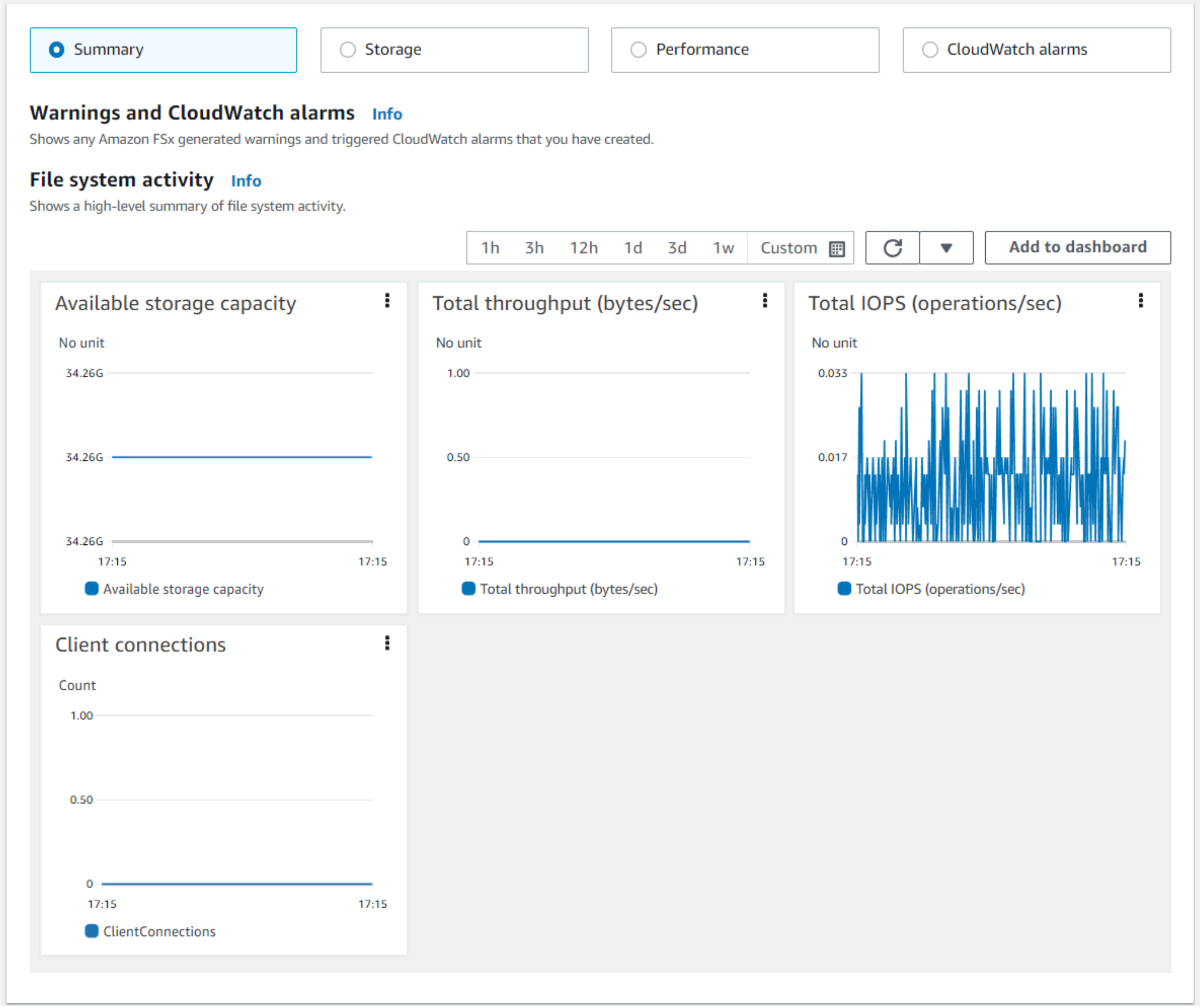
You can see Amazon FSx metrics for CloudWatch in the following ways.

- The Amazon FSx console
- The CloudWatch console
- The CloudWatch CLI
- The CloudWatch API

The following procedures describe how to access your file system's metrics using these various tools.

To view file system metrics using the Amazon FSx console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. To display the **File system details** page, choose **File systems** in the navigation pane.
3. Choose the file system whose metrics you want to view.
4. To view graphs of the file system's metrics, choose **Monitoring & performance** on the second panel.



● Summary ○ Storage ○ Performance ○ CloudWatch alarms

Warnings and CloudWatch alarms Info

Shows any Amazon FSx generated warnings and triggered CloudWatch alarms that you have created.

File system activity Info

Shows a high-level summary of file system activity.

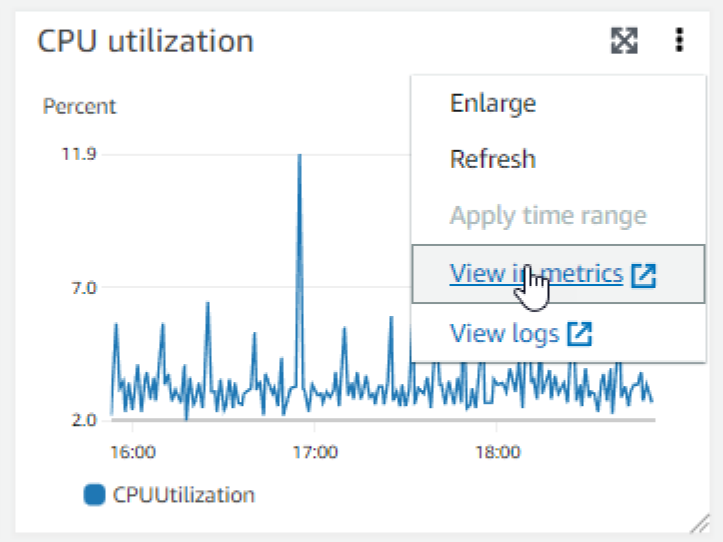
- The **Summary** metrics are displayed by default, showing any active warnings and CloudWatch alarms along with **File system activity** metrics.
- Choose **Storage** to view storage capacity and utilization metrics.
- Choose **Performance** to view file server and storage performance metrics

- Choose **CloudWatch alarms** to view graphs of any alarms configured for the file system.

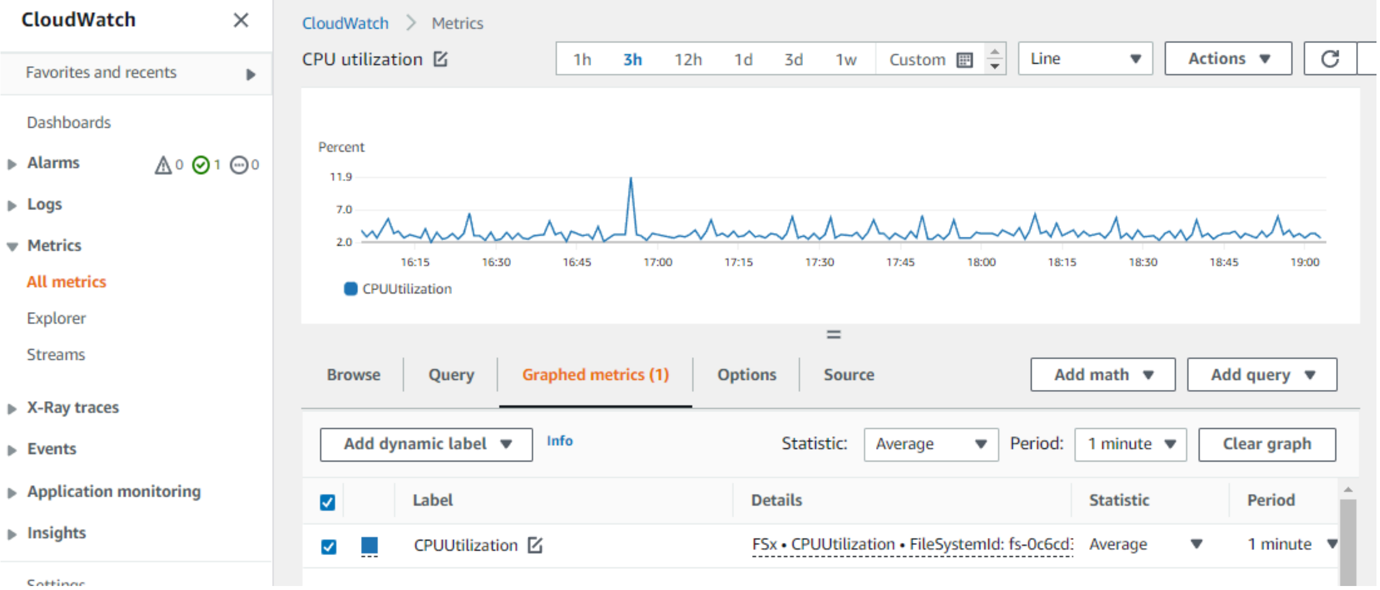
For more information, see Using file system metrics

To view metrics in the CloudWatch console

1. To view a file system metric in the **Metrics** page of the Amazon CloudWatch console, navigate to the metric in the **Monitoring & performance** panel of the Amazon FSx console.
2. Choose **View in metrics** from the actions menu in the upper right of the metric graph, as shown in the following image.



This opens the **Metrics** page in the CloudWatch console, showing the metric graph, as shown in the following image.



Statistic: Average
Period: 1 minute

☑		Label	Details	Statistic	Period
☑		CPUUtilization	FSx • CPUUtilization • FileSystemId: fs-0c6cd:	Average	1 minute

To add metrics to a CloudWatch dashboard

1. To add a set of FSx for Windows file system metrics to a dashboard in the CloudWatch console, choose the set of metrics (**Summary**, **Storage**, or **Performance**) in the **Monitoring & performance** panel of the Amazon FSx console.
2. Choose **Add to dashboard** in the upper right of the panel, this opens the CloudWatch console.
3. Select an existing CloudWatch dashboard from the list, or create a new dashboard. For more information, see Using Amazon CloudWatch dashboards in the *Amazon CloudWatch User Guide*.

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.

```
$ aws cloudwatch list-metrics --namespace "AWS/FSx"
aws cloudwatch list-metrics --namespace "AWS/FSx"
{
    "Metrics": [
        {
            "Namespace": "AWS/FSx",
            "MetricName": "DataWriteOperationTime",
            "Dimensions": [
                {
                    "Name": "FileSystemId",
                    "Value": "fs-09a106ebc3a0bb087"
                }
            ]
        },
        {
            "Namespace": "AWS/FSx",
            "MetricName": "CapacityPoolWriteBytes",
            "Dimensions": [
                {
                    "Name": "VolumeId",
                    "Value": "fsvol-0cb2281509f5db3c2"
                },
                {
                    "Name": "FileSystemId",
                    "Value": "fs-09a106ebc3a0bb087"
                }
            ]
        },
        {
            "Namespace": "AWS/FSx",
            "MetricName": "DiskReadBytes",
            "Dimensions": [
                {
                    "Name": "FileSystemId",
                    "Value": "fs-09a106ebc3a0bb087"
                }
            ]
        },
        {
            "Namespace": "AWS/FSx",
            "MetricName": "CompressionRatio",
            "Dimensions": [
                {
                    "Name": "FileSystemId",
                    "Value": "fs-0f84c9a176a4d7c92"
                }
            ]
        },
.
.
.
}
```

Using the CloudWatch API

To access metrics from the CloudWatch API

- Call `GetMetricStatistics`. For more information, see Amazon CloudWatch API Reference.

Creating CloudWatch alarms

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy.

Alarms invoke actions for sustained state changes only. CloudWatch alarms don't invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. You can create an alarm from the Amazon FSx console or the CloudWatch console.

The following procedures describe how to create alarms for Amazon FSx using the console, AWS CLI, and API.

To set a CloudWatch alarm (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation pane, choose **File systems**, and then choose the file system you want to create the alarm for.
3. Choose the **Actions** menu, and choose **View details**.
4. On the **Summary** page, choose **Monitoring and performance**.
5. Choose **CloudWatch alarms**.
6. Choose **Create CloudWatch alarm**. You are redirected to the CloudWatch console.
7. Choose **Select metrics**, and choose **Next**.
8. In the **Metrics** section, choose **FSX**.
9. Choose **File System Metrics**, choose the metric you want to set the alarm for, and then choose **Select metric**.
10. In the **Conditions** section, choose the conditions you want for the alarm, and choose **Next.**

> **Note**
>
> Metrics may not be published during file system maintenance for Single-AZ file systems, or during failover and failback to or from the primary or secondary servers for Multi-AZ file systems. To prevent unnecessary and misleading alarm condition changes and to configure your alarms so that they are resilient to missing data points, see Configuring how CloudWatch alarms treat missing data in the *Amazon CloudWatch User Guide*.

11. If you want CloudWatch to send you an email or SNS notification when the alarm state triggers the action, choose an alarm state for **Whenever this alarm state is**.

    For **select an SNS topic**, choose an existing SNS topic. If you select **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms. Choose **Next**.

    > **Note**
    >
    > If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.

12. Fill in the **Name**, **Description**, and **Whenever** values for the metric, and choose **Next**.
13. On the **Preview and create** page, review the alarm you're about to create, and then choose **Create Alarm**.

To set alarms using the CloudWatch console

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Choose **Create Alarm** to start the **Create Alarm Wizard**.
3. Choose **FSx Metrics**, and scroll through the Amazon FSx metrics to locate the metric you want to place an alarm on. To display just the Amazon FSx metrics in this dialog box, search on the file system ID of your file system. Select the metric to create an alarm on, and choose **Next**.
4. Fill in the **Name**, **Description**, and **Whenever** values for the metric.

5. If you want CloudWatch to send you an email when the alarm state is reached, for **Whenever this alarm**, choose **State is ALARM**. For **Send notification to**, choose an existing SNS topic. If you select **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms.

> **Note**
>
> If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.

6. At this point, the **Alarm Preview** area gives you a chance to preview the alarm you're about to create. Choose **Create Alarm**.

To set a CloudWatch alarm (CLI)

- Call `put-metric-alarm`. For more information, see *AWS CLI Command Reference*.

To set an alarm (API)

- Call `PutMetricAlarm`. For more information, see *Amazon CloudWatch API Reference*.

Logging Amazon FSx for Windows File Server API calls using AWS CloudTrail

Amazon FSx for Windows File Server 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. The calls captured include calls from the Amazon FSx console and code calls to the 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, 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 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 Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

All Amazon FSx actions are logged by CloudTrail and are documented in the Amazon FSx API Reference. For example, calls to the `CreateFileSystem`, `CreateBackup` and `TagResource` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Amazon FSx log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `TagResource` operation when a tag for a file system is created from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": “111122223333”,
        "arn": "arn:aws:sts::111122223333:root”,
        "accountId": “111122223333”,
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE”,
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T22:36:07Z"
            }
        }
    },
    "eventTime": "2018-11-14T22:36:07Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": “192.0.2.0”,
    "userAgent": “console.amazonaws.com”,
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789”
    },
    "responseElements": null,
    "requestID": “aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51”,
    "eventID": “bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p”,
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": “111122223333”
}
```

The following example shows a CloudTrail log entry that demonstrates the `UntagResource` action when a tag for a file system is deleted from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T23:40:54Z"
            }
        }
    },
    "eventTime": "2018-11-14T23:40:54Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789"
    },
    "responseElements": null,
    "requestID": "aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51",
    "eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": "111122223333"
}
```

Security in Amazon FSx

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the Amazon Web Services Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to Amazon FSx for Windows File Server, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon FSx for Windows File Server. The following topics show you how to configure Amazon FSx for Windows File Server 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.

Topics

- Data protection in Amazon FSx for Windows File Server
- File- and folder-level access control using Windows ACLs
- File system access control with Amazon VPC
- Logging end user access with file access auditing
- Identity and access management for Amazon FSx for Windows File Server
- Compliance Validation for Amazon FSx for Windows File Server
- Amazon FSx for Windows File Server and interface VPC endpoints

Data protection in Amazon FSx for Windows File Server

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

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

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

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

Data encryption in FSx for Windows File Server

Amazon FSx for Windows File Server supports encryption of data at rest and encryption of data in transit. Encryption of data at rest is automatically enabled when creating an Amazon FSx file system. 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. 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.

If your organization is subject to corporate or regulatory policies that require encryption of data and metadata at rest, your data is automatically encrypted at rest. We also recommend that you enable encryption of data in transit by mounting your file system using encryption of data in transit.

Encryption of data at rest

All Amazon FSx file systems are encrypted at rest with keys managed using AWS Key Management Service (AWS KMS). Data is automatically encrypted before being written to the file system, and automatically decrypted as it is read. These processes are handled transparently by Amazon FSx, so you don't have to modify your applications.

Amazon FSx uses an industry-standard AES-256 encryption algorithm to encrypt Amazon FSx data and metadata at rest. For more information, see Cryptography Basics in the *AWS Key Management Service Developer Guide*.

