aws

OpenZFS User Guide

FSx for OpenZFS

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

FSx for OpenZFS: OpenZFS User Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon FSx for OpenZFS? 1
Features of Amazon FSx for OpenZFS 2
Security and data protection 3
Availability and durability 3
Pricing for FSx for OpenZFS 4
Are you a first-time Amazon FSx user? 4
Setting up a file system 5
Prerequisites 5
..... 5
Step 1: Create a file system 7
Step 2: Mount your file system 14
Step 3: Clean up your resources 18
AWS Regions 20
Accessing your data 24
Accessing data through volumes 24
Automatically mounting file systems on reboot for Linux instances 24
Additional mounting options to maximize file system performance 26
Accessing data within the AWS Cloud 27
Accessing from the same VPC 27
Access from a different VPC 28
Accessing data from on-premises 31
Accessing Multi-AZ file systems 31
Accessing data using AWS container services 32
Mounting from Amazon ECS 32
Availability and durability 34
Choosing a deployment type based on availability and durability 34
Multi-AZ (HA) 34
Single-AZ (HA) 35
Single-AZ (non-HA) 35
Failover process for FSx for OpenZFS 36
Testing failover on Multi-AZ (HA) and Single-AZ (HA) file systems 36
Working with file system resources 37
Subnets 37
File system elastic network interfaces 37

Performance 40
File system performance 40
Choosing a deployment type based on performance 41
Choosing a storage class based on performance 41
Migrating between deployment types and storage classes 42
General tips for maximizing performance 42
Client considerations 43
File system and volume configurations 45
Monitoring performance 47
Working with SSD storage 48
Data access from cache 49
Data access from disk 54
SSD IOPS and performance 59
Working with Intelligent-Tiering 60
Data access from in-memory cache 62
Data access from SSD cache and Intelligent-Tiering 63
Performance considerations 65
Managing file systems 66
Creating a file system 66
Creating a file system 66
Configurable file system properties 67
Viewing a file system 69
Viewing file system details 69
File system status 69
Updating a file system 70
Updating a file system 71
Modifiable file system properties 72
Modifying SSD storage capacity and IOPS 73
Modifying SSD read cache 80
Modifying throughput capacity 84
Modifying maintenance windows 86
Deleting a file system 87
Deleting a file system 88
Managing volumes 89
Creating a volume 89
Creating a volume 90

Configurable volume properties ... 93
Viewing a volume ... 96
Updating a volume ... 97
Updating a volume ... 97
Modifiable volume properties ... 100
Deleting a volume ... 100
Tagging your resources ... 102
Tag basics ... 102
Tagging your resources ... 103
Tag restrictions ... 103
Permissions and tag ... 104
Protecting your data ... 105
Protect data with backups ... 105
Working with automatic daily backups ... 106
Working with user-initiated backups ... 107
Working with AWS Backup ... 107
Copying backups ... 108
Restoring backups ... 111
Deleting backups ... 115
Protecting data with snapshots ... 116
Using snapshots to create volumes ... 116
Creating a snapshot ... 117
Deleting a snapshot ... 118
Viewing a snapshot ... 119
Restoring a volume from a snapshot ... 119
Restoring individual files and folders ... 120
Setting up a custom snapshot schedule ... 120
Protect data with replication ... 125
Prerequisites for using on-demand data replication ... 126
Performance considerations for on-demand data replication ... 126
Using on-demand data replication ... 127
Monitoring progress of on-demand data replication ... 129
Setting up ongoing periodic data replication ... 131
Monitoring file systems ... 137
Monitoring with CloudWatch ... 137
Using CloudWatch metrics ... 139

Accessing CloudWatch metrics ... 142
Metrics and dimensions ... 144
Performance warnings and recommendations ... 150
Creating CloudWatch alarms ... 151
Logging API calls with AWS CloudTrail ... 153
Amazon FSx Information in CloudTrail ... 154
Understanding Amazon FSx Log File Entries ... 155
Migrating your existing file storage ... 157
Migrating files with AWS DataSync ... 158
Prerequisites ... 158
DataSync migration basic steps ... 158
Migrating files with rsync ... 159
Migrating files with Robocopy ... 160
Cutting over to your file system ... 162
Security ... 163
Data encryption ... 164
Encryption at rest ... 165
Encryption in transit ... 167
Managing file system access ... 167
Amazon VPC security groups ... 168
Identity and access management ... 169
Audience ... 170
Authenticating with identities ... 170
Managing access using policies ... 174
How Amazon FSx for OpenZFS works with IAM ... 176
Identity-based policy examples ... 182
AWS managed policies ... 185
Troubleshooting IAM ... 198
Using tags to control access to resources ... 200
Using service-linked roles ... 205
Compliance validation ... 211
Interface VPC endpoints ... 212
Considerations for Amazon FSx interface VPC endpoints ... 212
Creating an interface VPC endpoint for Amazon FSx API ... 213
Creating a VPC endpoint policy for Amazon FSx ... 214
Resilience ... 214

Backup and restore ... 214
Snapshots ... 215
Infrastructure security ... 215
Troubleshooting ... 216
Troubleshooting file system issues ... 216
Cannot create a file system because of misconfigured security group ... 216
The Elastic IP address attached to the file system elastic network interface was deleted ... 216
The file system's elastic network interface was modified or deleted ... 217
The compute instance's subnet doesn't use any of the route tables associated with your file system ... 217
Troubleshooting volume mounting issues ... 217
Mounting a volume fails right away ... 217
Mounting a volume hangs and then fails with timeout error ... 218
Mounting a volume using a DNS name fails ... 218
Troubleshooting storage issues ... 219
Deleting files does not reduce used storage capacity ... 219
Service quotas ... 220
Resource quotas for each AWS account ... 220
Resource quotas for each file system ... 221
Requesting a quota increase ... 224
Document history ... 226

What is Amazon FSx for OpenZFS?