Note

The AWS key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms. The infrastructure is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

How Amazon FSx uses AWS KMS

Amazon FSx integrates with AWS KMS for key management. Amazon FSx uses an AWS KMS key to encrypt your file system. You choose the KMS key used to encrypt and decrypt file systems (both data and metadata). You can enable, disable, or revoke grants on this KMS key. This KMS key can be one of the two following types:

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

If you use a customer managed key as your KMS key for file data encryption and decryption, you can enable key rotation. When you enable key rotation, AWS KMS automatically rotates your key once per year. Additionally, with a customer managed key, you can choose when to disable, re-enable, delete, or revoke access to your KMS key at any time. For more information, see Rotating AWS KMS keys in the *AWS Key Management Service Developer Guide.*

Amazon FSx Key policies for AWS KMS

Key policies are the primary way to control access to KMS keys. For more information on key policies, see Using key policies in AWS KMS in the *AWS Key Management Service Developer Guide.* The following list describes all the AWS KMS-related permissions supported by Amazon FSx for encrypted at rest file systems:

- **kms:Encrypt** – (Optional) Encrypts plaintext into ciphertext. This permission is included in the default key policy.
- **kms:Decrypt** – (Required) Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted. This permission is included in the default key policy.
- **kms:ReEncrypt** – (Optional) Encrypts data on the server side with a new KMS key, without exposing the plaintext of the data on the client side. The data is first decrypted and then re-encrypted. This permission is included in the default key policy.
- **kms:GenerateDataKeyWithoutPlaintext** – (Required) Returns a data encryption key encrypted under a KMS key. This permission is included in the default key policy under **kms:GenerateDataKey***.
- **kms:CreateGrant** – (Required) Adds a grant to a key to specify who can use the key and under what conditions. Grants are alternate permission mechanisms to key policies. For more

information on grants, see Using grants in the AWS Key Management Service Developer Guide. This permission is included in the default key policy.

- **kms:DescribeKey** – (Required) Provides detailed information about the specified KMS key. This permission is included in the default key policy.
- **kms:ListAliases** – (Optional) Lists all of the key aliases in the account. When you use the console to create an encrypted file system, this permission populates the list of KMS keys. We recommend using this permission to provide the best user experience. This permission is included in the default key policy.

Encryption of data in transit

Encryption of data in transit is supported on file shares that are mapped on a compute instance that supports SMB protocol 3.0 or newer. This includes all Windows versions starting from Windows Server 2012 and Windows 8, and all Linux clients with Samba client version 4.2 or newer. Amazon FSx for Windows File Server automatically encrypts data in transit using SMB encryption as you access your file system without the need for you to modify your applications.

SMB encryption uses AES-128-GCM or AES-128-CCM (with the GCM variant being chosen if the client supports SMB 3.1.1) as its encryption algorithm, and also provides data integrity with signing using SMB Kerberos session keys. The use of AES-128-GCM leads to better performance, for example, up to a 2x performance improvement when copying large files over encrypted SMB connections.

To meet compliance requirements for always encrypting data-in-transit, you can limit file system access to only allow access to clients that support SMB encryption. You can also enable or disable in-transit encryption per file share or to the entire file system. This allows you to have a mix of encrypted and unencrypted file shares on the same file system.

Managing encryption in transit

You can use a set of custom PowerShell commands to control the encryption of your data in transit between your FSx for Windows File Server file system and clients. You can limit file system access to only clients supporting SMB encryption so that data-in-transit is always encrypted. When enforcement is turned on for encryption of data-in-transit, users accessing the file system from clients that do not support SMB 3.0 encryption will not be able to access file shares for which encryption is turned on.

You can also control encryption of data-in-transit on a file share-level instead of file server-level. You can use file share-level encryption controls to have a mix of encrypted and unencrypted file shares on the same file system if you want to enforce encryption in-transit for some file shares that have sensitive data, and allow all users to access some other file shares. Server-wide encryption has precedence over share level encryption. If global encryption is enabled, you cannot selectively disable encryption for certain shares.

You can manage 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 Using the Amazon FSx CLI for PowerShell.

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

Encryption in Transit Command	Description
Get-FSxSmbServerConfiguration	Retrieves the Server Message Block (SMB) server configuration. In the system response you can determine the encryption in transit settings for your filesystem based on the values for the `EncryptData` and `RejectUnencryptedAccess` properties.
Set-FSxSmbServerConfiguration	This command has two options for configuring in-transit encryption globally on the file system: • `-EncryptData $True\|$False` – Set this parameter to `True` to turn on in-transit data encryption. Set this parameter to `False` to turn off in-transit data encryption. • `-RejectUnencryptedAccess $True\|$False` – Set this parameter to `True` to disallow clients that do not support encryption to access the file system. Set this parameter to `False` to allow clients that do not support encryption to access the file system.
Set-FSxSmbShare -name *name* -EncryptData $True	Set this parameter to `True` to turn on in-transit data encryption for the share. Set this parameter to `False` to turn off in-transit data encryption for the share.

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 -?**.

File- and folder-level access control using Windows ACLs

Amazon FSx for Windows File Server supports identity-based authentication over the Server Message Block (SMB) protocol through Microsoft Active Directory. Active Directory is the Microsoft directory service to store information about objects on the network and make this information easy for administrators and users to find and use. These objects typically include shared resources such as file servers, and the network user and computer accounts. To learn more about Active Directory support in Amazon FSx, see Working with Microsoft Active Directory.

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 **Authenticated Users**, including users in the domain the file system is joined to, and users in domains with a trust relationship. They also allow full control to the delegated administrators group in your Active Directory that is delegated to perform administrative actions on your file systems. If you're integrating your file system with AWS Managed Microsoft AD, this group is AWS Delegated FSx Administrators. If you're integrating your file system with your self-managed Microsoft AD setup, this group can be Domain Admins. Or it can be a custom delegated administrators group that you specified when creating the file system. To change the ACLs, you can map the share as a user that is a member of the delegated administrators group.

⚠ **Warning**

Amazon FSx requires that the SYSTEM user have **Full control** NTFS ACL permissions on all folders within your file system. Do not change the NTFS ACL permissions for this user on your folders. Doing so can make your file share inaccessible and prevent file system backups from being usable.

Related Links

- What Is AWS Directory Service? in the AWS Directory Service Administration Guide.
- Create Your AWS Managed Microsoft AD directory in the *AWS Directory Service Administration Guide*.
- When to Create a Trust Relationship in the *AWS Directory Service Administration Guide*.
- Step 1. Setting up an Active Directory.

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 Direct Connect or VPN, or resources within peered VPCs can access your file system's network interface. For more information, see What is Amazon VPC? in the *Amazon VPC User Guide.*

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

FSx for Windows File Server supports VPC sharing, which enables you to view, create, modify, and delete resources in a shared subnet in a VPC owned by another AWS account. For more information, see Working with Shared VPCs in the *Amazon VPC User Guide*.

Amazon VPC Security Groups

To further control network traffic going through your file system's elastic network interface(s) within your VPC, 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(s).

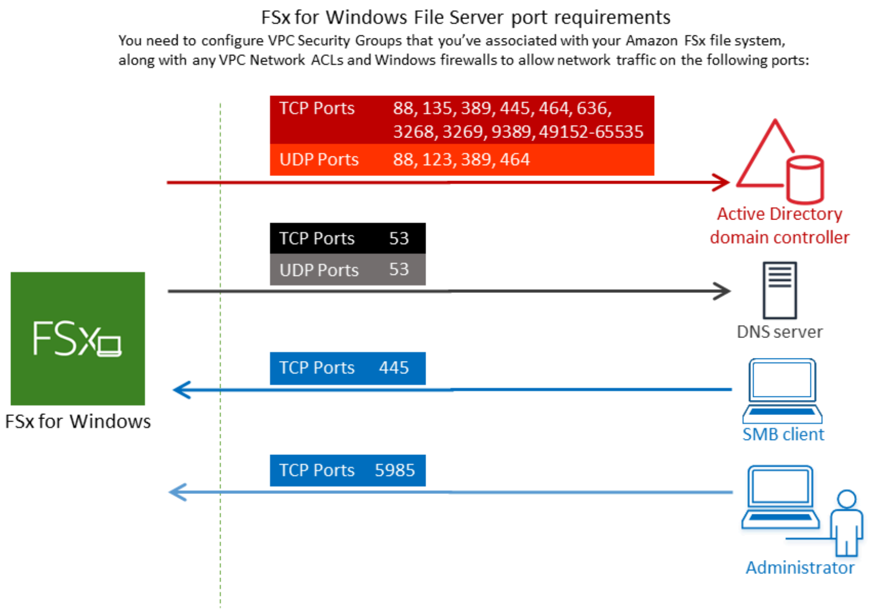
To use a security group to control access to your Amazon FSx file system, add inbound and outbound rules. Inbound rules control incoming traffic, and outbound rules control outgoing traffic from your file system. Make sure that you have the right network traffic rules in your security group to map your 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.*

To create a security group for Amazon FSx

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. Specify a name and description for the security group.
5. For **VPC**, choose the Amazon VPC associated with your file system to create the security group within that VPC.
6. Add the following rules to allow outbound network traffic on the following ports:

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



FSx for Windows File Server port requirements

You need to configure VPC Security Groups that you've associated with your Amazon FSx file system, along with any VPC Network ACLs and Windows firewalls to allow network traffic on the following ports:

TCP Ports 88, 135, 389, 445, 464, 636, 3268, 3269, 9389, 49152-65535
UDP Ports 88, 123, 389, 464
Active Directory domain controller
TCP Ports 53
UDP Ports 53
DNS server
FSx
FSx for Windows
TCP Ports 445
SMB client
TCP Ports 5985
Administrator

The following table identifies the role of each port.

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

Protocol	Ports	Role
TCP	3268	Microsoft Global Catalog
TCP	3269	Microsoft Global Catalog over SSL
TCP	5985	WinRM 2.0 (Microsoft Windows Remote Management)
TCP	9389	Microsoft AD DS Web Services, PowerShell
TCP	49152 - 65535	Ephemeral ports for RPC

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

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

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

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

Note

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

Now that you've created your security group, you can associate it with your Amazon FSx file system's elastic network interface(s).

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(s); for example, **ENI-01234567890123456**. For Single-AZ file systems, you'll see a single network interface. For Multi-AZ file systems, you'll see one network interface in the Preferred subnet and one in the Standby subnet.
4. For each network interface, choose the network interface and in **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.*

Logging end user access with file access auditing

Amazon FSx for Windows File Server supports auditing end-user access to files, folders, and file shares. You can choose to send a file system's audit event logs to other AWS services that offer a rich set of features. These include the enabling querying, processing, storing and archiving logs, issuing notifications, and triggering actions to further advance your security and compliance goals.

For more information about using file access auditing to get insights into access patterns and implement security notifications for end user activity, see File storage access patterns insights and Implementing security notifications for end user activity.

> **Note**
>
> File access auditing is supported only on FSx for Windows file systems with a throughput capacity of 32 MBps or greater. You can modify the throughput capacity on existing file systems. For more information, see Managing throughput capacity.

File access auditing enables you to record end-user accesses of individual files, folders, and file shares based on your defined audit controls. Audit controls are also known as NTFS system access control lists (SACLs). If you already have audit controls set up on your existing file data, you can take advantage of file access auditing by creating a new Amazon FSx for Windows File Server file system and migrating your data.

Amazon FSx supports the following Windows audit events for file, folder, and file share accesses:

- For file accesses, it supports: All, Traverse folder / Execute file, List folder / Read data, Read attributes, Create files / Write data, Create folders / Append data, Write attributes, Delete subfolders and files, Delete, Read permissions, Change permissions, and Take ownership.
- For file share accesses, it supports: Connect to a file share.

Across file, folder, and file share accesses, Amazon FSx supports logging of successful attempts (such as a user with sufficient permissions successfully accessing a file or file share), failed attempts, or both.

You can configure whether you want access auditing only on files and folders, only on file shares, or both. You can also configure which types of accesses should be logged (successful attempts only, failed attempts only, or both). You can also turn off file access auditing at any time.

> **Note**
>
> File access auditing records end-user access data only from the time it is enabled. That is, file access auditing doesn't generate audit event logs of end-user file, folder, and file share access activity that occurred before file access auditing was enabled.

The maximum rate of access audit events supported is 5,000 events per second. Access audit events are not generated for each file read and write operation, but generated once per file metadata operation, such as when a user creates, opens, or deletes a file.

Topics

- Audit event log destinations
- Migrating your audit controls
- Viewing event logs
- Setting file and folder auditing controls
- Managing file access auditing

Audit event log destinations

When you enable file access auditing, you must configure an AWS service to which Amazon FSx sends the audit event logs. You can send audit event logs to either an Amazon CloudWatch Logs log stream in a CloudWatch Logs log group or an Amazon Data Firehose delivery stream. You choose the audit event logs destination either when you create your Amazon FSx for Windows File Server file system, or anytime after by updating an existing file system. For more information, see Managing file access auditing.

Following are some recommendations that may help you decide which audit event logs destination to choose:

- Choose CloudWatch Logs if you want to store, view, and search audit event logs in the Amazon CloudWatch console, run queries on the logs using CloudWatch Logs Insights, and trigger CloudWatch alarms or Lambda functions.

- Choose Amazon Data Firehose if you want to continuously stream events to storage in Amazon S3, to a database in Amazon Redshift, to Amazon OpenSearch Service, or to AWS Partner solutions such as Splunk or Datadog for further analysis.

By default, Amazon FSx will create and use a default CloudWatch Logs log group in your account as the audit event log destination. If you want to use a custom CloudWatch Logs log group or use Firehose as the audit event log destination, here are the requirements for the names and locations of the audit event log destination:

- The name of the CloudWatch Logs log group must begin with the `/aws/fsx/` prefix. If you don't have an existing CloudWatch Logs log group when you create or update a file system on the console, Amazon FSx can create and use a default log stream in the CloudWatch Logs `/aws/fsx/windows` log group. If you don't want to use the default log group, the configuration UI lets you create a CloudWatch Logs log group when you create or update your file system on the console.
- The name of the Firehose delivery stream must begin with the `aws-fsx-` prefix. If you don't have an existing Firehose delivery stream, you can create one when you create or update your file system at the console.
- The Firehose delivery stream must be configured to use `Direct PUT` as its source. You cannot use an existing Kinesis data stream as a data source for your delivery stream.
- The destination (either CloudWatch Logs log group or Firehose delivery stream) must be in the same AWS partition, AWS Region, and AWS account as your Amazon FSx file system.

You can change the audit event log destination at any time (for example, from CloudWatch Logs to Firehose). When you do so, new audit event logs are sent only to the new destination.

Best effort audit event log delivery

Typically, audit event log records are delivered to the destination in minutes, but can sometimes take longer. On very rare occasions, audit event log records might be missed. If your use case requires particular semantics (for example, ensuring that no audit events are missed), we recommend that you account for missed events when designing your workflows. You can audit for missed events by scanning the file and folder structure on your file system.

Migrating your audit controls

If you have audit controls (SACLs) already set up on your existing file data, you can create an Amazon FSx file system and migrate your data to your new file system. We recommend using AWS DataSync to transfer data and the associated SACLs to your Amazon FSx file system. As an alternative solution, you can use Robocopy (Robust File Copy). For more information, see Migrating existing file storage to Amazon FSx.

Viewing event logs

You can view the audit event logs after Amazon FSx has started emitting them. Where and how you view the logs depends on the audit event log destination:

- You can view CloudWatch Logs logs by going to the CloudWatch console and choosing the log group and log stream to which your audit event logs are sent. For more information, see View log data sent to CloudWatch Logs in the *Amazon CloudWatch Logs User Guide*.

  You can use CloudWatch Logs Insights to interactively search and analyze your log data. For more information, see Analyzing Log Data with CloudWatch Logs Insights, in the *Amazon CloudWatch Logs User Guide*.

  You can also export the audit event logs to Amazon S3. For more information, see Exporting Log Data to Amazon S3, also in the *Amazon CloudWatch Logs User Guide*.
- You can't view the audit event logs on Firehose. However, you can configure Firehose to forward the logs to a destination that you can read from. The destinations include Amazon S3, Amazon Redshift, Amazon OpenSearch Service, and partner solutions such as Splunk and Datadog, For more information, see Choose destination in the *Amazon Data Firehose Developer Guide*.

Audit event fields

This section provides descriptions of the information in audit event logs and examples of audit events.

Following are descriptions of the salient fields in a Windows audit event.

- **EventID** refers to the Microsoft-defined Windows event log event ID. See Microsoft documentation for information on file system events and file share events.
- **SubjectUserName** refers to the user performing the access.

- **ObjectName** refers to the target file, folder, or file share that was accessed.
- **ShareName** is available for events that are generated for file share access. For example, `EventID 5140` is generated when a network share object was accessed.
- **IpAddress** refers to the client that initiated the event for file share events.
- **Keywords**, when available, refer to whether the file access was successful or a failure. For successful accesses, the value is `0x8020000000000000`. For failed accesses, the value is `0x8010000000000000`.
- **TimeCreated SystemTime** refers to the time the event was generated in the system and shown in <YYYY-MM-DDThh:mm:ss.s>Z format.
- **Computer** refers to the DNS name of the file system Windows Remote PowerShell Endpoint and can be used to identify the file system.
- **AccessMask**, when available, refers to the type of file access performed (for example, ReadData, WriteData).
- **AccessList** refers to requested or granted access to an Object. For details, see the table below and Microsoft documentation (such as in Event 4556).