Amazon FSx for OpenZFS is a fully managed file storage service that makes it easy to move data to AWS from on-premises ZFS or other Linux-based file servers. You can do this without changing your application code or how you manage data. It offers highly reliable, scalable, performance, and feature-rich file storage built on the open-source OpenZFS file system. It combines these capabilities with the agility, scalability, and simplicity of a fully managed AWS service.

Amazon FSx for OpenZFS file systems are broadly accessible from Linux, Windows, and macOS compute instances and containers using the industry-standard NFS protocol (v3, v4.0, v4.1, v4.2). Powered by the latest AWS compute, disk, and networking technologies, including AWS Scalable Reliable Datagram networking and the AWS Nitro system, Amazon FSx for OpenZFS delivers up to 2 million IOPS with latencies of hundreds of microseconds. With complete support for OpenZFS features like instant point-in-time snapshots and data cloning, FSx for OpenZFS makes it easy for you to replace your on-premises file servers with AWS storage that provides familiar file system capabilities and eliminates the need to perform lengthy qualifications and change or re-architect existing applications or tools. What's more, by combining the power of OpenZFS data management capabilities with the high performance and cost efficiency of the latest AWS technologies, FSx for OpenZFS enables you to build and run high-performance, data-intensive applications.

As a fully managed service, FSx for OpenZFS makes it easy to launch, run, and scale fully managed file systems on AWS that replace the file servers you run on premises while helping to provide better agility and lower costs. With Amazon FSx for OpenZFS, you no longer have to worry about setting up and provisioning file servers and storage volumes, replicating data, installing and patching file server software, detecting and addressing hardware failures, or manually performing backups. FSx for OpenZFS also provides rich integration with other AWS services, such as AWS Identity and Access Management IAM, AWS Key Management Service (AWS KMS), Amazon CloudWatch, and AWS CloudTrail.

For a list of AWS Regions in which Amazon FSx for OpenZFS is available, see Availability by AWS Region.

Topics

- Features of Amazon FSx for OpenZFS
- Security and data protection
- Availability and durability

- Pricing for FSx for OpenZFS
- Are you a first-time Amazon FSx user?

Features of Amazon FSx for OpenZFS

With FSx for OpenZFS, you get a fully managed file storage solution with:

- Support for access from Linux, Windows, and macOS compute instances and containers, including those running on AWS or on-premises, via the industry-standard NFS protocol (v3, v4.0, v4.1, and v4.2).
- Millions of IOPS with latencies of a few hundred microseconds, and up to 21 GBps of throughput for frequently accessed data from in-memory or NVMe cache. Up to 400,000 IOPS and 10 GBps of read/write throughput (up to 21 GBps compressed) for data accessed from disk. For more information, see File system performance.
- Powerful OpenZFS data management capabilities including data compression, near instant point-in-time snapshots, and data cloning, designed for use with the Amazon FSx API.
- Three levels of availability and durability, with Multi-AZ (HA), Single-AZ (HA), and Single-AZ (non-HA) file systems.
- Two storage classes: Intelligent-Tiering and SSD storage. The Intelligent-Tiering storage class offers fully elastic, cost-effective storage that is suitable for most workloads, as well as an optional SSD read cache that you can provision. With Intelligent-Tiering, you are billed for the data you store, depending on the size of your dataset, and do not need to specify a file system size. The SSD storage class provides high performance with low-latency access to your full dataset. With SSD storage, you specify a file system size and pay for the amount of storage that you provision.
- Support for multiple volumes per file system, thin provisioning, and user and group quotas for cost-efficient shared file systems across multiple users and applications.
- Support for the following data protection and security features:
  - Built-in, fully managed file system backups stored on S3, with support for cross-region backup copies.
  - Near-instant point-in-time OpenZFS snapshots stored locally on each file system.
  - Automatic encryption of file system data and backups at rest using KMS keys.
  - Automatic encryption in-transit when accessed from supported EC2 instances.

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 in file systems and backups using keys that you manage in AWS Key Management Service (AWS KMS). Encryption of data in transit is automatically enabled when you access an Amazon FSx file system from Amazon EC2 instances that support this feature. For more information, see Data encryption in Amazon FSx for OpenZFS.

Amazon FSx has been assessed to comply with International Organization for Standardization (ISO), Payment Card Industry Data Security Standard (PCI DSS), and System and Organization Controls (SOC) certifications, and is Health Insurance Portability and Accountability Act of 1996 (HIPAA) eligible. For more information, see Compliance validation for Amazon FSx for OpenZFS.