Access Type	Access Mask	Value
Read Data or List Directory	0x1	%%4416
Write Data or Add File	0x2	%%4417
Append Data or Add Subdirectory	0x4	%%4418
Read Extended Attributes	0x8	%%4419
Write Extended Attributes	0x10	%%4420
Execute/Traverse	0x20	%%4421
Delete Child	0x40	%%4422
Read Attributes	0x80	%%4423
Write Attributes	0x100	%%4424

Access Type	Access Mask	Value
Delete	0x10000	%%1537
Read ACL	0x20000	%%1538
Write ACL	0x40000	%%1539
Write Owner	0x80000	%%1540
Synchronize	0x100000	%%1541
Access Security ACL	0x1000000	%%1542

Following are some key events with examples. Note that the XML is formatted for readability.

Event ID 4660 is logged when an object is deleted.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4660</EventID><Version>0</Version><Level>0</Level>
<Task>12800</Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-05-18T04:51:56.916563800Z'/>
<EventRecordID>315452</EventRecordID><Correlation/>
<Execution ProcessID='4' ThreadID='5636'/><Channel>Security</Channel>
<Computer>amznfsxgyzohmw8.example.com</Computer><Security/></System><EventData>
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x50932f71</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='HandleId'>0x12e0</Data><Data Name='ProcessId'>0x4</Data><Data
 Name='ProcessName'></Data>
<Data Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data></EventData></
Event>
```

Event ID 4659 is logged on a request to delete a file.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
```

```
<EventID>4659</EventID><Version>0</Version><Level>0</Level><Task>12800</
Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-0603T19:18:09.951551200Z'/>
<EventRecordID>308888</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='5540'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/
></System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</
Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device\HarddiskVolume8\shar
\event.txt</Data>
<Data Name='HandleId'>0x0</Data><Data
 Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data>
<Data Name='AccessList'>%%1537
    %%4423
    </Data><Data Name='AccessMask'>0x10080</Data><Data Name='PrivilegeList'>-</Data>
<Data Name='ProcessId'>0x4</Data></EventData></Event>
```

Event ID 4663 is logged when a specific operation was performed on the object. The following example shows reading data from a file, which can be interpreted from `AccessList %%4416`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4663< /EventID><Version>1</Version><Level>0</Level><Task>12800</
Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:10:13.887145400Z'/>
<EventRecordID>308831</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='6916'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/
></System>
<EventData>< Data
 Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113< /Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device
\HarddiskVolume8\share\event.txt</Data>
<Data Name='HandleId'>0x101c</Data><Data Name='AccessList'>%%4416
    </Data>
```

```
<Data Name='AccessMask'>0x1</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data><Data Name='ResourceAttributes'>S:AI</Data>
</EventData></Event>
```

The following example shows write/append data from a file, which can be interpreted from `AccessList %%4417`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4663</EventID><Version>1</Version><Level>0</Level><Task>12800</
Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:12:16.813827100Z'/>
<EventRecordID>308838</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='5828'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/
></System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</
Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device
\HarddiskVolume8\share\event.txt</Data>
<Data Name='HandleId'>0xa38</Data><Data Name='AccessList'>%%4417
    </Data><Data Name='AccessMask'>0x2</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data><Data Name='ResourceAttributes'>S:AI</Data></
EventData></Event>
```

Event ID 4656 indicates that a specific access was requested for an object. In the following example, the Read request was initiated to ObjectName "permtest" and was a failed attempt, as seen in the Keywords value of `0x8010000000000000`.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4656</EventID><Version>1</Version><Level>0</Level><Task>12800</
Task><Opcode>0</Opcode>
<Keywords>0x8010000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:22:55.113783500Z'/>
<EventRecordID>308919</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='4924'/>
```

```
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/
></System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</
Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device
\HarddiskVolume8\share\permtest</Data>
<Data Name='HandleId'>0x0</Data><Data
 Name='TransactionId'>{00000000-0000-0000-0000-000000000000}</Data>
<Data Name='AccessList'>%%1541
    %%4416
    %%4423
    </Data><Data Name='AccessReason'>%%1541: %%1805
    %%4416: %%1805
    %%4423: %%1811 D:(A;OICI;0x1301bf;;;AU)
    </Data><Data Name='AccessMask'>0x100081</Data><Data Name='PrivilegeList'>-</Data>
<Data Name='RestrictedSidCount'>0</Data><Data Name='ProcessId'>0x4</Data><Data
 Name='ProcessName'></Data>
<Data Name='ResourceAttributes'>-</Data></EventData></Event>
```

Event ID 4670 is logged when permissions for an object are changed. The following example shows that user "admin" modified the permission on ObjectName "permtest" to add permissions to SID "S-1-5-21-658495921-4185342820-3824891517-1113". Refer to Microsoft documentation for more information on how to interpret the permissions.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>4670</EventID><Version>0</Version><Level>0</Level>
<Task>13570</Task><Opcode>0</Opcode><Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime='2021-06-03T19:39:47.537129500Z'/><EventRecordID>308992</
EventRecordID>
<Correlation/><Execution ProcessID='4' ThreadID='2776'/><Channel>Security</Channel>
<Computer>amznfsxgyzohmw8.example.com</Computer><Security/></System><EventData>
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-1113</Data>
<Data Name='SubjectUserName'>Admin</Data><Data Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2a9a603f</Data><Data Name='ObjectServer'>Security</Data>
<Data Name='ObjectType'>File</Data><Data Name='ObjectName'>\Device
\HarddiskVolume8\share\permtest</Data>
<Data Name='HandleId'>0xcc8</Data>
<Data Name='OldSd'>D:PAI(A;OICI;FA;;;SY)
(A;OICI;FA;;;S-1-5-21-658495921-4185342820-3824891517-2622)</Data>
```

```
<Data Name='NewSd'>D:PARAI(A;OICI;FA;;;S-1-5-21-658495921-4185342820-3824891517-1113)
(A;OICI;FA;;;SY)(A;OICI;FA;;;
S-1-5-21-658495921-4185342820-3824891517-2622)</Data><Data Name='ProcessId'>0x4</Data>
<Data Name='ProcessName'></Data></EventData></Event>
```

Event ID 5140 is logged every time a file share is accessed.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>5140</EventID><Version>1</Version><Level>0</Level><Task>12808</
Task><Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords><TimeCreated
 SystemTime='2021-06-03T19:32:07.535208200Z'/>
<EventRecordID>308947</EventRecordID><Correlation/><Execution ProcessID='4'
 ThreadID='3120'/>
<Channel>Security</Channel><Computer>amznfsxgyzohmw8.example.com</Computer><Security/
></System>
<EventData><Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-2620</
Data>
<Data Name='SubjectUserName'>EC2AMAZ-1GP4HMN$</Data><Data
 Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x2d4ca529</Data><Data Name='ObjectType'>File</Data><Data
 Name='IpAddress'>172.45.6.789</Data>
<Data Name='IpPort'>49730</Data><Data Name='ShareName'>\\AMZNFSXCYDKLDZZ\share</Data>
<Data Name='ShareLocalPath'>\??\D:\share</Data><Data Name='AccessMask'>0x1</Data><Data
 Name='AccessList'>%%4416
    </Data></EventData></Event>
```

Event ID 5145 is logged when access is denied at the file share level. The following example shows access to ShareName "demoshare01" was denied.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-
A5BA-3E3B0328C30D}'/>
<EventID>5145</EventID><Version>0</Version><Level>0</Level>
<Task>12811</Task><Opcode>0</Opcode><Keywords>0x8010000000000000</Keywords>
<TimeCreated SystemTime='2021-05-19T22:30:40.485188700Z'/><EventRecordID>282939</
EventRecordID>
<Correlation/><Execution ProcessID='4' ThreadID='344'/><Channel>Security</Channel>
<Computer>amznfsxtmn9autz.example.com</Computer><Security/></System><EventData>
<Data Name='SubjectUserSid'>S-1-5-21-658495921-4185342820-3824891517-
```

```
1113</Data><Data Name='SubjectUserName'>Admin</Data><Data
 Name='SubjectDomainName'>example</Data>
<Data Name='SubjectLogonId'>0x95b3fb7</Data><Data Name='ObjectType'>File</Data>
<Data Name='IpAddress'>172.31.7.112</Data><Data Name='IpPort'>59979</Data>
<Data Name='ShareName'>\\AMZNFSXDPNTE0DC\demoshare01</Data><Data Name='ShareLocalPath'>
\??\D:\demoshare01</Data>
<Data Name='RelativeTargetName'>Desktop.ini</Data><Data Name='AccessMask'>0x120089</
Data>
<Data Name='AccessList'>%%1538 %%1541 %%4416 %%4419 %%4423 </Data><Data
 Name='AccessReason'>%%1538:
%%1804 %%1541: %%1805 %%4416: %%1805 %%4419: %%1805 %%4423: %%1805 </Data></
EventData></Event>
```

If you use CloudWatch Logs Insights to search your log data, you can run queries on the event fields, as shown by the following examples:

- To query for a specific event ID:

  ```
  fields @message
      | filter @message like /4660/
  ```

- To query all events matching a particular file name:

  ```
  fields @message
      | filter @message like /event.txt/
  ```

For more information on the CloudWatch Logs Insights query language, see Analyzing Log Data with CloudWatch Logs Insights, in the *Amazon CloudWatch Logs User Guide*.

Setting file and folder auditing controls

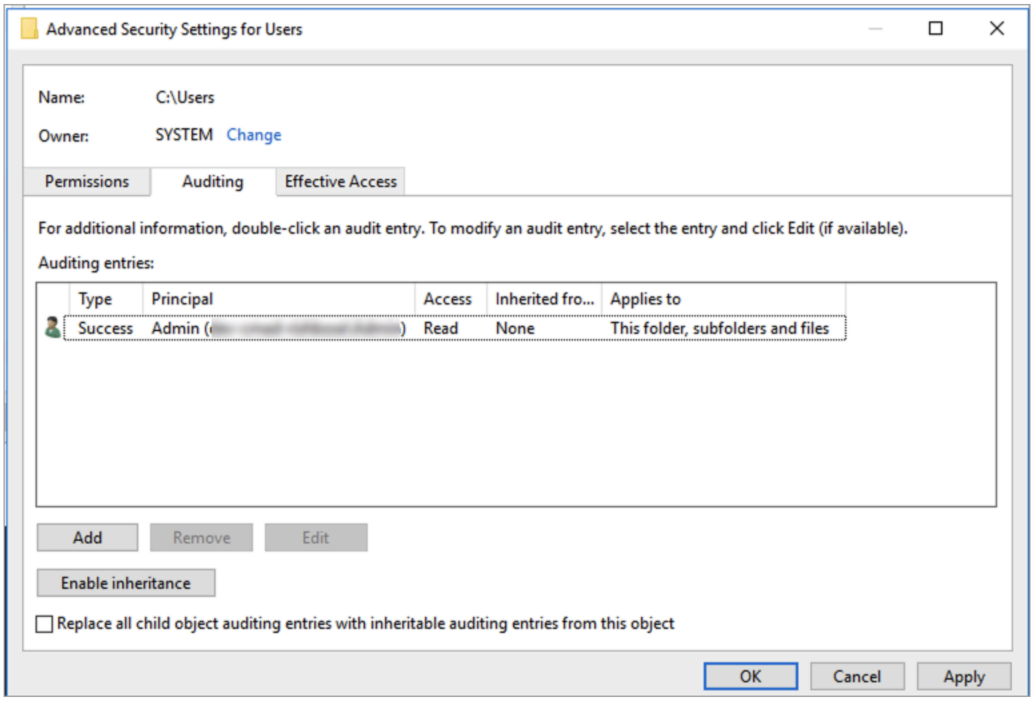
You need to set audit controls on the files and folders that you want audited for user access attempts. Audit controls are also known as NTFS system access control lists (SACLs).

You configure audit controls using the Windows-native GUI interface or programmatically using Windows PowerShell commands. If inheritance is enabled, you typically need to set audit controls only on the top-level folders you want to log accesses for.

Using the Windows GUI to set auditing access

To use a GUI for setting audit controls on your files and folders, use Windows File Explorer. On a given file or folder, open Windows File Explorer and select the **Properties > Security > Advanced > Auditing** tab.

The following audit control example audits successful events for a folder. A Windows event log entry will be emitted whenever that handle is opened for read successfully by the Admin user.



Advanced Security Settings for Users

Name: C:\Users

Owner: SYSTEM Change

For additional information, double-click an audit entry. To modify an audit entry, select the entry and click Edit (if available).

Auditing entries:

Type	Principal	Access	Inherited fro...	Applies to
Success	Admin ([redacted])	Read	None	This folder, subfolders and files

☐ Replace all child object auditing entries with inheritable auditing entries from this object

The **Type** field indicates what actions you want to audit. Set this field to **Success** to audit successful attempts, **Fail** to audit failed attempts, or **All** to audit both successful and failed attempts.

For more information on the auditing entry fields, see Apply a basic audit policy on a file or folder in the Microsoft documentation.

Using PowerShell commands to set auditing access

You can use the Microsoft Windows `Set-Acl` command to set the auditing SACL on any file or folder. For information about this command, see the Microsoft Set-Acl documentation.

Following is an example of using a series of PowerShell commands and variables to set auditing access for successful attempts. You can adapt these example commands to fit the needs on your file system.

```
$path = "C:\Users\TestUser\Desktop\DemoTest\"
```

```
$ACL = Get-Acl $path

$ACL | Format-List

$AuditUser = "TESTDOMAIN\TestUser"

$AuditRules = "FullControl"

$InheritType = "ContainerInherit,ObjectInherit"

$AuditType = "Success"

$AccessRule = New-Object System.Security.AccessControl.FileSystemAuditRule($AuditUser,
$AuditRules,$InheritType,"None",$AuditType)

$ACL.SetAuditRule($AccessRule)

$ACL | Set-Acl $path

Get-Acl $path -Audit | Format-List
```

Managing file access auditing

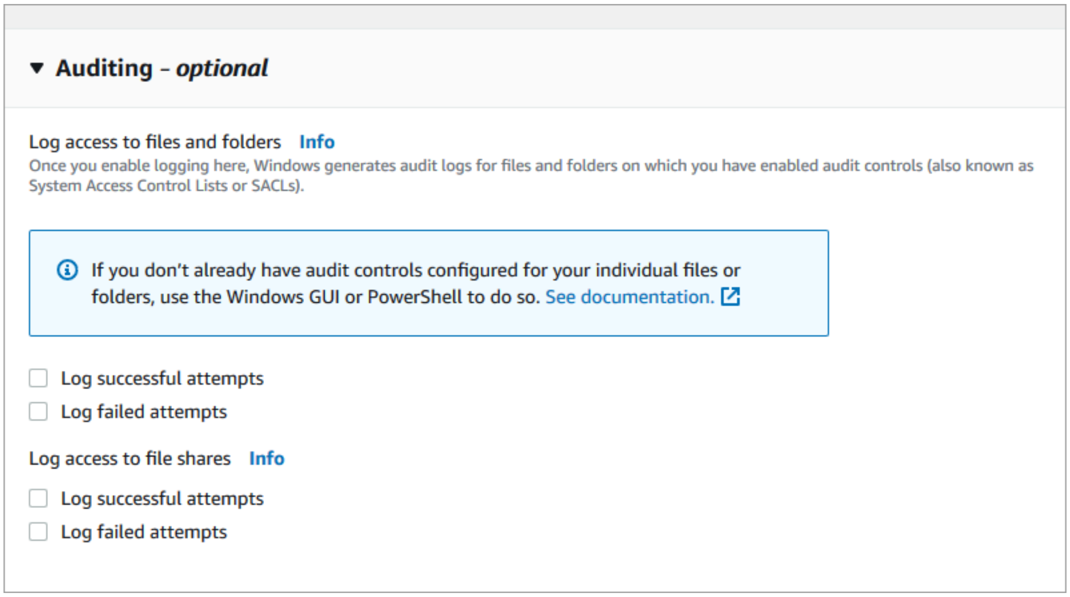
You can enable file access auditing when creating a new Amazon FSx for Windows File Server file system. File access auditing is turned off by default when you create a file system from the Amazon FSx console.

On existing file systems that have file access auditing enabled, you can change the file access auditing settings, including changing the access attempt types for file and file share accesses, and the audit event log destination. You can perform these tasks using the Amazon FSx console, AWS CLI, or API.

> **Note**
>
> File access auditing is supported only on Amazon FSx for Windows File Server file systems with a throughput capacity of 32 MBps or greater. You cannot create or update a file system with a throughput capacity of less than 32 MBps if file access auditing is enabled. You can modify the throughput capacity at any time after you create the file system. For more information, see Managing throughput capacity.

To enable file access auditing when creating a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system described in Step 5. Create your file system in the Getting Started section.
3. Open the **Auditing - optional** section. File access auditing is disabled by default.



▼ **Auditing** - *optional*

Log access to files and folders Info
Once you enable logging here, Windows generates audit logs for files and folders on which you have enabled audit controls (also known as System Access Control Lists or SACLs).

If you don't already have audit controls configured for your individual files or folders, use the Windows GUI or PowerShell to do so. See documentation.

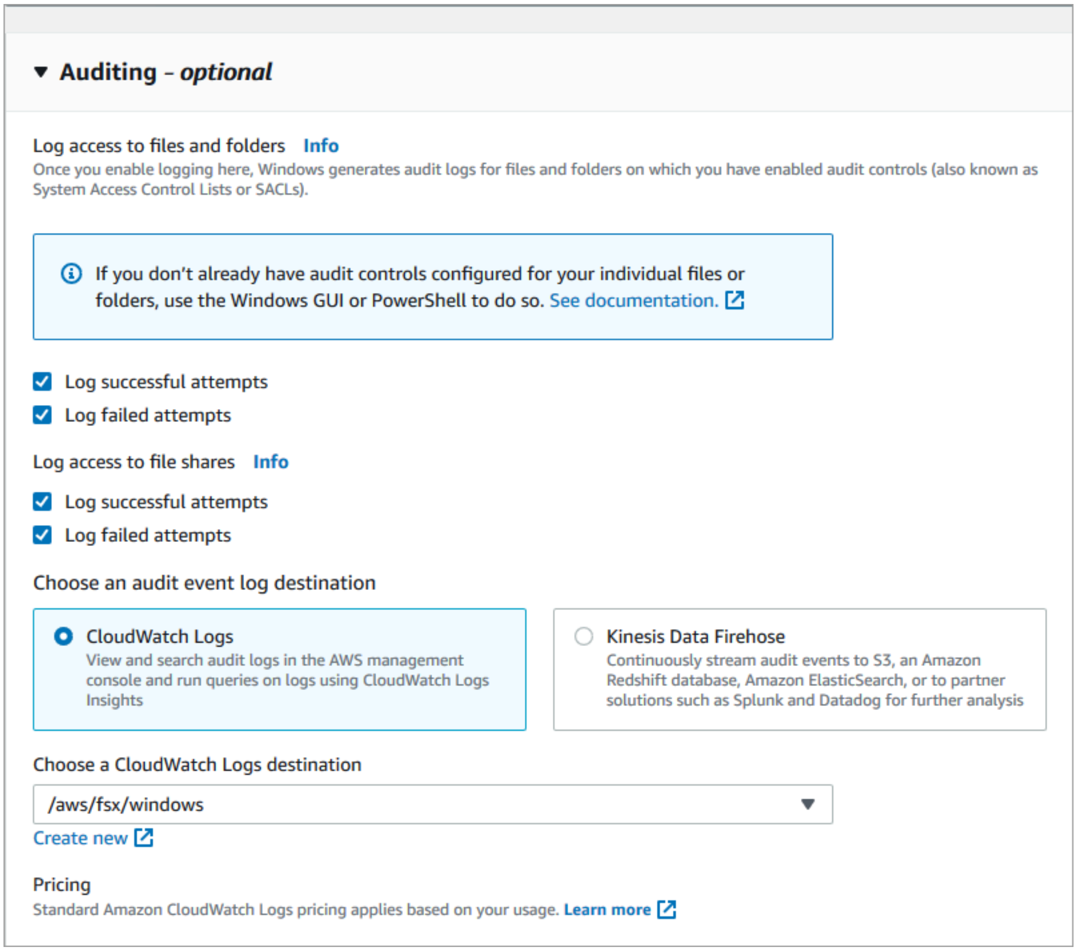
□ Log successful attempts
□ Log failed attempts

Log access to file shares Info

□ Log successful attempts
□ Log failed attempts

4. To enable and configure file access auditing, do the following.
   - For **Log access to files and folders**, select the logging of successful and/or failed attempts. Logging is disabled for files and folders if you don't make a selection.
   - For **Log access to file shares**, select the logging of successful and/or failed attempts. Logging is disabled for file shares if you don't make a selection.
   - For **Choose an audit event log destination**, choose **CloudWatch Logs** or **Firehose**. Then choose an existing log or delivery stream or create a new one. For CloudWatch Logs, Amazon FSx can create and use a default log stream in the CloudWatch Logs `/aws/fsx/windows` log group.

Following is an example of a file access auditing configuration that will audit successful and failed access attempts of end users for files, folders, and file shares. The audit event logs will be sent to the default CloudWatch Logs `/aws/fsx/windows` log group destination.



▼ Auditing - *optional*

Log access to files and folders Info

Once you enable logging here, Windows generates audit logs for files and folders on which you have enabled audit controls (also known as System Access Control Lists or SACLs).

ⓘ If you don't already have audit controls configured for your individual files or folders, use the Windows GUI or PowerShell to do so. See documentation.

☑ Log successful attempts

☑ Log failed attempts

Log access to file shares Info

☑ Log successful attempts

☑ Log failed attempts

Choose an audit event log destination

● CloudWatch Logs
View and search audit logs in the AWS management console and run queries on logs using CloudWatch Logs Insights

○ Kinesis Data Firehose
Continuously stream audit events to S3, an Amazon Redshift database, Amazon ElasticSearch, or to partner solutions such as Splunk and Datadog for further analysis

Choose a CloudWatch Logs destination

/aws/fsx/windows ▼

Create new

Pricing

Standard Amazon CloudWatch Logs pricing applies based on your usage. Learn more

5. Continue with the next section of the file system creation wizard.

When the file system is **Available**, the file access auditing feature is enabled.

To enable file access auditing when creating a file system (CLI)

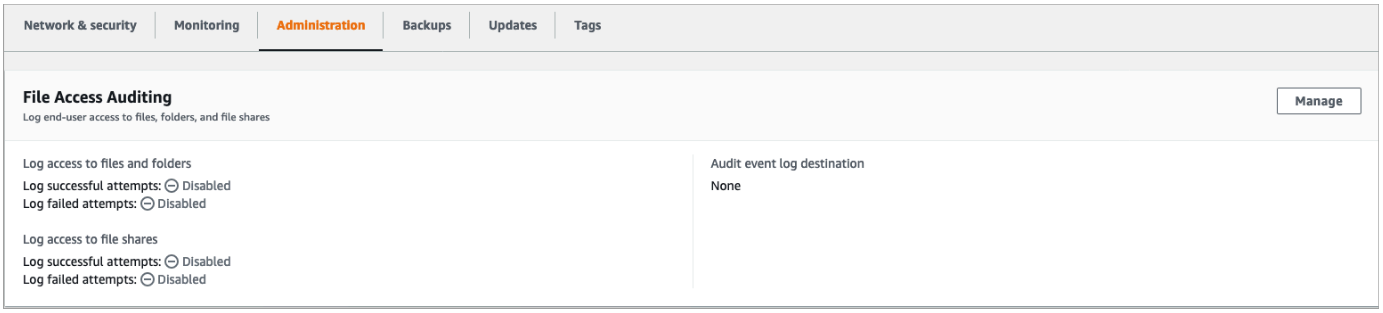
1. When creating a new file system, use the `AuditLogConfiguration` property with the CreateFileSystem API operation to enable file access auditing for the new file system.

```
aws fsx create-file-system \
  --file-system-type WINDOWS \
  --storage-capacity 300 \
  --subnet-ids subnet-123456 \
  --windows-configuration
 AuditLogConfiguration='{FileAccessAuditLogLevel="SUCCESS_AND_FAILURE", \
    FileShareAccessAuditLogLevel="SUCCESS_AND_FAILURE", \
    AuditLogDestination="arn:aws:logs:us-east-1:123456789012:log-group:/aws/fsx/my-
customer-log-group"}'
```

2. When the file system is **Available**, the file access auditing feature is enabled.

To change the file access auditing configuration (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Windows file system that you want to manage file access auditing for.
3. Choose the **Administration** tab.
4. On the **File Access Auditing** panel, choose **Manage**.



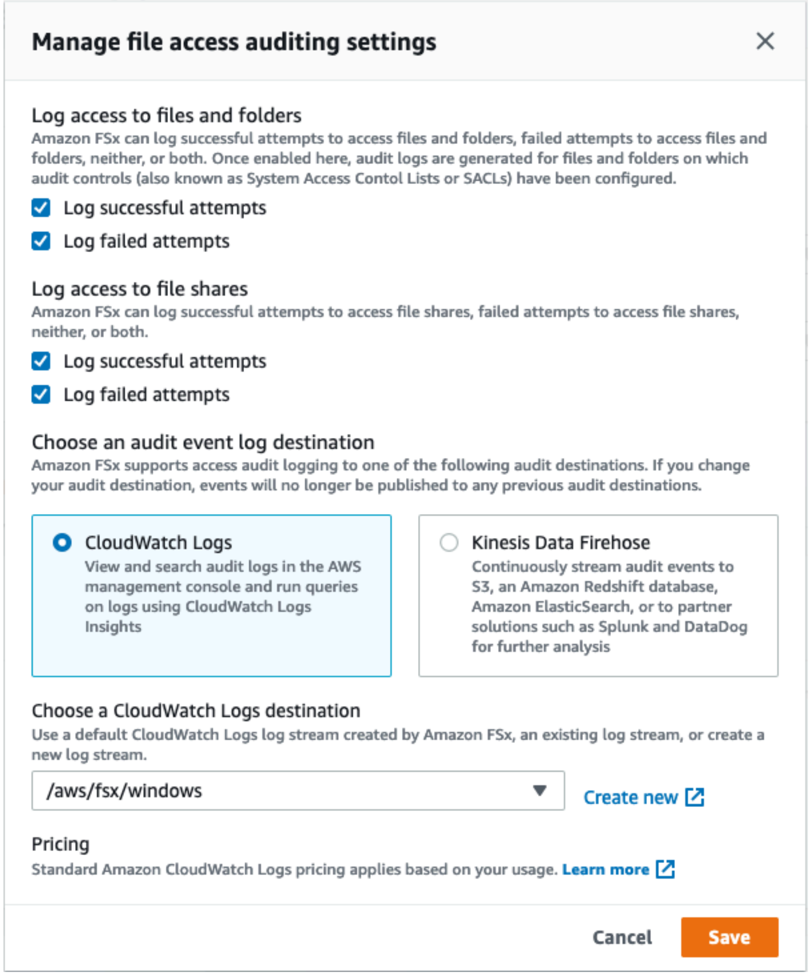
File Access Auditing
Log end-user access to files, folders, and file shares

Log access to files and folders
Log successful attempts: ⊖ Disabled
Log failed attempts: ⊖ Disabled

Log access to file shares
Log successful attempts: ⊖ Disabled
Log failed attempts: ⊖ Disabled

Audit event log destination
None

5. On the **Manage file access auditing settings** dialog, change the desired settings.



Manage file access auditing settings

Log access to files and folders

Amazon FSx can log successful attempts to access files and folders, failed attempts to access files and folders, neither, or both. Once enabled here, audit logs are generated for files and folders on which audit controls (also known as System Access Contol Lists or SACLs) have been configured.

☑ Log successful attempts

☑ Log failed attempts

Log access to file shares

Amazon FSx can log successful attempts to access file shares, failed attempts to access file shares, neither, or both.

☑ Log successful attempts

☑ Log failed attempts

Choose an audit event log destination

Amazon FSx supports access audit logging to one of the following audit destinations. If you change your audit destination, events will no longer be published to any previous audit destinations.

● CloudWatch Logs
View and search audit logs in the AWS management console and run queries on logs using CloudWatch Logs Insights

○ Kinesis Data Firehose
Continuously stream audit events to S3, an Amazon Redshift database, Amazon ElasticSearch, or to partner solutions such as Splunk and DataDog for further analysis

Choose a CloudWatch Logs destination

Use a default CloudWatch Logs log stream created by Amazon FSx, an existing log stream, or create a new log stream.

/aws/fsx/windows

Pricing

Standard Amazon CloudWatch Logs pricing applies based on your usage. Learn more

- For **Log access to files and folders**, select the logging of successful and/or failed attempts. Logging is disabled for files and folders if you don't make a selection.
- For **Log access to file shares**, select the logging of successful and/or failed attempts. Logging is disabled for file shares if you don't make a selection.
- For **Choose an audit event log destination**, choose **CloudWatch Logs** or **Firehose**. Then choose an existing log or delivery stream or create a new one.

6. Choose **Save**.

To change the file access auditing configuration (CLI)

- Use the `update-file-system` CLI command or the equivalent `UpdateFileSystem` API operation.

```
aws fsx update-file-system \
  --file-system-id fs-0123456789abcdef0 \
  --windows-configuration
 AuditLogConfiguration='{FileAccessAuditLogLevel="SUCCESS_ONLY", \
    FileShareAccessAuditLogLevel="FAILURE_ONLY", \
    AuditLogDestination="arn:aws:logs:us-east-1:123456789012:log-group:/aws/fsx/my-
customer-log-group"}'
```

Identity and access management for Amazon FSx for Windows File Server

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

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How Amazon FSx for Windows File Server works with IAM
- Identity-based policy examples for Amazon FSx for Windows File Server
- AWS managed policies for Amazon FSx for Windows File Server
- Troubleshooting Amazon FSx for Windows File Server identity and access
- Using tags with Amazon FSx
- Using service-linked roles for FSx for Windows File Server

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting Amazon FSx for Windows File Server identity and access)
- **Service administrator** - determine user access and submit permission requests (see How Amazon FSx for Windows File Server works with IAM)

- **IAM administrator** - write policies to manage access (see Identity-based policy examples for Amazon FSx for Windows File Server)

Authenticating with identities

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

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

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

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

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

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

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

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries –** Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs) –** Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs) –** Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies –** Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

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

How Amazon FSx for Windows File Server works with IAM

Before you use IAM to manage access to FSx for Windows File Server, learn what IAM features are available to use with FSx for Windows File Server.

IAM features you can use with Amazon FSx for Windows File Server

IAM feature	FSx support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward access sessions	Yes
Service roles	No
Service-linked roles	Yes

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

Identity-based policies for FSx

Supports identity-based policies: Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for FSx

To view examples of FSx for Windows File Server identity-based policies, see Identity-based policy examples for Amazon FSx for Windows File Server.

Resource-based policies within FSx

Supports resource-based policies: No

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

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for FSx

Supports policy actions: Yes

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

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of FSx actions, see Actions defined by Amazon FSx for Windows File Server in the *Service Authorization Reference*.

Policy actions in FSx use the following prefix before the action:

```
fsx
```

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

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

To view examples of FSx for Windows File Server identity-based policies, see Identity-based policy examples for Amazon FSx for Windows File Server.

Policy resources for FSx

Supports policy resources: Yes

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

The Resource JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

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

To see a list of FSx resource types and their ARNs, see Resources defined by Amazon FSx for Windows File Server in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Amazon FSx for Windows File Server.

To view examples of FSx for Windows File Server identity-based policies, see Identity-based policy examples for Amazon FSx for Windows File Server.

Policy condition keys for FSx

Supports service-specific policy condition keys: Yes

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

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of FSx condition keys, see Condition keys for Amazon FSx for Windows File Server in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by Amazon FSx for Windows File Server.

To view examples of FSx for Windows File Server identity-based policies, see Identity-based policy examples for Amazon FSx for Windows File Server.

ACLs in FSx

Supports ACLs: No

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

ABAC with FSx

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

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

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

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

Using temporary credentials with FSx

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for FSx

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for FSx

Supports service roles: No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

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

Service-linked roles for FSx

Supports service-linked roles: Yes

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

For details about creating or managing FSx for Windows File Server service-linked roles, see Using service-linked roles for FSx for Windows File Server.

Identity-based policy examples for Amazon FSx for Windows File Server

By default, users and roles don't have permission to create or modify FSx for Windows File Server resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

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

For details about actions and resource types defined by FSx, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Amazon FSx for Windows File Server in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the FSx console
- Allow users to view their own permissions

Policy best practices

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

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

Using the FSx console

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

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

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

Allow users to view their own permissions

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

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

AWS managed policies for Amazon FSx for Windows File Server

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AmazonFSxServiceRolePolicy

Allows Amazon FSx to manage AWS resources on your behalf. See Using service-linked roles for FSx for Windows File Server to learn more.

AWS managed policy: AmazonFSxDeleteServiceLinkedRoleAccess

You can't attach `AmazonFSxDeleteServiceLinkedRoleAccess` to your IAM entities. This policy is linked to a service and used only with the service-linked role for that service. You cannot attach, detach, modify, or delete this policy. For more information, see Using service-linked roles for FSx for Windows File Server.

This policy grants administrative permissions that allow Amazon FSx to delete its Service Linked Role for Amazon S3 access, used only by Amazon FSx for Lustre.

Permissions details

This policy includes permissions in `iam` to allow Amazon FSx to view, delete, and view the deletion status for the FSx Service Linked Roles for Amazon S3 access.

To view the permissions for this policy, see AmazonFSxDeleteServiceLinkedRoleAccess in the AWS Managed Policy Reference Guide.