Amazon FSx provides access control at the file system level using Amazon Virtual Private Cloud (Amazon VPC) security groups, and at the API level using AWS Identity and Access Management (IAM) access policies. To provide access control at the file and folder level, Amazon FSx supports Unix permissions. Amazon FSx integrates with AWS CloudTrail to monitor and log your Amazon FSx API calls so that you can see actions taken by users on your Amazon FSx resources. For more information, see Logging FSx for OpenZFS API calls with AWS CloudTrail.

Additionally, Amazon FSx protects your data with highly durable file system backups. Amazon FSx performs automatic daily backups, and you can take additional backups at any point. For more information, see Protecting your Amazon FSx for OpenZFS data.

Availability and durability

FSx for OpenZFS supports three file system deployment types—Multi-AZ (HA), Single-AZ (HA), and Single-AZ (non-HA)—with each one offering a different level of availability and durability:

- Multi-AZ (HA) file systems offer high availability and high durability by replicating your data and supporting failover across multiple Availability Zones in the same AWS Region, with a separate copy of your data in each Availability Zone. Failover typically completes within 60 seconds.
- Single-AZ (HA) file systems offer high availability by deploying a primary and standby file system within the same Availability Zone to deliver continuous availability in the event of failover and failback. Failover typically completes within 60 seconds.
- Single-AZ (non-HA) file systems ensure self-healing recovery within a single Availability Zone by automatically detecting and addressing component failures. Recovery typically completes within 30 minutes.

For more information, see Availability and durability for Amazon FSx for OpenZFS.

Pricing for FSx for OpenZFS

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 FSx for OpenZFS pricing.

Are you a first-time Amazon FSx user?

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

1. If you're new to AWS, see Prerequisites to set up an AWS account.
2. If you're ready to create your first Amazon FSx file system, follow the instructions in Setting up an Amazon FSx for OpenZFS file system.
3. For information about performance, see Performance for Amazon FSx for OpenZFS.
4. For Amazon FSx security details, see Security in Amazon FSx for OpenZFS.
5. For information about the Amazon FSx API, see Amazon FSx API Reference.
6. For information about the Amazon FSx AWS CLI, see AWS Command Line Interface Command Reference for Amazon FSx.
7. For more information about pricing and fees associated with the service, see FSx for OpenZFS pricing.

Setting up an Amazon FSx for OpenZFS file system

If you are getting started with FSx for OpenZFS for the first time, follow these steps to learn how to create your file system, mount it from an Amazon EC2 instance, and clean up your resources once you are done.

Topics

- Prerequisites
- Step 1: Create a file system
- Step 2: Mount your file system from an Amazon EC2 instance
- Step 3: Clean up your resources

Prerequisites

Before you use Amazon FSx for the first time, make sure that you have completed 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 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: Create a file system

The following procedures detail how to create a file system using the **Quick create** and **Standard create** options on the Amazon FSx console. For instructions on how to create a file system using the AWS CLI instead of the AWS Management Console, see Creating an Amazon FSx for OpenZFS file system.

Use the **Quick create** option to rapidly and easily create a file system with the default root volume configuration. This configuration automatically creates one root volume named `fsx` with a path of `/fsx`, a record size of 128 KiB, and an `NFS exports` setting in which **Client addresses** is an asterisk (`*`) and **NFS options** is `rw,crossmnt`. With these settings, any clients permitted by your VPC and security group settings can access the volume with read and write permissions. The file system data is encrypted at rest using your default service manages AWS KMS key, named `aws/fsx/(default)`.

Use the **Standard create** option to create a file system with a customized root volume configuration. For a list of the file system properties that you can customize, see Configurable file system properties. We recommend using **Standard create** only when you are familiar with FSx for OpenZFS file systems and volumes.

Quick create (recommended)

To create a file system using Quick create

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

2. On the dashboard, choose **Create file system** to start the file system creation wizard.
3. On the **Select file system type** page, choose **Amazon FSx for OpenZFS**, and then choose **Next**. The **Create OpenZFS file system** page appears. For **Creation method**, choose **Quick create**. To create a file system using the **Standard create** method, see Creating an Amazon FSx for OpenZFS file system.
4. In the **Quick configuration** section, for **File system name - optional**, enter 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 these special characters: **+ -** (hyphen) **= . _** (underscore) **: /**.
5. For **Storage class**, select **Intelligent-Tiering (elastic)** or **SSD (provisioned)**.
   - **Intelligent-Tiering (elastic)** offers fully elastic storage that is suitable for most workloads, as well as an optional SSD read cache that provides SSD latencies for reads of frequently accessed data. With Intelligent-Tiering, you are billed for the data you store, depending on the size of your dataset, and do not need to specify a file system size. Intelligent-Tiering is only supported for Multi-AZ (HA) file systems.
   - **SSD (provisioned)** provides low-latency access to your data. With SSD storage, you are billed for the amount of storage that you provision.
6. For **Deployment type**, select **Multi-AZ (HA)** or **Single-AZ (HA)**.
   - **Multi-AZ (HA)** file systems offer high availability and high durability by replicating your data and supporting failover across multiple Availability Zones in the same AWS Region, with a separate copy of your data in each availability zone. Failover typically completes within 60 seconds.
   - **Single-AZ (HA)** file systems offer high availability by deploying a primary and standby file system within the same Availability Zone to deliver continuous availability in the event of failover and failback. Failover typically completes within 60 seconds. Single-AZ (HA) is only available for file systems using the SSD (provisioned) storage class.