AWS managed policy: AmazonFSxFullAccess

You can attach AmazonFSxFullAccess to your IAM entities. Amazon FSx also attaches this policy to a service role that allows Amazon FSx to perform actions on your behalf.

Provides full access to Amazon FSx and access to related AWS services.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals full access to perform all Amazon FSx actions, except for `BypassSnaplockEnterpriseRetention`.
- `ds` – Allows principals to view information about the Directory Service directories.
- `ec2`
  - Allows principals to create tags under the specified conditions.
  - To provide enhanced security group validation of all security groups that can be used with a VPC.
- `iam` – Allows principles to create an Amazon FSx service linked role on the user's behalf. This is required so that Amazon FSx can manage AWS resources on the user's behalf.
- `firehose` – Allows principals to write records to a Amazon Data Firehose. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to Firehose.
- `logs` – Allows principals to create log groups, log streams, and write events to log streams. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to CloudWatch Logs.

To view the permissions for this policy, see AmazonFSxFullAccess in the AWS Managed Policy Reference Guide.

AWS managed policy: AmazonFSxConsoleFullAccess

You can attach the `AmazonFSxConsoleFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow full access to Amazon FSx and access to related AWS services via the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to perform all actions in the Amazon FSx management console, except for `BypassSnaplockEnterpriseRetention`.
- `cloudwatch` – Allows principals to view CloudWatch Alarms and metrics in the Amazon FSx management console.
- `ds` – Allows principals to list information about an Directory Service directory.
- `ec2`
  - Allows principals to create tags on route tables, list network interfaces, route tables, security groups, subnets and the VPC associated with an Amazon FSx file system.
  - Allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.
  - Allows principals to view the elastic network interfaces associated with an Amazon FSx file system.
- `kms` – Allows principals to list aliases for AWS Key Management Service keys.
- `s3` – Allows principals to list some or all of the objects in an Amazon S3 bucket (up to 1000).
- `secretsmanager` – Allows principals to list secrets in AWS Secrets Manager for selecting domain join service account credentials.
- `iam` – Grants permission to create a service linked role that allows Amazon FSx to perform actions on the user's behalf.

To view the permissions for this policy, see AmazonFSxConsoleFullAccess in the AWS Managed Policy Reference Guide.

AWS managed policy: AmazonFSxConsoleReadOnlyAccess

You can attach the `AmazonFSxConsoleReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions to Amazon FSx and related AWS services so that users can view information about these services in the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to view information about Amazon FSx file systems, including all tags, in the Amazon FSx Management Console.
- `cloudwatch` – Allows principals to view CloudWatch Alarms and metrics in the Amazon FSx Management Console.
- `ds` – Allows principals to view information about an Directory Service directory in the Amazon FSx Management Console.
- `ec2`
  - Allows principals to view network interfaces, security groups, subnets and the VPC associated with an Amazon FSx file system in the Amazon FSx Management Console.
  - Allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.
  - Allows principals to view the elastic network interfaces associated with an Amazon FSx file system.
- `kms` – Allows principals to view aliases for AWS Key Management Service keys in the Amazon FSx Management Console.
- `log` – Allows principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.
- `secretsmanager` – Allows principals to list secrets in AWS Secrets Manager for selecting domain join service account credentials.
- `firehose` – Allows principals to describe the Amazon Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.

To view the permissions for this policy, see AmazonFSxConsoleReadOnlyAccess in the AWS Managed Policy Reference Guide.

AWS managed policy: AmazonFSxReadOnlyAccess

You can attach the `AmazonFSxReadOnlyAccess` policy to your IAM identities.

- `fsx` – Allows principals to view information about Amazon FSx file systems, including all tags, in the Amazon FSx Management Console.

- ec2 – To provide enhanced security group validation of all security groups that can be used with a VPC.

To view the permissions for this policy, see AmazonFSxReadOnlyAccess in the AWS Managed Policy Reference Guide.

Amazon FSx updates to AWS managed policies

View details about updates to AWS managed policies for Amazon FSx since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon FSx Document history page.

Change	Description	Date
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added a new permission, secretsmanager:ListSecrets that allows principals to list secrets in AWS Secrets Manager for selecting domain join service account credentials.	November 5, 2025
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added a new permission, secretsmanager:ListSecrets that allows principals to list secrets in AWS Secrets Manager for selecting domain join service account credentials.	November 3, 2025
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added a new permission, ec2:AssignIpv6Addresses that allows principals to assign IPv6 addresses to customer network interfaces that have	July 22, 2025

Change	Description	Date
	an `AmazonFSx.FileSystemId` tag.	
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added a new permission, `ec2:UnassignIpv6Addresses` that allows principals to unassign IPv6 addresses from customer network interfaces that have an `AmazonFSx.FileSystemId` tag.	July 22, 2025
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:CreateAndAttachS3AccessPoint` that allows principals to create an S3 access point and attach it to an FSx volume.	June 25, 2025
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:DescribeS3AccessPointAttachments` that allows principals to list all S3 access points in an AWS account in an AWS Region.	June 25, 2025
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:DetachAndDeleteS3AccessPoint` that allows principals to delete an S3 access point.	June 25, 2025

Change	Description	Date
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:CreateAndAttachS3AccessPoint` that allows principals to create an S3 access point and attach it to an FSx volume.	June 25, 2025
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:DescribeS3AccessPointAttachments` that allows principals to list all S3 access points in an AWS account in an AWS Region.	June 25, 2025
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added a new permission, `fsx:DetachAndDeleteS3AccessPoint` that allows principals to delete an S3 access point.	June 25, 2025
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:DescribeNetworkInterfaces` that allows principals to view the elastic network interfaces associated with their file system.	February 25, 2025

Change	Description	Date
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:DescribeNetworkInterfaces` that allows principals to view the elastic network interfaces associated with their file system.	February 07, 2025
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSecurityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024
AmazonFSxReadOnlyAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSecurityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSecurityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024

Change	Description	Date
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSecurityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSecurityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permission to enable users to perform cross-region and cross-account data replication for FSx for OpenZFS file systems.	December 20, 2023
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permission to enable users to perform cross-region and cross-account data replication for FSx for OpenZFS file systems.	December 20, 2023

Change	Description	Date
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permission to enable users to perform on-demand replication of volumes for FSx for OpenZFS file systems.	November 26, 2023
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permission to enable users to perform on-demand replication of volumes for FSx for OpenZFS file systems.	November 26, 2023
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permissions to enable users to view, enable, and disable shared VPC support for FSx for ONTAP Multi-AZ file systems.	November 14, 2023
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to enable users to view, enable, and disable shared VPC support for FSx for ONTAP Multi-AZ file systems.	November 14, 2023
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to manage network configurations for FSx for OpenZFS Multi-AZ file systems.	August 9, 2023

Change	Description	Date
AWS managed policy: AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx modified the existing cloudwatch:PutMetricData permission so that Amazon FSx publishes CloudWatch metrics to the AWS/FSx namespace.	July 24, 2023
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx updated the policy to remove the fsx:* permission and add specific fsx actions.	July 13, 2023
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx updated the policy to remove the fsx:* permission and add specific fsx actions.	July 13, 2023
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added new permissions to enable users to view enhanced performance metrics and recommended actions for FSx for Windows File Server file systems in the Amazon FSx console.	September 21, 2022
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to enable users to view enhanced performance metrics and recommended actions for FSx for Windows File Server file systems in the Amazon FSx console.	September 21, 2022

Change	Description	Date
AmazonFSxReadOnlyAccess – Started tracking policy	This policy grants read-only access to all Amazon FSx resources and any tags associated with them.	February 4, 2022
AmazonFSxDeleteServiceLinkedRoleAccess – Started tracking policy	This policy grants administrative permissions that allow Amazon FSx to delete its Service Linked Role for Amazon S3 access.	January 7, 2022
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to manage network configurations for Amazon FSx for NetApp ONTAP file systems.	September 2, 2021
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to create Amazon FSx for NetApp ONTAP Multi-AZ file systems.	September 2, 2021
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021

Change	Description	Date
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to CloudWatch Logs log streams. This is required so that users can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to Amazon Data Firehose delivery streams. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Data Firehose.	June 8, 2021
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow principal s to describe and create CloudWatch Logs log groups, log streams, and write events to log streams. This is required so that principals can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021

Change	Description	Date
AmazonFSxFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow principal s to describe and write records to a Amazon Data Firehose. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Data Firehose.	June 8, 2021
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow principal s to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can choose an existing CloudWatch Logs log group when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021

Change	Description	Date
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Data Firehose delivery streams associated with the account making the request. This is required so that principals can choose an existing Firehose delivery stream when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021

Change	Description	Date
AmazonFSxConsoleReadOnlyAccess – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021
Amazon FSx started tracking changes	Amazon FSx started tracking changes for its AWS managed policies.	June 8, 2021

Troubleshooting Amazon FSx for Windows File Server identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with FSx for Windows File Server and IAM.

Topics

- I am not authorized to perform an action in FSx
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my FSx resources

I am not authorized to perform an action in FSx

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `fsx:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 fsx:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `fsx:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to FSx for Windows File Server.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in FSx for Windows File Server. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my FSx resources

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

To learn more, consult the following:

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

Using tags with Amazon FSx

You can use tags to control access to Amazon FSx resources and to implement attribute-based access control (ABAC). Users need to have permission to apply tags to Amazon FSx resources during creation.

Grant permission to tag resources during creation

Some resource-creating FSx for Windows File Server API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based access control (ABAC). For more information, see What is ABAC for AWS in the *IAM User Guide*.

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `fsx:CreateFileSystem` or `fsx:CreateBackup`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `fsx:TagResource` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `fsx:TagResource` action.

The following example demonstrates a policy that allows users to create file systems and apply tags to file systems during creation in a specific AWS account.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateFileSystem",
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/*"
    }
  ]
}
```

Similarly, the following policy allows users to create backups on a specific file system and apply any tags to the backup during backup creation.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateBackup"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/file-system-id*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:backup/*"
    }
  ]
}
```

The `fsx:TagResource` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `fsx:TagResource` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `fsx:TagResource` action.

For more information about tagging Amazon FSx resources, see Tagging your Amazon FSx resources. For more information about using tags to control access to FSx resources, see Using tags to control access to your Amazon FSx resources.

Using tags to control access to your Amazon FSx resources

To control access to Amazon FSx resources and actions, you can use AWS Identity and Access Management (IAM) policies based on tags. You can provide the control in two ways:

1. Control access to Amazon FSx resources based on the tags on those resources.
2. Control what tags can be passed in an IAM request condition.

For information about how to use tags to control access to AWS resources, see Controlling access using tags in the *IAM User Guide*. For more information about tagging Amazon FSx resources at creation, see Grant permission to tag resources during creation. For more information about tagging resources, see Tagging your Amazon FSx resources.

Controlling access based on tags on a resource

To control what actions a user or role can perform on an Amazon FSx resource, you can use tags on the resource. For example, you might want to allow or deny specific API operations on a file system resource based on the key-value pair of the tag on the resource.

Example policy – Create a file system on when providing a specific tag

This policy allows the user to create a file system only when they tag it with a specific tag key value pair, in this example, `key=Department, value=Finance`.

```
{
    "Effect": "Allow",
    "Action": [
        "fsx:CreateFileSystem",
        "fsx:TagResource"
    ],
    "Resource": "arn:aws:fsx:region:account-id:file-system/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Department": "Finance"
        }
    }
}
```

Example policy – Create backups only of Amazon FSx file systems with a specific tag

This policy allows users to create backups only of file systems that are tagged with the key value pair `key=Department, value=Finance`, and the backup will be created with the tag `Deparment=Finance`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource",
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Example policy – Create a file system with a specific tag from backups with a specific tag

This policy allows users to create file systems that are tagged with `Department=Finance` only from backups that are tagged with `Department=Finance`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Example policy – Delete file systems with specific tags

This policy allows a user to delete only file systems that are tagged with `Department=Finance`. If they create a final backup, then it must be tagged with `Department=Finance`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:DeleteFileSystem"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Using service-linked roles for FSx for Windows File Server

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 FSx for

Windows File Server. Service-linked roles are predefined by FSx for Windows File Server and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up FSx for Windows File Server easier because you don’t have to manually add the necessary permissions. FSx for Windows File Server defines the permissions of its service-linked roles, and unless defined otherwise, only FSx for Windows File Server 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 FSx for Windows File Server 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 FSx for Windows File Server

FSx for Windows File Server 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 FSx for Windows File Server to complete the following actions on the all applicable AWS resources:

You can't attach AmazonFSxServiceRolePolicy to your IAM entities. This policy is attached to a service-linked role that allows FSx to manage AWS resources on your behalf. For more information, see Using service-linked roles for FSx for Windows File Server.

For updates to this policy, see AmazonFSxServiceRolePolicy.

This policy grants administrative permissions that allows FSx to manage AWS resources on the user's behalf.

Permissions details

The AmazonFSxServiceRolePolicy role permissions are defined by the AmazonFSxServiceRolePolicy AWS managed policy. AmazonFSxServiceRolePolicy has the following permissions:

Note

AmazonFSxServiceRolePolicy is used by all Amazon FSx file system types; some of the listed permissions may not applicable to FSx for Windows.

- `ds` – Allows FSx to view, authorize, and unauthorize applications in your Directory Service directory.
- `ec2` – Allows FSx to do the following:
  - View, create, and disassociate network interfaces associated with an Amazon FSx file system.
  - View one or more Elastic IP addresses associated with an Amazon FSx file system.
  - View Amazon VPCs, security groups, and subnets associated with an Amazon FSx file system.
  - Assign IPv6 addresses to customer network interfaces that have an `AmazonFSx.FileSystemId` tag.
  - Unassign IPv6 addresses from customer network interfaces that have an `AmazonFSx.FileSystemId` tag.
  - To provide enhanced security group validation of all security groups that can be used with a VPC.
  - Create a permission for an AWS-authorized user to perform certain operations on a network interface.
- `cloudwatch` – Allows FSx to publish metric data points to CloudWatch under the AWS/FSx namespace.
- `route53` – Allows FSx to associate an Amazon VPC with a private hosted zone.
- `logs` – Allows FSx to describe and write to CloudWatch Logs log streams. This is so that users can send file access audit logs for an FSx for Windows File Server file system to a CloudWatch Logs stream.
- `firehose` – Allows FSx to describe and write to Amazon Data Firehose delivery streams. This is so that users can publish the file access audit logs for an FSx for Windows File Server file system to an Amazon Data Firehose delivery stream.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CreateFileSystem",
            "Effect": "Allow",
            "Action": [
                "ds:AuthorizeApplication",
                "ds:GetAuthorizedApplicationDetails",
                "ds:UnauthorizeApplication",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeAddresses",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVPCs",
                "ec2:DisassociateAddress",
                "ec2:GetSecurityGroupsForVpc",
                "route53:AssociateVPCWithHostedZone"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PutMetrics",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/FSx"
                }
            }
        },