   We recommend using Multi-AZ (HA) for most production workloads. We recommend using Single-AZ (HA) for workloads that require consistent single-AZ latencies and as a cost-effective solution for workloads that do not require the high levels of durability that Multi-AZ (HA) provides. For more information on how to choose between deployment types, see Availability by AWS Region and File system performance.
7. For **Throughput capacity** (Intelligent-Tiering only), select the desired throughput capacity of your file system, in MBps. For file systems using the Intelligent-Tiering storage class, the

minimum throughput capacity is 160 MBps. To specify throughput capacity for a file system using the SSD (provisioned) storage class, please create a file system using **Standard Create**.

8. For **SSD read cache sizing mode** (Intelligent-Tiering only), select either **Automatic**, **Custom**, or **None**. Automatic is selected by default. With this option, Amazon FSx automatically chooses a read cache size based on your provisioned throughput. If you know the approximate size of your active working dataset, you can select Custom to customize the size of the SSD read cache. If your workload is not latency-sensitive, you can also choose None to create an elastic file system without an SSD cache and reduce costs.
9. For **SSD storage capacity** (SSD (provisioned) only), specify the storage capacity of your file system, in gibibytes (GiBs). Enter any whole number in the range of 64–524,288.
10. For **Virtual Private Cloud (VPC)**, choose the Amazon VPC that you want to associate with your file system.
11. For **Subnet**, choose the subnet in which your file system's elastic network interface resides.
12. Choose **Next**.
13. Review the file system configuration shown on the **Create OpenZFS file system** page. For your reference, note which file system settings you can modify after the file system is created.
14. Choose **Create file system**.

Standard create

To create a file system using Standard create

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 OpenZFS** , and then choose **Next**. The **Create file system** page appears.
4. For **Creation method**, choose **Standard create**.

   Begin your configuration with the **File system details** section.
5. For **File system name - optional**, enter 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 these special characters: + - = . _ : /
6. For **Storage class**, select **Intelligent-Tiering (elastic)** or **SSD (provisioned)**.

- **Intelligent-Tiering (elastic)** offers fully elastic, cost-effective storage that is suitable for most workloads, as well as an optional SSD read cache that provides SSD latencies for reads of frequently accessed data. With Intelligent-Tiering, you are billed for the data you store, depending on the size of your dataset, and do not need to specify a file system size. Intelligent-Tiering is only supported for Multi-AZ (HA) file systems.
- **SSD (provisioned)** provides high performance with low-latency access to your data. With SSD storage, you are billed for the amount of storage that you provision.

7. For **Deployment type**, select **Multi-AZ (HA)**, **Single-AZ (HA)**, or **Single-AZ (non-HA)**.
   - **Multi-AZ (HA)** file systems offer high availability and high durability by replicating your data and supporting failover across multiple Availability Zones in the same AWS Region, with a separate copy of your data in each Availability Zone. Failover typically completes within 60 seconds.
   - **Single-AZ (HA)** file systems offer high availability by deploying a primary and standby file system within the same Availability Zone to deliver continuous availability in the event of failover and failback. Failover typically completes within 60 seconds. Single-AZ (HA) is only available on file systems using the SSD (provisioned) storage class.
   - **Single-AZ (non-HA)** file systems ensure self-healing recovery within a single Availability Zone by automatically detecting and addressing component failures. Recovery typically completes within 30 minutes. Single-AZ (non-HA) is only available on file systems using the SSD (provisioned) storage class.

   We recommend using Multi-AZ (HA) for most production workloads. We recommend using Single-AZ (HA) for workloads that require consistent single-AZ latencies and as a cost-effective solution for workloads that do not require the high levels of durability that Multi-AZ (HA) provides. For more information on how to choose between deployment types, see Availability by AWS Region and File system performance.
8. For **SSD storage capacity** (SSD (provisioned) only), specify the storage capacity of your file system, in gibibytes (GiBs). Enter any whole number in the range of 64–524,288.
9. For **Provisioned SSD IOPS** (SSD (provisioned) only), you have two options to provision the number of IOPS for your file system:
   - Choose **Automatic** (the default) if you want Amazon FSx to automatically provision 3 IOPS per GB of SSD storage.
   - Choose **User-provisioned** if you want to specify the number of IOPS, up to the maximum for your file system. You pay for SSD IOPS that you provision above 3 IOPS per GB of SSD storage.

10. **Throughput capacity** is the sustained speed at which the file server that hosts your file system can serve data. For **Throughput capacity**, choose from two options to provide your desired throughput capacity in megabytes per second (MBps).

    - Choose the default **Recommended throughput capacity** if you want Amazon FSx to automatically choose the throughput capacity. The recommended value is based on the storage capacity that you choose.
    - Choose **Specify throughput capacity** if you want to specify the throughput capacity value.
      - For Multi-AZ and SINGLE_AZ_2 file systems, valid values are 160, 320, 640, 1280, 2560, 3840, 5120, 7680, or 10240 MBps.
      - For SINGLE_AZ_1 file systems, valid values are 64, 128, 256, 512, 1024, 2048, 3072, or 4096 MBps.

      You pay for throughput capacity that you provision that exceeds the recommended amount.

    You can increase the amount of throughput capacity as needed at any time after you create the file system. For more information, see Modifying throughput capacity.

11. For **SSD read cache sizing mode** (Intelligent-Tiering only), select either **Automatic**, **Custom**, or **None**. Automatic is selected by default. With this option, Amazon FSx automatically chooses a read cache size based on your provisioned throughput. If you know the approximate size of your active working dataset, you can select Custom to customize the size of the SSD read cache. If your workload is not latency-sensitive, you can also choose None to create an elastic file system without an SSD cache and reduce costs.

12. In the **Network & security** section, provide networking and security group information:

    - For **Virtual Private Cloud (VPC)**, choose the Amazon VPC that you want to associate with your file system.
    - For **VPC Security Groups**, the ID for the default security group for your VPC should already be populated.
    - For **Subnet**, choose any value from the list of available subnets. If you are creating a Multi-AZ file system, also choose a **Standby subnet** for the standby file server.
    - (Multi-AZ only) For **Select route tables**, specify the VPC route tables in which rules for routing traffic to the correct file server will be created. Select all VPC route tables associated with the subnets in which your clients are located. By default, Amazon FSx selects your VPC's default route table.

- (Multi-AZ only) **Endpoint IP address range** specifies the IP address range in which the endpoints to access your file system are created. You have three options for the endpoint IP address range:
  - **Unallocated IP address range from your VPC** – Amazon FSx chooses a block of 16 available IP addresses from the VPC's CIDR range to use as the endpoint IP address range for the file system.
  - **Floating IP address range outside your VPC** – Amazon FSx chooses a 198.19.x.0/24 address range.
  - **Enter an IP address range** – You can provide a CIDR range of your own choosing. The IP address range that you choose can either be inside or outside the VPC's IP address range, as long as it doesn't overlap with any subnet.

13 In the **Encryption** section, for **Encryption key**, choose the AWS Key Management Service (AWS KMS) encryption key that protects your file system's data at rest.

14 For **Root volume configuration**, you can set the following options for the file system's root volume:

- For **Data compression type**, choose the type of compression to use for your volume—either **Zstandard**, **LZ4**, or **No compression**. Zstandard compression provides more data compression and higher read throughput than LZ4 compression. LZ4 compression provides less compression and higher write throughput performance than Zstandard compression. For more information about the storage and performance benefits of the volume data compression options, see Data compression.
- For **Copy tags to snapshots**, choose whether to copy tags to the volume's snapshot.
- For **NFS exports**, you can modify or remove the default client configuration setting. Client configurations determine client access and permissions for the volume.

  To provide additional client configurations:

  a. In the **Client addresses** field, specify which clients can access the volume. Enter an asterisk (*) for any client, a specific IP address, or a CIDR range of IP addresses.

  b. In the **NFS options** field, enter a comma-delimited set of export options. For example, enter `rw` to allow read and write permissions to the volume for the specified **Client addresses**.

  c. Choose **Add client configuration**.

  d. Repeat the procedure to add another client configuration.

  For more information, see NFS exports.

- For **Record size**, choose whether to use the default suggested record size of 128 KiB, or to set a custom suggested record size for the volume. Workloads that write in fixed small or large record sizes might benefit from setting a custom record size, such as database workloads (small record size) or media streaming workloads (large record size). We recommend using the default setting in most cases. For more information about setting record size, see Configurable volume properties.
- For **User and group quotas**, you can set a storage quota for a user or group:

  a. For **Quota type**, choose USER or GROUP.

  b. For **User or group ID**, choose the ID number for the user or group.

  c. For **Usage quota**, choose the storage quota number for the user or group.

  d. Choose **Add quota**.

  e. Repeat the procedure to add a quota for another user or group.

15. In **Backup and maintenance - *optional***, you can set the following options:

    - For **Daily automatic backup**, choose **Enabled** for automatic daily backups. This option is enabled by default.
    - For **Daily automatic backup window**, set 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.
    - For **Automatic backup retention period**, set a period from 1–90 days to retain automatic backups.
    - For **Weekly maintenance window**, you can set 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.

16. For **Tags - *optional***, you can enter a key and value to add tags to your file system. A tag is a case-sensitive key-value pair that helps you manage, filter, and search for your file system.

    Choose **Next**.

17. Review the file system configuration on the **Create file system** page. Note which file system settings you can modify after the file system is created.

18. Choose **Create file system**.

After your file system is created, you can create additional volumes as needed to organize your data. Any new volumes that you create will be children of the root volume. For more information on how to create additional volumes, see Creating an Amazon FSx for OpenZFS volume.

Step 2: Mount your file system from an Amazon EC2 instance

Once you have created your file system, you can access the data stored within it by mounting individual volumes on your client from an Amazon Elastic Compute Cloud (Amazon EC2) instance. FSx for OpenZFS supports a wide variety of compute instances and operating systems using the Network File System (NFS) protocol (v3, v4.0, v4.1, and v4.2), including Amazon EC2 instances running Linux, macOS, and Microsoft Windows.

The following instructions detail how to mount a volume from an Amazon EC2 instance on a Linux, macOS, or Windows client. Note that you can also view and copy the exact commands needed to mount any FSx for OpenZFS volume by choosing **Attach** on the details page for that volume in the Amazon FSx console.