        {
            "Sid": "TagResourceNetworkInterface",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "AmazonFSx.FileSystemId"
                }
            }
        },
        {
            "Sid": "ManageNetworkInterface",
            "Effect": "Allow",
            "Action": [
                "ec2:AssignPrivateIpAddresses",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:UnassignPrivateIpAddresses"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonFSx.FileSystemId": "false"
                }
            }
        },
        {
            "Sid": "ManageRouteTable",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:ReplaceRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/AmazonFSx": "ManagedByAmazonFSx"
                }
            }
        },
        {
            "Sid": "PutCloudWatchLogs",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/fsx/*"
        },
        {
            "Sid": "ManageAuditLogs",
            "Effect": "Allow",
            "Action": [
                "firehose:DescribeDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/aws-fsx-*"
        }
    ]
}
```

Any updates to this policy are described in Amazon FSx updates to AWS managed policies.

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 FSx for Windows File Server

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, FSx for Windows File Server 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, FSx for Windows File Server creates the service-linked role for you again.

Editing a service-linked role for FSx for Windows File Server

FSx for Windows File Server does not allow you to edit the 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 FSx for Windows File Server

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 FSx for Windows File Server 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 service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported regions for FSx for Windows File Server service-linked roles

FSx for Windows File Server supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

Compliance Validation for Amazon FSx for Windows File Server

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Amazon FSx for Windows File Server and interface VPC endpoints

You can improve the security posture of your VPC by configuring Amazon FSx to use an interface VPC endpoint. Interface VPC endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Amazon FSx APIs without an internet gateway, NAT device, VPN connection, or Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon FSx APIs. Traffic between your VPC and Amazon FSx does not leave the AWS network.

Each interface VPC endpoint is represented by one or more elastic network interfaces in your subnets. A network interface provides a private IP address that serves as an entry point for traffic to the Amazon FSx API. Amazon FSx supports VPC endpoints configured with IPv4-only and dual-stack (IPv4 and IPv6) IP address types. For more information, see Creating an interface VPC endpoint in the *Amazon VPC User Guide*.

Considerations for Amazon FSx interface VPC endpoints

Before you set up an interface VPC endpoint for Amazon FSx, be sure to review Interface VPC endpoint properties and limitations in the *Amazon VPC User Guide*.

You can call any of the Amazon FSx API operations from your VPC. For example, you can create an FSx for Windows File Server file system by calling the CreateFileSystem API from within your VPC. For the full list of Amazon FSx APIs, see Actions in the Amazon FSx API Reference.

VPC peering considerations

You can connect other VPCs to the VPC with interface VPC endpoints using VPC peering. VPC peering is a networking connection between two VPCs. You can establish a VPC peering connection between your own two VPCs, or with a VPC in another AWS account. The VPCs can also be in two different AWS Regions.

Traffic between peered VPCs stays on the AWS network and does not traverse the public internet. Once VPCs are peered, resources like Amazon Elastic Compute Cloud (Amazon EC2) instances in both VPCs can access the Amazon FSx API through interface VPC endpoints created in the one of the VPCs.

Creating an interface VPC endpoint for Amazon FSx API

You can create a VPC endpoint for the Amazon FSx API using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface VPC endpoint in the *Amazon VPC User Guide*.

To create an interface VPC endpoint for Amazon FSx, use one of the following:

- **com.amazonaws.*region*.fsx** – Creates an endpoint for Amazon FSx API operations.
- **com.amazonaws.*region*.fsx-fips** – Creates an endpoint for the Amazon FSx API that complies with Federal Information Processing Standard (FIPS) 140-2.

To use the private DNS option, you must set the enableDnsHostnames and enableDnsSupport attributes of your VPC. For more information, see Viewing and updating DNS support for your VPC in the *Amazon VPC User Guide*.

Excluding AWS Regions in China, if you enable private DNS for the endpoint, you can make API requests to Amazon FSx with the VPC endpoint using its default DNS name for the AWS Region, for example fsx.us-east-1.amazonaws.com. For the China (Beijing) and China (Ningxia) AWS Regions, you can make API requests with the VPC endpoint using fsx-api.cn-north-1.amazonaws.com.cn and fsx-api.cn-northwest-1.amazonaws.com.cn, respectively.

For more information, see Accessing a service through an interface VPC endpoint in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Amazon FSx

To further control access to the Amazon FSx API, you can optionally attach an AWS Identity and Access Management (IAM) policy to your VPC endpoint. The policy specifies the following:

- The principal that can perform actions.
- The actions that can be performed.
- The resources upon which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Working with other services

In addition to Amazon CloudWatch, AWS Identity and Access Management, AWS CloudTrail, and AWS DataSync, FSx for Windows File Server also integrates with the following AWS services:

- Amazon WorkSpaces Applications – WorkSpaces Applicationsis a fully managed application streaming service that provides users with instant access to their desktop applications from anywhere. WorkSpaces Applications manages the AWS resources required to host and run your applications, scales automatically, and provides access to your users on demand. Learn how to create persistent storage for individual users, and share storage across many users on your FSx for Windows File Server file systems using WorkSpaces Applications. For more information, see Using Amazon FSx with Amazon WorkSpaces Applications.
- Amazon Kendra – Amazon Kendra is an intelligent search service that uses natural language processing and advanced machine learning algorithms to return specific answers to search questions from your data. With Amazon Kendra, you can create a unified search experience by connecting multiple data repositories to an index and ingesting and crawling documents. For more information on using Amazon Kendra with FSx for Windows File Server, see Using FSx for Windows File Server with Amazon Kendra.

Topics

- Using Amazon FSx with Amazon WorkSpaces Applications
- Using FSx for Windows File Server with Amazon Kendra

Using Amazon FSx with Amazon WorkSpaces Applications

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 WorkSpaces Applications instances. WorkSpaces Applications is a fully managed application streaming service. You centrally manage your desktop applications on WorkSpaces Applications and securely deliver them to a browser on any computer. For more information on WorkSpaces Applications, see the *Amazon WorkSpaces Applications Administration Guide*. For instructions on how you can streamline the management of your Amazon WorkSpaces Applications images and fleets, see the AWS blog post Automatically create customized AppStream 2.0 Windows images.

The following procedures show you how to use Amazon FSx with WorkSpaces Applications to provide personal persistent storage to each user, and to provide a shared folder so that multiple users can 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 WorkSpaces Applications 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 for Windows File Server.
2. After the file system is available, create a folder for every domain WorkSpaces Applications 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 Creating, updating, removing file shares.

To launch a domain-joined WorkSpaces Applications image builder

1. Sign into the WorkSpaces Applications 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 WorkSpaces Applications in the *Amazon WorkSpaces Applications Administration Guide*.
3. Choose **Images**, **Image Builder**, and launch a new image builder.
4. Choose the directory config object created earlier in the image builder launch wizard to join the image builder to your Active Directory domain.
5. Launch the image builder in the same VPC as that of your Amazon FSx file system. Make sure to associate the image builder with the same AWS Managed Microsoft AD directory to which your Amazon FSx file system is joined. The VPC security groups that you associate with the image builder must allow access to your Amazon FSx file system.

6. Once the image builder is available, connect to the image builder and login using your domain administrator account.
7. Install your applications.

To link Amazon FSx file shares with WorkSpaces Applications

1. In the image builder, create a batch script with the following command and store it in a known file location (for example: C:\Scripts\map-fs.bat). The following example uses S: as the drive letter to map the shared folder on your Amazon FSx file system. You use the DNS name of your Amazon FSx file system or a DNS alias associated with the file system in this script, which you can get from the file system details view in the Amazon FSx console.

   If you're using the file system's DNS name:

   ```
   @echo off
   net use S: /delete
   net use S: \\file-system-DNS-name\users\%username%
   ```

   If you're using a DNS alias associated with the file system:

   ```
   @echo off
   net use S: /delete
   net use S: \\fqdn-DNS-alias\users\%username%
   ```
2. Open a PowerShell prompt and run `gpedit.msc`.
3. From **User Configuration** choose **Windows Settings** and then **Logon**.
4. Navigate to the batch script that you created in the first step of this procedure, and choose it.
5. From **Computer Configuration**, choose **Windows Administrative Templates**, **System**, and then **Group Policy**.
6. Choose the policy **Configure Logon Script delay**. Enable the policy and reduce the time delay to `0`. This setting helps to ensure that the user logon script is executed immediately when the user starts a streaming session.
7. Create your image and assign it to an WorkSpaces Applications fleet. Ensure that you also join the WorkSpaces Applications 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 WorkSpaces Applications 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 for Windows File Server.
2. Every Amazon FSx file system includes a shared folder by default that you can access using the address \\`file-system-DNS-name`\share, or \\`fqdn-DNS-alias`\share if you are using DNS aliases. You can use the default share or create a different shared folder. For more information, see Creating, updating, removing file shares.

To launch an WorkSpaces Applications image builder

1. From the WorkSpaces Applications 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 WorkSpaces Applications

1. Create a batch script, as described in the previous procedure, to automatically mount the shared folder whenever a user launches a streaming session. To complete the script, you need the file system's DNS name or a DNS alias that is associated with the file system (which you

can obtain from the file system details view in the Amazon FSx Console), and credentials for accessing the shared folder.

If you're using the file system's DNS name:

```
@echo off
net use S: /delete
net use S: \\file-system-DNS-name\share /user:username password
```

If you're using a DNS alias associated with the file system:

```
@echo off
net use S: /delete
net use S: \\fqdn-DNS-alias\share /user:username password
```

2. Create a Group Policy to execute this batch script at every user logon. You can follow the same instructions as described in the previous section.
3. Create your image and assign it to your fleet.
4. Launch a streaming session. You should now see the shared folder automatically mapped to the drive letter.

Using FSx for Windows File Server with Amazon Kendra

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

- For more information about Amazon Kendra, see What is Amazon Kendra in the *Amazon Kendra Developer's Guide*.
- For more information about how to add your file system as an Amazon Kendra data source, see Getting started with an Amazon FSx data source (console) in the *Amazon Kendra Developer's Guide*.
- For overview information about Amazon Kendra, see the Amazon Kendra website.

- For a walkthrough of how to search your file system using Amazon Kendra, see Securely search unstructured data on Windows file systems with the Amazon Kendra connector for Amazon FSx for Windows File Server on the *AWS Machine Learning Blog*.

File system performance

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

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

Quotas

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

Topics

- Quotas that you can increase
- Resource quotas for each file system
- Additional considerations
- Quotas specific to Microsoft Windows

Quotas that you can increase

Following are the quotas for Amazon FSx for Windows File Server for each AWS account, per AWS Region, that you can increase.

Resource	Default	Description
Windows file systems	100	The maximum number of Amazon FSx for Windows Server file systems that you can create in this account.
Windows throughput capacity	10240	The total amount of throughput capacity (in MBps) allowed for all Amazon FSx for Windows file systems in this account.
Windows HDD storage capacity	524288	The maximum amount of HDD storage capacity (in GiB) allowed for all Amazon FSx for Windows File Server file systems in this account.
Windows SSD storage capacity	524288	The maximum amount of SSD storage capacity (in GiB)

Resource	Default	Description
		allowed for all Amazon FSx for Windows File Server file systems in this account.
Windows total SSD IOPS	500,000	The total amount of SSD IOPS allowed for all Amazon FSx for Windows File Server file systems in this account.
Windows backups	500	The maximum number of user-initiated backups for all Amazon FSx for Windows File Server file systems that you can have in this account.

To request a quota increase

1. Open the Service Quotas console.
2. In the navigation pane, choose **AWS services**.
3. Choose **Amazon FSx**.
4. Choose a quota.
5. Choose **Request quota increase**, and follow the directions to request a quota increase.
6. To view the status of the quota request, choose **Quota request history** in the console navigation pane.

For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Resource quotas for each file system

Following are the quotas on Amazon FSx for Windows File Server resources for each file system in an AWS Region.

Resource	Limit per file system
Maximum number of tags	50
Maximum retention period for automated backups	90 days
Maximum number of backup copy requests in progress to a single destination Region per account.	5
Minimum storage capacity, SSD file systems	32 GiB
Minimum storage capacity, HDD file systems	2,000 GiB
Maximum storage capacity, SSD and HDD	64 TiB
Minimum SSD IOPS	96
Maximum SSD IOPS	400,000
Minimum throughput capacity	8 MBps
Maximum throughput capacity	12,288 MBps
Maximum number of file shares	100,000

Additional considerations

In addition, note the following:

- You can use each AWS Key Management Service (AWS KMS) key on up to 125 Amazon FSx file systems.
- For a list of AWS Regions where you can create file systems, see Amazon FSx Endpoints and Quotas in the *AWS General Reference*.
- You map your file shares from Amazon EC2 instances in your virtual private cloud (VPC) with their Domain Name Service (DNS) names.

Quotas specific to Microsoft Windows

For more information, see NTFS limits on the Microsoft Windows Dev Center.

Troubleshooting Amazon FSx

Use the following sections to help troubleshoot problems you have with Amazon FSx.

If you encounter problems not listed following while using Amazon FSx, try asking a question in the Amazon FSx forum.

Topics

- You can't access your file system
- Creating a new Amazon FSx file system fails
- File system is in a misconfigured state
- You can't configure DFS-R on a Multi-AZ or Single-AZ 2 file system
- Storage or throughput capacity updates fail

You can't access your file system

There are a number of potential causes for being unable to access your file system, each with their own resolution, as follows.

Topics

- The file system elastic network interface was modified or deleted
- The Elastic IP address attached to the file system elastic network interface was deleted
- The file system security group lacks the required inbound or outbound rules.
- The compute instance's security group lacks the required outbound rules
- Compute instance not joined to an Active Directory
- The file share doesn't exist
- Active Directory user lacks required permissions
- Allow Full control NTFS ACL permissions removed
- Can't access a file system using an on-premises client
- New file system is not registered in DNS
- Can't access the file system using a DNS alias
- Can't access the file system using an IP address

The file system elastic network interface was modified or deleted

You must not modify or delete the file system's elastic network interface. Modifying or deleting the network interface can cause a permanent loss of connection between your VPC and your file system. Create a new file system, and do not modify or delete the Amazon FSx elastic network interface. For more information, see File system access control with Amazon VPC.

The Elastic IP address attached to the file system elastic network interface was deleted

Amazon FSx doesn't support accessing file systems from the public internet. Amazon FSx automatically detaches any Elastic IP address, which is a public IP address reachable from the internet, that gets attached to a file system's elastic network interface. For more information, see Accessing your data.

The file system security group lacks the required inbound or outbound rules.

Review the inbound rules specified in Amazon VPC Security Groups, and make sure that the security group associated with your file system has the corresponding inbound rules.

The compute instance's security group lacks the required outbound rules

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

Compute instance not joined to an Active Directory

Your compute instances might not be correctly joined to one of two types of Active Directory:

- The AWS Managed Microsoft AD directory to which your file system is joined.
- A Microsoft Active Directory directory that has a one-way forest trust relationship established with the AWS Managed Microsoft AD directory.

Make sure that your compute instances are joined to one of two types of directory. One type is the AWS Managed Microsoft AD directory to which your file system is joined. The other type is a

Microsoft Active Directory directory that has a one-way forest trust relationship established with the AWS Managed Microsoft AD directory. For more information, see Using Amazon FSx with AWS Directory Service for Microsoft Active Directory.

The file share doesn't exist

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

If you're using an existing file share, make sure that the file system DNS name and the share name are correctly specified. To manage your file shares, see Creating, updating, removing file shares.

Active Directory user lacks required permissions

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

Make sure that the access permissions for the file share and Windows access control lists (ACLs) for the shared folder allow access to the Active Directory users that need to access it.

Allow Full control NTFS ACL permissions removed

If you remove **Allow Full control** NTFS ACL permissions for the SYSTEM user on a folder that you shared, that share can become inaccessible and any file system backups taken from that point onwards may not be usable.

You will need to re-create the affected file share. For more information, see Creating, updating, removing file shares. After you recreate the folder or share, you can map and use the Windows file shares from your compute instances.

Can't access a file system using an on-premises client

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

Amazon FSx only supports access from on-premises clients with non-private IP addresses on file systems created after December 17, 2020.

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

New file system is not registered in DNS

For file systems joined to a self-managed Active Directory, Amazon FSx did not register the file system DNS when it was created because the customer network does not use Microsoft DNS.

Amazon FSx does not register file systems in DNS if your network uses a third-party DNS service instead of Microsoft DNS. You must manually set up DNS A entries for your Amazon FSx file systems. For Single-AZ 1 file systems, you will need to add one DNS A entry; for Single-AZ 2 and Multi-AZ file systems, you will need to add two DNS A entries. Use the following procedure to obtain the file system IP address or addresses to use when manually adding the DNS A entries.

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

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

Can't access the file system using a DNS alias

If you're unable to access a file system using a DNS alias, use the following procedure to troubleshoot the issue.

1. Verify that the alias is associated with the file system by doing either of the following steps:

    a. **Using the Amazon FSx console** – Choose the file system that you're trying to access. On the **File system details** page, the **DNS aliases** are shown on the **Network & security** tab.

    b. **Using the CLI or API** – Use the `describe-file-system-aliases` CLI command, or the DescribeFileSystemAliases API operation to retrieve the aliases currently associated with the file system.

2. If the DNS alias is not listed, you must associate it with the file system. For more information, see Managing DNS aliases on existing file systems.

3. If the DNS alias is associated with the file system, verify that you've also configured the following required items:

    - Created service principal names (SPNs) corresponding to the DNS alias on your Amazon FSx file system's Active Directory computer object.

      For more information, see Configure service principal names (SPNs) for Kerberos.

    - Created a DNS CNAME record for the DNS alias that resolves to the default DNS name of the Amazon FSx file system.

      For more information, see Update or create a DNS CNAME record.

4. If you created valid SPNs and a DNS CNAME record, verify that the client's DNS has the DNS CNAME record that resolves to the correct file system.

    a. Run `nslookup` to confirm that the record exists and that it resolves to the file system's default DNS name.

    b. If the DNS CNAME resolves to another file system, wait for the client's DNS cache to refresh, and then check the CNAME record again. You can accelerate the process by flushing the client's DNS cache using the following command.

    ```
    ipconfig /flushdns
    ```

5. If the DNS CNAME record resolves to the Amazon FSx file system's default DNS, and the client is still unable to access the file system, see You can't access your file system for additional troubleshooting steps.

Can't access the file system using an IP address

If you're unable to access your file system using an IP address, try using the DNS name or associated DNS alias instead.

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

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

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

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

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

Creating a new Amazon FSx file system fails

There are a number of potential causes when a file system creation request fails, as described in the following section.

Topics

- Misconfigured VPC security group and network ACLs
- Duplicate file system administrators group names
- DNS servers or domain controllers unreachable
- Invalid service account credentials
- Amazon FSx can't access your Active Directory service account credentials in AWS Secrets Manager
- Insufficient service account permissions
- Service account capacity exceeded
- Amazon FSx can't access the organizational unit (OU)
- Service account can't access the administrators group

- Amazon FSx lost connectivity in domain
- Service account does not have correct permissions
- Unicode characters used in creation parameters
- Switching storage type to HDD while restoring a backup fails

Misconfigured VPC security group and network ACLs

Make sure that the VPC security groups and network ACLs are configured using the recommended security group configuration. For more information, see Creating security groups.

Duplicate file system administrators group names

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
File system creation failed. Amazon FSx is unable to apply your Microsoft Active
 Directory configuration with the
specified file system administrators group. Please ensure that your Active Directory
 does not contain multiple domain
groups with the name: domain_group.
```

Amazon FSx did not create the file system because there are multiple administrator groups in the domain with the same name.