> **Note**
>
> The commands to mount a volume require the DNS name of the file system in which the volume is created. To identify a file system's DNS name in the Amazon FSx console, choose **File systems**, then choose the FSx for OpenZFS file system whose volume you are mounting. The **DNS name** will be displayed in the **Network & security** panel. This information can also be found in the response of the DescribeVolumes API operation.

Linux client

To mount a volume from an Amazon EC2 instance on Linux

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Create or select an Amazon EC2 instance running Amazon Linux 2 that is in the same virtual private cloud (VPC) as your file system. For more information about launching an instance, see Step 1: Launch an instance in the *Amazon EC2 User Guide*.
3. Connect to your Amazon EC2 Linux instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide*.
4. Open a terminal on your Amazon EC2 instance using secure shell (SSH), and log in with the appropriate credentials.

5. If you are using CentOS, RedHat, or Ubuntu, install the NFS client. This step is not necessary if you are using the latest version of the Amazon Linux 2.
   - For CentOS and RedHat use the following command: **sudo yum –y install nfs-utils**
   - For Ubuntu use this command: **sudo apt-get -y install nfs-common**
6. Create a directory on your Amazon EC2 instance for the volume's local mount path with the following command. In the following example, replace *fsx* with your desired location.

   ```
   sudo mkdir /fsx
   ```

7. Use the following `mount` command to mount your Amazon FSx for OpenZFS file system to the directory that you created. Replace the following:
   - Replace `nfs-version` with an NFS protocol version, such as `4.2`.
   - Replace `fs-dns-name` with the DNS name or the IP address of the file system.
   - Replace `volume-path` with the path of the volume to mount. For example, use `/fsx` to mount the root volume or a path such as `/fsx/sales` to mount the top-level `fsx/sales` directory.
   - Replace `local-mount-path` with the directory path of your local mount path, such as `/fsx` for the directory you created in step 5.

   ```
   sudo mount -t nfs -o nfsvers=nfs-version fs-dns-name:volume-path local-mount-path
   ```

   The following example uses sample values.

   ```
   sudo mount -t nfs -o nfsvers=4.2 fs01234567.fsx.us-east-1.amazonaws.com:/fsx /fsx
   ```

   You can also use the IP address of the file system instead of its DNS name.

   ```
   sudo mount -t nfs -o nfsvers=4.2 198.51.100.5:/fsx /fsx
   ```

If you have issues with your Amazon EC2 instance (such as connections timing out), see Troubleshoot EC2 instances in the *Amazon EC2 User Guide*.

macOS client

To mount a volume from an Amazon EC2 instance on macOS

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Create or select an Amazon EC2 Mac instance running the macOS that is in the same VPC as the file system.

   For more information on launching an instance, see Step 1: Launch an instance in the *Amazon EC2 User Guide*.
3. Connect to your Amazon EC2 Mac instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide*.
4. Open a terminal on your EC2 Mac instance using secure shell (SSH), and log in with the appropriate credentials.
5. Create a directory on the EC2 instance for mounting the volume as follows:

   ```
   sudo mkdir /localpath
   ```
6. Mount the volume using the following command.

   ```
   sudo mount -t nfs -o resvport file-system-dns-name:/vol_path mount-point
   ```

   The following example uses sample values.

   ```
   sudo mount -t nfs -o resvport fs-01234567890abcde5.fsx.us-east-1.amazonaws.com:/
   fsx/vol1 /fsx
   ```

Windows

To mount a volume from an Amazon EC2 instance on Windows

> **Note**
>
> Mounting FSx for OpenZFS volumes to Windows clients leverages the NFS v3 protocol. The following instructions include the necessary steps to install the NFS client on your Windows-based EC2 instance.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Create or select an Amazon EC2 instance running Microsoft Windows that is in the same VPC as the file system.

   For more information on launching an instance, see Step 1: Launch an instance in the *Amazon EC2 User Guide*.
3. Connect to your Amazon EC2 Windows instance. For more information, see Connecting to your Windows instance in the *Amazon EC2 User Guide*.
4. Open PowerShell as an administrator, and install the NFS client.

   ```
   Install-WindowsFeature -Name NFS-Client
   ```

   If prompted to do so, restart and reconnect to your Windows instance.
5. Open a command prompt window with standard user privileges. If you run the mount command as Administrator, the mounted drive will not appear in File Explorer.

   > **Note**
   >
   > To ensure this mounted drive appears in File Explorer, please open the Command Prompt window with standard user privileges. If you run this command as Administrator, it will not appear in File Explorer.
6. You can mount the drive using a command prompt, or using a Powershell path

   a. Mount the volume to any available drive letter by running the following command, replacing Z: with any available drive letter:

      - Replace *filesystem-dns-name* with the DNS name or the IP address of the file system.
      - Replace *vol_path* with the path of the FSx for OpenZFS volume you are trying to mount.
      - Replace *Z:* with any available drive letter.

      ```
      mount \\filesystem-dns-name\vol_path Z:
      ```

      The following example uses sample values.

```
mount \\fs-01234567890abcdef1.fsx.us-east-1.amazonaws.com\fsx\vol1 Z:
```

b. You can also mount the file system using the following Powershell path:

```
New-PSDrive -Name "Z" -PSProvider "FileSystem" -Root "\\filesystem-dns-
name\" -Persist
```

The following example uses a sample file system DNS name.

```
New-PSDrive -Name "Z" -PSProvider "FileSystem" -Root "\
\fs-0239c0e31af65bff1.fsx.us-east-1.amazonaws.com\" -Persist
```

Step 3: Clean up your resources

Follow these steps to clean up your resources, delete your file system as needed, and protect your AWS account.

To clean up your resources and delete your file system

1. On the Amazon EC2 console, terminate your instance. For more information, see Terminate Your Instance in the *Amazon EC2 User Guide.*
2. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
3. On the Amazon FSx console, delete your file system. When you delete a file system, all volumes and automatic backups are deleted automatically. However, you still must delete any manually created backups. The following steps outline this process.

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

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

   c. 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.

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

e. Choose **Delete file system**.

f. 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. Any automatic backups are deleted along with the file system.

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

h. From the dashboard, choose any backups that have the same **File system ID** as the file system that you deleted, and choose **Delete backup**. Be sure to retain the final backup, if you created one.

i. The **Delete backups** dialog box opens. Keep the check box selected for the IDs of the backups that you want to delete, and then choose **Delete backups**.

Your Amazon FSx file system and any related automatic backups are now deleted, along with any manual backups that you chose to delete as well.

Availability by AWS Region

Amazon FSx for OpenZFS is available in the following AWS Regions, with deployment type and storage class support indicated for each region:

AWS Region	Single-AZ 1 (non-HA) SSD	Single-AZ 2 (non-HA) SSD	Single-AZ 1 (HA) SSD	Single-AZ 2 (HA) SSD	Multi-AZ (HA) SSD	Multi-AZ (HA) Intelligent-Tiering
US East (N. Virginia)*	✓	✓		✓	✓	✓
US East (Ohio)	✓	✓		✓	✓	✓
US West (N. California)	✓		✓		✓	
US West (Oregon)*	✓	✓		✓	✓	✓
AWS GovCloud (US-West)	✓				✓	
AWS GovCloud (US-East)	✓				✓	
Asia Pacific (Hong Kong)	✓		✓		✓	
Asia Pacific (Tokyo)	✓	✓		✓	✓	✓

AWS Region	Single-AZ 1 (non-HA) SSD	Single-AZ 2 (non-HA) SSD	Single-AZ 1 (HA) SSD	Single-AZ 2 (HA) SSD	Multi-AZ (HA) SSD	Multi-AZ (HA) Intelligent-Tiering
Asia Pacific (Seoul)	✓		✓		✓	
Asia Pacific (Osaka)	✓				✓	
Asia Pacific (Singapore)*	✓	✓		✓	✓	✓
Asia Pacific (Sydney)	✓	✓		✓	✓	✓
Asia Pacific (Jakarta)	✓				✓	
Asia Pacific (Malaysia)	✓		✓		✓	
Asia Pacific (Mumbai)	✓	✓		✓	✓	✓
Asia Pacific (Hyderabad)	✓				✓	
Canada (Central)*	✓	✓		✓	✓	✓
Canada West (Calgary)*	✓				✓	

AWS Region	Single-AZ 1 (non-HA) SSD	Single-AZ 2 (non-HA) SSD	Single-AZ 1 (HA) SSD	Single-AZ 2 (HA) SSD	Multi-AZ (HA) SSD	Multi-AZ (HA) Intelligent-Tiering
Europe (Milan)	✓				✓	
Europe (Spain)	✓				✓	
Europe (Frankfurt)	✓	✓		✓	✓	✓
Europe (Zurich)	✓				✓	
Europe (Ireland)	✓	✓		✓	✓	✓
Europe (London)	✓		✓		✓	
Europe (Paris)	✓				✓	
Europe (Stockholm)	✓		✓		✓	
Middle East (UAE)	✓				✓	
Middle East (Bahrain)	✓		✓		✓	

AWS Region	Single-AZ 1 (non-HA) SSD	Single-AZ 2 (non-HA) SSD	Single-AZ 1 (HA) SSD	Single-AZ 2 (HA) SSD	Multi-AZ (HA) SSD	Multi-AZ (HA) Intelligent-Tiering
South America (São Paulo)	✓				✓	
Israel (Tel Aviv)	✓				✓	
Africa (Cape Town)	✓				✓	

Note

*Due to differences in infrastructure capabilities and configurations, your file system type may be unavailable in specific Availability Zones within these regions.

Accessing your data

You can access data on your FSx for OpenZFS file systems within the AWS Cloud and from on premise environments using a variety of supported clients. You can also use AWS Container Services such as Amazon ECS with your FSx for OpenZFS volumes to access your data.

Topics

- Accessing your data through volumes
- Accessing your data within the AWS Cloud
- Accessing your data from on-premises
- Accessing your data using AWS container services

Accessing your data through volumes

The primary way to access data on your file system is through mounting individual volumes from an Amazon EC2 instance. This section provides details on how to configure your file system to automatically remount volumes on an Amazon EC2 instance when the instance reboots, and tips for mounting a volume to maximize your file systems overall performance. For detailed instructions on how to mount a volume to a Linux, macOS, or Windows client, see Step 2: Mount your file system from an Amazon EC2 instance.