If you don't specify a group name, Amazon FSx will attempt to use the default value "Domain Admins" as the administrator group. The request will fail if there is more than one group using the default "Domain Admins" name.

Use the following steps to resolve the issue.

1. Review the prerequisites for joining your file system to your self-managed Active Directory.
2. Use the Amazon FSx Active Directory Validation Tool to validate your self-managed Active Directory configuration prior to creating an FSx for Windows File Server file system that's joined to a self-managed Active Directory.
3. Create a new file system using the AWS Management Console or AWS CLI. For more information, see Joining an Amazon FSx file system to a self-managed Microsoft Active Directory domain.

4. Provide a name for the file system administrator group that is unique in the domain for your self-managed Active Directory.

DNS servers or domain controllers unreachable

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't reach the DNS servers provided or the domain controllers for your
 self-managed directory in Microsoft Active Directory.
File system creation failed. Amazon FSx is unable to communicate with your Microsoft
 Active Directory domain controllers.
This is because Amazon FSx can't reach the DNS servers provided or domain controllers
 for your domain.
To fix this problem, delete your file system and create a new one with valid DNS
 servers and networking configuration that allows
traffic from the file system to the domain controller.
```

Use the following steps to troubleshoot and resolve the issue.

1. Verify that you followed the prerequisites for having network connectivity and routing established between the subnet where you're creating an Amazon FSx file system, and your self-managed Active Directory. For more information, see Prerequisites.

   Use the Amazon FSx Active Directory Validation tool to test and verify these network settings.

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

2. Verify that you configured the VPC security groups that you associated with your Amazon FSx file system, along with any VPC network ACLs, to allow outbound network traffic on all ports.

Note

If you want to implement least privilege, you can allow outbound traffic only to the specific ports required for communication with the Active Directory domain controllers. For more information, see the Microsoft Active Directory documentation.

3. Verify that the values for Microsoft Windows file server or network administrative properties do not contain non-Latin-1 characters. For example, the file system creation fails if you use `Domänen-Admins` as the name of the file system administrators group.
4. Verify that your Active Directory domain's DNS servers and domain controllers are active and able to respond to requests for the domain provided.
5. Ensure that the functional level of your Active Directory domain is Windows Server 2008 R2 or higher.
6. Make sure that the firewall rules on your Active Directory domain's domain controllers allow traffic from your Amazon FSx file system. For more information, see the Microsoft Active Directory documentation.

Invalid service account credentials

Creating a file system joined to a self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to establish a connection with your Microsoft Active Directory
 domain controllers
because the service account credentials provided are invalid. To fix this problem,
 delete your file
system and create a new one using a valid service account.
```

Use the following steps to troubleshoot and resolve the issue.

Case 1: If you are using an AWS Secrets Manager secret to store your Active Directory credentials

1. Review Storing Active Directory credentials using AWS Secrets Manager.
2. at the secret ARN is correct and follows the proper format: `arn:aws:secretsmanager:region:account-id:secret:secret-name-6chars`.

3. Verify that the secret contains both required fields with non-empty values:
   - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME` – Your AD service account username.
   - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD` – Your AD service account password.
4. Verify that the secret and key have a resource-based policy that grants the Amazon FSx service principal `fsx.amazonaws.com` permission to retrieve the secret value.

Case 2: If you are using plaintext credentials to join your Active Directory

1. Verify that you're entering only the user name as input for the **Service account username**, such as `ServiceAcct`, in the self-managed Active Directory configuration.

   > ⚠ **Important**
   >
   > DO NOT include a domain prefix (`corp.com\ServiceAcct`) or domain suffix (`ServiceAcct@corp.com`) when entering the service account user name.
   > DO NOT use the distinguished name (DN) when entering the service account user name (CN=ServiceAcct,OU=example,DC=corp,DC=com).

2. Verify that the service account that you provided exists in your Active Directory domain.
3. Make sure that you delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain to which you're joining the file system. The service account also needs, at a minimum, to have permissions to do the following:
   - Reset passwords
   - Restrict accounts from reading and writing data
   - Validated ability to write to the DNS hostname
   - Validated ability to write to the service principal name

   For more information about creating a service account with correct permissions, see Amazon FSx service account.

Amazon FSx can't access your Active Directory service account credentials in AWS Secrets Manager

The following sections describe common issues and how to resolve them.

Joining a file system to your self-managed Active Directory fails with the following error message:

```
You can't provide both username/password and a domain join service account
secret to connect to your Active Directory. Provide only one set of
credentials.
```

To resolve this issue

1. Choose whether you want to provide credentials stored in a Secrets Manager secret, or in plaintext.
2. When joining an Active Directory, only provide one of those parameters and not both.

Joining a file system to your self-managed Active Directory fails with the following error message:

```
The domain join service account secret ARN format you entered isn't
valid. Use the format: arn:partition:secretsmanager:region:account-
id:secret:secret-name-6chars
```

To resolve this issue

1. Review Storing Active Directory credentials using AWS Secrets Manager.
2. Verify that the ARN format you are entering is correct. A correct format example is `arn:aws:secretsmanager:us-east-1:123456789012:secret:MyDatabaseSecret-Ab3d5f`.

Joining a file system to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't access the domain join service account secret [ARN]. Add
a resource permission to the secret that grants the FSx service principal
(fsx.amazonaws.com) permission to access it.
```

To resolve this issue

1. Review Storing Active Directory credentials using AWS Secrets Manager.
2. Verify that the Secrets Manager secret you are providing has the correct policies that allow Amazon FSx to use the secret.

Joining a file system to your self-managed Active Directory fails with the following error message:

```
You don't have permission to access the domain join service account secret
[ARN]. A resource permission needs to be added to the secret to grant you
access.
```

To resolve this issue

- The Secrets Manager secret owner or administrator needs to give your account access to use this secret. For more information, see Identity-based policies.

Joining a file system to your self-managed Active Directory fails with the following error message:

```
The domain join service account secret format or content isn't valid. Make
sure the secret includes both CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME
and CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD fields with non-empty
values.
```

To resolve this issue

1. Review Storing Active Directory credentials using AWS Secrets Manager.
2. Verify that the Secrets Manager secret you are providing has both of the required fields.

Insufficient service account permissions

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to establish a connection with your
```

```
Microsoft Active Directory domain controllers. This is because the service account
 provided does not
have permission to join the file system to the domain with the specified organizational
 unit.
To fix this problem, delete your file system and create a new one using a service
 account with
permission to join the file system to the domain with the specified organizational
 unit.
```

Use the following procedure to troubleshoot and resolve the issue.

- Make sure that you delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain to which you're joining the file system. The service account also needs, at a minimum, to have permissions to do the following:

  - Reset passwords
  - Restrict accounts from reading and writing data
  - Validated ability to write to the DNS hostname
  - Validated ability to write to the service principal name

  For more information about creating a service account with correct permissions, see Amazon FSx service account.

Service account capacity exceeded

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't establish a connection with your Microsoft Active Directory
domain controllers. This is because the service account provided has reached the
maximum number of computers that it can join to the domain. To fix this problem,
delete your file system and create a new one, supplying a service account that
is able to join new computers to the domain.
```

To resolve the issue, verify that the service account you provided has reached the maximum number of computers it can join to the domain. If it has reached the maximum limit, create a new service account with the correct permissions. Use the new service account and create a new file system. For more information, see Amazon FSx service account.

Amazon FSx can't access the organizational unit (OU)

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx can't establish a connection with your Microsoft Active Directory domain
 controller(s).
This is because the organizational unit you specified either doesn't exist or isn't
 accessible
to the service account provided. To fix this problem, delete your file system and
 create a new one specifying an
organizational unit to which the service account can join the file system.
```

Use the following steps to troubleshoot and resolve the issue.

1. Verify that the OU you provided is in your Active Directory domain.
2. Make sure that you have delegated the required permissions to the service account that you provided. The service account must be able to create and delete computer objects in the OU in the domain that you're joining the file system to. The service account also needs to have, at a minimum, permissions to do the following:

   - Reset passwords
   - Restrict accounts from reading and writing data
   - Validated ability to write to the DNS hostname
   - Validated ability to write to the service principal name
   - Be delegated control to create and delete computer objects
   - Validated ability to read and write Account Restrictions

   For more information about creating a service account with the correct permissions, see Amazon FSx service account.

Service account can't access the administrators group

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to apply your Microsoft Active Directory configuration. This is
 because the file system
administrators group you provided either doesn't exist or isn't accessible to the
 service account you
provided. To fix this problem, delete your file system and create a new one specifying
 a file
system administrators group in the domain that is accessible to the service account
provided.
```

Use the following steps to troubleshoot and resolve the issue.

1. Ensure that you're providing just the name of the group as a string for the administrators group parameter.

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

2. Ensure that the administrators group provided exists in the same Active Directory domain as the one that you want to join the file system to.
3. If you did not provide an administrator group parameter, Amazon FSx attempts to use the `Builtin Domain Admins` group in your Active Directory domain. If the name of this group has been changed, or if you're using a different group for domain administration, you need to provide that name for the group.

Amazon FSx lost connectivity in domain

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
Amazon FSx is unable to apply your Microsoft Active Directory configuration. To fix
 this problem, delete your file system and create a new one
meeting the pre-requisites described in the Amazon FSx user guide.
```

When creating your file system, Amazon FSx was able to reach your Active Directory domain's DNS servers and domain controllers, and join the file system successfully to your Active Directory domain. However, while completing file system creation, Amazon FSx lost connectivity to or membership in your domain. Use the following steps to troubleshoot and resolve the issue.

1. Ensure that network connectivity continues to exist between your Amazon FSx file system and your Active Directory. And, ensure that network traffic continues to be allowed between them by using routing rules, VPC security group rules, VPC network ACLs, and domain controller firewall rules.
2. Ensure that the computer objects created by Amazon FSx for your file systems in your Active Directory domain are still active, and were not deleted or otherwise manipulated.

Service account does not have correct permissions

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
File system creation failed. Amazon FSx is unable to establish a connection with your
 Microsoft Active Directory domain controller(s).
This is because the service account provided does not have permission to join the file
 system to the domain with the specified
organizational unit (OU). To fix this problem, delete your file system and create a new
 one using a service account with permission
to create computer objects and reset passwords within the specified organizational
 unit.
```

Make sure that you have delegated the required permissions to the service account that you provided. Use the following steps to troubleshoot and resolve the issue.

The service account needs to have, at a minimum, the following permissions:

- Be delegated control to create and delete computer objects in the OU that you're joining the file system to
- Have the following permissions in the OU that you're joining the file system to:

- Ability to reset passwords
- Ability to restrict accounts from reading and writing data
- Validated ability to write to the DNS hostname
- Validated ability to write to the service principal name
- Ability (can be delegated) to create and delete computer objects
- Validated ability to read and write Account Restrictions
- Ability to modify permissions

For more information about creating a service account with the correct permissions, see Amazon FSx service account.

Unicode characters used in creation parameters

Creating a file system joined to your self-managed Active Directory fails with the following error message:

```
File system creation failed. Amazon FSx is unable to create a file system within the
 specified
Microsoft Active Directory. To fix this problem, please delete your file system and
 create a new one
meeting the pre-requisites described in the FSx for ONTAP User Guide.
```

Amazon FSx does not support Unicode characters. Verify that none of the creation parameters have Unicode characters, such as accent marks. This includes parameters that can be left blank where a default value is filled in automatically. Ensure the corresponding default values in your Active Directory also do not contain Unicode characters.

Switching storage type to HDD while restoring a backup fails

Creating a file system from a backup fails with the following error message:

```
Switching storage type to HDD while creating a file system from backup
backup_id is not supported because a storage scaling activity was still
under way on the source file system to increase storage capacity from less
than 2000 GiB when the backup backup_id was taken, and the minimum storage
capacity for HDD storage is 2000 GiB.
```

This issue occurs when restoring a backup and you have changed the storage type from SSD to HDD. The restore from backup fails because the backup that you are restoring was taken while a storage capacity increase was still in progress on the original file system. The file system's SSD storage capacity before the increase request was less than 2000 GiB, which is the minimum storage capacity required to create an HDD file system.

Use the following procedure to resolve this issue.

1. Wait for the storage capacity increase request to complete and the file system has at least 2000 GiB of SSD storage capacity. For more information, see Monitoring storage capacity increases.
2. Take a user-initiated backup of the file system. For more information, see Working with user-initiated backups.
3. Restore the user-initiated backup to a new file system using HDD storage. For more information, see Restoring backups to new file system.

File system is in a misconfigured state

An FSx for Windows File Server file system can get into a **Misconfigured** state due to a change in your Active Directory environment. In this state, your file system is either currently unavailable or at risk of losing availability, and backups may not succeed.

The **Misconfigured** state includes an error message and recommended corrective action that you can access using the Amazon FSx console, API, or AWS CLI. After taking the corrective action, verify that your file system's state eventually changes to `Available` – note that this change can take several minutes to complete.

Your file system can get into a **Misconfigured** state for several reasons, such as the following:

- The DNS Server IP addresses are no longer valid.
- The service account credentials are no longer valid, or lack required permissions.
- The Active Directory domain controller is not reachable due to network connectivity issues, such as invalid VPC Security Groups, VPC Network ACL or routing table configuration, or domain controller firewall settings.

⚠ **Important**

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

(For the full list of Active Directory requirements, see Prerequisites. You can also validate that your Active Directory environment is properly configured to meet these requirements by using the Amazon FSx Active Directory Validation tool.)

Resolving some of these issues requires directly updating one or more parameters in your file system's Active Directory configuration, such as changing DNS Server IP addresses, or changing the service account username or password. In these cases, your corrective action will necessarily involve using the Amazon FSx console, API, or AWS CLI to update the required configuration parameters.

Other issues may not require changing any Active Directory configuration parameters, such as changing your domain controller firewall settings or VPC Security Groups. In these cases, however, you will need to take further action before the file system can become `Available`. After ensuring your Active Directory environment is configured properly, select the **Attempt Recovery** button next to the **Misconfigured** status in the Amazon FSx console, or use the `StartMisconfiguredStateRecovery` command in the Amazon FSx console, API, or AWS CLI.

Topics

- Misconfigured file system: Amazon FSx can't reach either the DNS servers or domain controllers for your domain.
- Misconfigured file system: The service account credentials are invalid
- Misconfigured file system: The AWS Secrets Manager secret or KMS key is not configured correctly
- Misconfigured file system: The service account provided doesn't have permission to join the file system to the domain
- Misconfigured file system: The service account can't join any more computers to domain
- Misconfigured file system: The service account doesn't have access to the OU

Misconfigured file system: Amazon FSx can't reach either the DNS servers or domain controllers for your domain.

A file system will go into a `Misconfigured` state when Amazon FSx can't communicate with your Microsoft Active Directory domain controller or controllers.

To resolve this situation, do the following:

1. Make sure that your networking configuration allows traffic from the file system to the domain controller.
2. Use the Amazon FSx Active Directory Validation tool to test and verify the network settings for your self-managed Active Directory. For more information, see Using a self-managed Microsoft Active Directory.
3. Review the file system's self-managed Active Directory configuration in the Amazon FSx console.
4. To update the file system's self-managed Active Directory configuration, you can use the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.
    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx CLI `update-file-system` command or the API operation UpdateFileSystem.

Misconfigured file system: The service account credentials are invalid

Amazon FSx can't establish a connection with your Microsoft Active Directory domain controller or controllers. This is because the service account credentials provided are invalid. For more information, see Using a self-managed Microsoft Active Directory.

To resolve the misconfiguration, do the following:

1. Verify that you are using the correct service account, and you are using the correct credentials for that account.
2. Then update the file system's configuration with the correct service account or account credentials using the Amazon FSx console.

a. On the navigation pane, choose **File systems**, and choose the misconfigured file system to update.

b. On the **File system details** page, choose **Update** in the **Networking and security** tab.

You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the Amazon FSx API Reference.

Misconfigured file system: The AWS Secrets Manager secret or KMS key is not configured correctly

Amazon FSx can't establish a connection with your Microsoft Active Directory domain controller or controllers. This is because your AWS Secrets Manager secret or AWS KMS key is not configured correctly. For more information, see Storing Active Directory credentials using AWS Secrets Manager.

To resolve the misconfiguration, do the following:

1. Verify that the secret ARN is correct and follows the proper format: `arn:aws:secretsmanager:region:account-id:secret:secret-name-6chars`.
2. Verify that the secret contains both required fields with non-empty values:
   - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME` – Your AD service account username.
   - `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD` – Your AD service account password.
   - Verify that the secret and key have a resource-based policy that grants the Amazon FSx service principal `fsx.amazonaws.com` permission to retrieve the secret value.

Misconfigured file system: The service account provided doesn't have permission to join the file system to the domain

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers. This is because the service account provided doesn't have permission to join the file system to the domain with the specified OU.

To resolve the misconfiguration, do the following:

1. Add the required permissions to the Amazon FSx service account, or create a new service account with the required permissions. For more information about doing this, see Amazon FSx service account.
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials. To update the configuration, you can use the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the Amazon FSx API Reference.

Misconfigured file system: The service account can't join any more computers to domain

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers. In this case, this is because the service account provided has reached the maximum number of computers that it can join to the domain.

To resolve the misconfiguration, do the following:

1. Identify another service account or create a new service account that can join new computers to the domain.
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials using the Amazon FSx console.

    a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

    b. On **File system details** page, choose **Update** on the **Networking and security** tab.

    You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the Amazon FSx API Reference.

Misconfigured file system: The service account doesn't have access to the OU

Amazon FSx can't establish a connection to your Microsoft Active Directory domain controllers because the service account provided doesn't have access to the OU specified.

To resolve the misconfiguration, do the following:

1. Identify another service account or create a new service account that has access to the OU.
2. Then update the file system's self-managed Active Directory configuration with the new service account credentials.

   a. On the navigation pane, choose **File systems**, and choose the file system to update; the **File system details** page appears.

   b. On **File system details** page, choose **Update** on the **Networking and security** tab.

   You can also use the Amazon FSx API operation `update-file-system`. To learn more, see the UpdateFileSystem in the Amazon FSx API Reference.

You can't configure DFS-R on a Multi-AZ or Single-AZ 2 file system

Microsoft Distributed File System Replication (DFS-R) is not supported on Multi-AZ and Single-AZ 2 file systems.

Multi-AZ file systems are configured for redundancy across multiple access zones natively. Use the Multi-AZ deployment type for high availability across multiple Availability Zones. For more information, see Availability and durability: Single-AZ and Multi-AZ file systems.

Storage or throughput capacity updates fail

There are a number of potential causes for file system storage and throughput capacity update requests to fail, each with their own resolution.

Storage capacity increase fails because Amazon FSx can't access the file system's AWS KMS key

A storage capacity increase request failed because Amazon FSx was unable to access the KMS key used to encrypt file system.

You need to ensure that Amazon FSx has access to the KMS key used to encrypt the file system in order to run the administrative action. Use the following information to resolve the key access issue.

- If the KMS key has been deleted, the file system and any of its backups using the deleted KMS key are unrecoverable. For more information, see Deleting AWS KMS keys in the AWS Key Management Service Developer Guide.
- If the KMS key is disabled, and it is a customer managed key, you will need to re-enable it, and then retry the storage capacity increase request. For more information, see Enabling and disabling keys in the AWS Key Management Service Developer Guide.
- If the key is invalid because of its pending deletion, you must cancel the key deletion while it is still in a `PendingDeletion` state. You can retry the request once the KMS key is `Enabled`.
- If the key is invalid because of its pending import, you must wait until the import has completed, and then retry the storage increase request.
- If the key's grant limit has been exceeded, you must request an increase in the number of grants for the key. For more information, see Resource quotas in the AWS Key Management Service Developer Guide. When the quota increase is granted, retry the storage increase request.

Storage or throughput capacity update fails because the self-managed Active Directory is misconfigured

The storage capacity or throughput capacity update request failed because your file system's self-managed Active Directory is in a misconfigured state.

To resolve the specific misconfigured state, see File system is in a misconfigured state.

Storage capacity increase fails because of insufficient throughput capacity

The storage capacity increase request failed because the file system's throughput capacity is set to 8 MBps.

Increase the file system's throughput capacity to a minimum of 16 MBps, then retry the request. For more information, see Managing throughput capacity.

Throughput capacity update to 8 MBps fails

A request to modify a file system's throughput capacity to 8 MBps failed.

This can occur when a storage capacity increase request is pending or in progress. Storage capacity increases require a minimum throughput of 16 MBps. Wait until the storage capacity increase request has completed, and then retry the throughput capacity modification request.

Document history

- **API version:** 2018-03-01
- **Latest documentation update:** September 30, 2025

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.

Change	Description	Date
Support added for AWS Secrets Manager integration	Amazon FSx now integrates with AWS Secrets Manager for enhanced management of Active Directory credentials. For more information, see Storing Active Directory credentials using AWS Secrets Manager.	November 5, 2025
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added a new permission, `secretsmanager:ListSecrets` , that allows principals to list secrets in AWS Secrets Manager for selecting domain join service account credentials. For more information, see AWS managed policy: AmazonFSxFullAccess .	November 5, 2025
Support added for Internet Protocol version 6 (IPv6)	FSx for Windows File Server file systems now support two network type options: IPv4-only and dual-stack (for both IPv4 and IPv6). You must specify one of	September 30, 2025

	those options when creating your file system. You can change the network type of an existing FSx for Windows File Server file system at any time. For more information, see Managing network type.	
Amazon FSx updated the AmazonFSxServiceRolePolicy AWS managed policy	Amazon FSx added the `ec2:AssignIpv6Addresses` and `ec2:UnassignIpv6Addresses` permissions to the AmazonFSxServiceRolePolicy. For more information, see AmazonFSxServiceRolePolicy.	July 22, 2025
Amazon FSx updated the AmazonFSxFullAccess AWS managed policy	The AmazonFSxFullAccess managed policy was updated to add the `fsx:CreateAndAttachS3AccessPoint` , `fsx:DescribeS3AccessPointAttachments` , and `fsx:DetachAndDeleteS3AccessPoint` permissions.	June 25, 2025

Amazon FSx updated the AmazonFSxConsoleFullAccess AWS managed policy	The AmazonFSxConsoleFullAccess managed policy was updated to add the `fsx:CreateAndAttachS3AccessPoint` , `fsx:DescribeS3AccessPointAttachments` , and `fsx:DetachAndDeleteS3AccessPoint` permissions.	June 25, 2025
Amazon FSx updated the AmazonFSxConsoleReadOnlyAccess AWS managed policy	Amazon FSx updated the AmazonFSxConsoleReadOnlyAccess policy to add the `ec2:DescribeNetworkInterfaces` permission. For more information, see the AmazonFSxConsoleReadOnlyAccess policy.	February 25, 2025
Support added for dual-stack VPC interface endpoints for Amazon FSx	You can now create dual-stack VPC interface endpoints for Amazon FSx with both IPv4 and IPv6 IP addresses and DNS names. For more information, see FSx for Windows File Server and interface VPC endpoints.	February 7, 2025
Support added for dual-stack API endpoints	The Amazon FSx service API for creating and managing file systems have new dual-stack endpoints. For more information, see API endpoints in the Amazon FSx API Reference.	February 7, 2025

Amazon FSx updated the AmazonFSxConsoleFullAccess AWS managed policy	Amazon FSx updated the AmazonFSxConsoleFullAccess policy to add the `ec2:DescribeNetworkInterfaces` permission. For more information, see AmazonFSxConsoleFullAccess policy.	February 7, 2025
Updated version of the FSx for Windows File Server Active Directory Validation tool	An updated version of the FSx for Windows File Server Active Directory Validation tool is available. For more information, see Validating your Active Directory configuration	November 6, 2024
Support added for higher levels of IOPS on file systems with throughput capacities of 4 GBps and higher	FSx for Windows File Server is increasing maximum IOPS from 130K to 150K for file systems with 4 GBps of throughput capacity or higher, from 175K to 200K for file systems with 6 GBps of throughput capacity or higher, from 260K to 300K for file systems with 9 GBps of throughput capacity or higher, and from 350K to 400K for file systems with 12 GBps of throughput capacity or higher. For more information, see FSx for Windows File Server performance.	January 17, 2024

Amazon FSx updated the AmazonFSxFullAccess, AmazonFSxConsoleFullAccess, AmazonFSxReadOnlyAccess, AmazonFSxConsoleReadOnlyAccess, and AmazonFSxServiceRolePolicy AWS managed policies	Amazon FSx updated the AmazonFSxFullAccess, AmazonFSxConsoleFullAccess, AmazonFSxReadOnlyAccess, AmazonFSxConsoleReadOnlyAccess, and AmazonFSxServiceRolePolicy policies to add the `ec2:GetSecurityGroupsForVpc` permission. For more information, see Amazon FSx updates to AWS managed policies.	January 9, 2024
Amazon FSx updated the AmazonFSxFullAccess and the AmazonFSxConsoleFullAccess AWS managed policies	Amazon FSx updated the AmazonFSxFullAccess and AmazonFSxConsoleFullAccess policies to add the `ManageCrossAccountDataReplication` action. For more information, see Amazon FSx updates to AWS managed policies.	December 20, 2023
Amazon FSx updated the AmazonFSxFullAccess and the AmazonFSxConsoleFullAccess AWS managed policies	Amazon FSx updated the AmazonFSxFullAccess and AmazonFSxConsoleFullAccess policies to add the `fsx:CopySnapshotAndUpdateVolume` permission. For more information, see Amazon FSx updates to AWS managed policies.	November 26, 2023

Amazon FSx updated the AmazonFSxFullAccess and the AmazonFSxConsoleFullAccess AWS managed policies	Amazon FSx updated the AmazonFSxFullAccess and AmazonFSxConsoleFullAccess policies to add the `fsx:DescribeSharedVPCConfiguration` and `fsx:UpdateSharedVPCConfiguration` permissions. For more information, see Amazon FSx updates to AWS managed policies.	November 14, 2023
Support added for updating file system storage type	FSx for Windows File Server file systems now support updating from HDD storage type to SSD storage type. For more information, see Managing storage type.	August 9, 2023
Support added for higher maximum throughput capacity	FSx for Windows File Server file systems now support up to 12 GBps throughput capacity. For more information, see FSx for Windows File Server performance.	August 9, 2023
Support added for SSD IOPS provisioning	FSx for Windows File Server file systems now support SSD IOPS provisioning independently of storage capacity, up to a maximum of 350,000 IOPS. For more information, see Managing SSD IOPS.	August 9, 2023

Amazon FSx updated the AmazonFSxServiceRolePolicy AWS managed policy	Amazon FSx updated the `cloudwatch:PutMetricData` permission in the AmazonFSxServiceRolePolicy. For more information, see AmazonFSxServiceRolePolicy.	July 24, 2023
Amazon FSx updated the AmazonFSxFullAccess AWS managed policy	Amazon FSx updated the AmazonFSxFullAccess policy to remove the `fsx:*` permission and add specific `fsx` actions. For more information, see AmazonFSxFullAccess policy.	July 13, 2023
Amazon FSx updated the AmazonFSxConsoleFullAccess AWS managed policy	Amazon FSx updated the AmazonFSxConsoleFullAccess policy to remove the `fsx:*` permission and add specific `fsx` actions. For more information, see AmazonFSxConsoleFullAccess policy.	July 13, 2023
Support added for new CloudWatch metrics for Amazon FSx for Windows File Server	FSx for Windows File Server now provides additional CloudWatch metrics that monitor file server and storage volume performance and capacity usage. For more information, see Metrics and dimensions.	September 22, 2022

Support added for file system performance warnings	Amazon FSx now provides warnings in the **Performance & monitoring** window when any of a set of CloudWatc h metrics approach or cross predetermined threshold s for these metrics. Each warning also provides an actionable recommendation for improving the file system's performance. For more information, see Performan ce warnings and recommend ations.	September 22, 2022
Support added for enhanced file system performance monitoring	The Amazon FSx console file system monitoring dashboard for FSx for Windows File Server file systems includes new **Summary**, **Storage**, and **Performance** sections. These sections display graphs of new CloudWatch metrics that provide you with enhanced performance monitoring. For more information, see Monitoring metrics with CloudWatch.	September 22, 2022
Support added for AWS PrivateLink interface VPC endpoints.	You can now use interface VPC endpoints to access the Amazon FSx API from your VPC without sending traffic over the internet. For more information, see Amazon FSx and interface VPC endpoints.	April 5, 2022

Support added for Amazon Kendra	You can now use your FSx for Windows File Server file system as a data source for Amazon Kendra, allowing you to index and search for information contained in documents stored on your file system. For more information, see Using FSx for Windows File Server with Amazon Kendra.	March 26, 2022
Support added for file access auditing	You can now enable auditing of end-user accesses on files, folders,and file shares. You can choose to send audit event logs to the Amazon CloudWatch Logs or Amazon Data Firehose services. For more information, see File access auditing.	June 8, 2021
Support added for copying backups	You can now use Amazon FSx to copy backups within the same AWS account to another AWS Region (cross-Region copies) or within the same AWS Region (in-Region copies). For more information, see Copying backups.	April 12, 2021

Automatically increase a file system's storage capacity	Use an AWS-developed customizable CloudFormation template to automatically increase your file system's stoage capacity when its capacity reaches a threshold that you specify. For more information, see Increasing storage capacity dynamically.	February 17, 2021
Support added for client access using non-private IP addresses	You can access FSx for Windows File Server file systems with on-premises clients using non-private IP addresses. For more information, see Supported environments. You can join FSx for Windows File Server file system to a self-managed Microsoft Active Directory with DNS servers and AD domain controllers that use non-private IP addresses. For more information, see Using Amazon FSx with Your Self-Managed Microsoft Active Directory.	December 17, 2020

Support added for using DNS aliases	You can now associate DNS aliases with your FSx for Windows File Server file systems that you can use to access the data on your file system. For more information, see Managing DNS aliases and Walkthrough 5: Using DNS aliases to access your file system.	November 9, 2020
Support added for Amazon Elastic Container Service	You can now use FSx for Windows File Server with Amazon ECS. For more information, see Supported Clients.	November 9, 2020
Amazon FSx is now integrated with AWS Backup	You can now use AWS Backup to back up and restore your FSx file systems in addition to using native Amazon FSx backups. For more information, see Using AWS Backup with Amazon FSx.	November 9, 2020
Support added for throughput capacity scaling	You can now modify the throughput capacity for existing FSx for Windows File Server file systems as your throughput requirements evolve. For more information, see Managing Throughput Capacity.	June 1, 2020

Support added for storage capacity scaling	You can now increase the storage capacity for existing FSx for Windows File Server file systems as your storage requirements evolve. For more information, see Managing Storage Capacity.	June 1, 2020
Support added for hard disk drive (HDD) storage	HDD storage gives you price and performance flexibility when using FSx for Windows File Server. For more information, see Optimizing Costs with Amazon FSx.	March 26, 2020
Support added for file transfer using AWS DataSync	You can now use AWS DataSync to transfer files to and from your FSx for Windows File Server. For more information, see Migrate Files to Amazon FSx for Windows File Server Using AWS DataSync.	February 4, 2020
FSx for Windows File Server releases support for additiona l Windows file system administration tasks	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 PowerShel l. For more information, see Administering File Systems.	November 20, 2019

FSx for Windows File Server releases native Multi-AZ support	You can use Multi-AZ deployment for FSx for Windows File Server to more easily create file systems with high availability that span multiple Availability Zones (AZs). For more information, see Availability and Durability: Single-AZ and Multi-AZ File Systems.	November 20, 2019
FSx for Windows File Server releases support for managing user sessions and open files	You can now use the Shared Folders tool native to Microsoft Windows to manage user sessions and open files on your FSx for Windows File Server file systems. For more information, see Managing User Sessions and Open Files.	October 17, 2019
Amazon FSx releases support for Microsoft Windows shadow copies	You can now configure Windows shadow copies on your FSx for Windows File Server file systems. Shadow copies enable your users to easily undo file changes and compare file versions by restoring files to previous versions. For more information, see Working with Shadow Copies.	July 31, 2019

Amazon FSx releases shared Microsoft Active Directory support	You can now join FSx for Windows File Server file systems to AWS Managed Microsoft AD directories that are in a different VPC or in a different AWS account than the file system. For more information, see Active Directory Support.	June 25, 2019
Amazon FSx releases enhanced Microsoft Active Directory support	You can now join FSx for Windows File Server file systems to your self-managed Microsoft Active Directory domains, either on-premises or in the cloud. For more information, see Active Directory Support.	June 24, 2019
Amazon FSx complies with SOC certification	Amazon FSx has been assessed to comply with SOC certification. For more information, see Security and Data Protection.	May 16, 2019
Added clarifying note regarding Direct Connect, VPN, and inter-region VPC peering connection support	Amazon FSx file systems created after February 22, 2019 are accessible using Direct Connect, VPN, and inter-region VPC peering. For more information, see Supported Access Methods.	February 25, 2019

Direct Connect, VPN, and inter-region VPC peering connection support added	You can now access Amazon FSx for Windows File Server file systems from on-premises resources and from resources in a different Amazon VPC or AWS account. For more information, see Supported Access Methods.	February 22, 2019
Amazon FSx is now generally available	Amazon FSx for Windows File Server provides Microsoft Windows file servers that are fully managed, backed by a fully native Windows file system. Amazon FSx for Windows File Server provides the features, performance, and compatibility to easily lift and shift enterprise applications to AWS.	November 28, 2018