Topics

- Automatically mounting file systems on reboot for Linux instances
- Additional mounting options to maximize file system performance

Automatically mounting file systems on reboot for Linux instances

You can use the `/etc/fstab` file, which contains information about your file systems, to automatically remount your volumes on an Amazon EC2 Linux instance when the instance reboots. The command `mount -a`, which runs during instance start-up, mounts the file systems listed in `/etc/fstab`.

> **Note**
>
> FSx for OpenZFS file systems do not support automatic mounting using `/etc/fstab` on Amazon EC2 Mac instances.

To automatically mount your file system on reboot

1. Connect to your EC2 instance:

   - To connect to your instance from a computer running macOS or Linux, specify the .pem file for your SSH command. To do this, use the `-i` option and the path to your private key.
   - To connect to your instance from a computer running Windows, you can use either MindTerm or PuTTY. To use PuTTY, install it and convert the .pem file to a .ppk file.

   For more information, see the following topics in the *Amazon EC2 User Guide*:

   - Connecting to your Linux instance using SSH
   - Connecting to your Linux instance from Windows using PuTTY
2. Create a local directory that will be used to mount the FSx for OpenZFS volume.

   ```
   sudo mkdir /fsx
   ```
3. Open the `/etc/fstab` file in an editor of your choice.
4. Add the following line to the `/etc/fstab` file. Insert a tab character between each parameter. It should appear as one line with no line breaks.

   ```
   filesystem-dns-name:volume-path /localpath nfs vers=nfs-version,defaults 0 0
   ```

   The last three parameters indicate NFS options (which we set to default), dumping of file system and filesystem check (these are typically not used so we set them to 0).
5. Save the changes to the file.
6. Test the fstab entry by using the mount command with the `fake all verbose` options.

   ```
   sudo mount -fav
                           fs-dns-name:/vol_path      : successfully mounted
   ```

Now mount the volume using the following command. The next time the EC2 instance restarts, the volume will be mounted automatically.

```
sudo mount /localpath
sudo mount filesystem-dns-name:/volume-path
```

Your EC2 instance is now configured to mount the FSx for OpenZFS volume whenever it restarts.

Additional mounting options to maximize file system performance

You can also include the following options when mounting a volume to improve your file system's overall performance.

- `rsize=1048576` – Sets the maximum number of bytes of data that the NFS client can receive for each network READ request. This value applies when reading data from a file on an FSx for OpenZFS volume. We recommend that you use the largest size possible, 1048576. Due to lower memory capacity on file systems with 64 MBps and 128 MBps of provisioned throughput, these file systems will only accept a maximum rsize of 262144 and 524288 bytes, respectively.
- `wsize=1048576` – Sets the maximum number of bytes of data that the NFS client can send for each network WRITE request. This value applies when writing data to a file on an FSx for OpenZFS volume. We recommend that you use the largest size possible, 1048576. Due to lower memory capacity on file systems with 64 MBps and 128 MBps of provisioned throughput, these file systems will only accept a maximum wsize of 262144 and 524288 bytes, respectively.
- `timeo=600` – Sets the timeout value that the NFS client uses to wait for a response before it retries an NFS request to 600 deciseconds (60 seconds).
- `_netdev` – When present in `/etc/fstab`, prevents the client from attempting to mount the FSx for OpenZFS volume until the network has been enabled.

The following example uses sample values.

```
sudo mount -t nfs -o rsize=1048576,wsize=1048576,timeo=600
 fs-01234567890abcdef1.fsx.us-east-1.amazonaws.com:/fsx/vol1 /fsx
```

Accessing your data within the AWS Cloud

Amazon VPC helps you to launch AWS resources into a virtual network that you define. This virtual network closely resembles a traditional network that you operate in your own data center, with the benefits of using the scalable infrastructure of AWS. For more information, see What is Amazon VPC in the *Amazon Virtual Private Cloud User Guide*.

Each Amazon FSx file system is associated with a Virtual Private Cloud (VPC). You can access your FSx for OpenZFS file system from anywhere in the same VPC within which it is deployed regardless of the Availability Zone (AZ). You can also access your file system from other VPCs. These VPCs can be in different accounts or regions. In addition to any requirements listed in the following sections for accessing FSx for OpenZFS resources, you also need to ensure that your file system's VPC security group has the correct settings. It needs to allow data to flow between your file system and any clients that connect to it. For more information, see Amazon VPC security groups.

Topics

- Access from within the same VPC
- Access from a different VPC

Access from within the same VPC

When you create your Amazon FSx for OpenZFS file system, you select the Amazon VPC in which it is located. All volumes associated with the FSx for OpenZFS file system are also located in the same VPC. When the file system and the client mounting the volume are located in the same VPC and AWS account, you can mount a volume using the file system's DNS name over the NFS protocol. For more information, see Step 2: Mount your file system from an Amazon EC2 instance.

You can achieve better performance and avoid data transfer charges by accessing an FSx for OpenZFS volume using a client in the same Availability Zone as the file system's subnet. To identify a file system's subnet, choose **File systems** in the Amazon FSx console, then choose the FSx for OpenZFS file system whose volume you are mounting. The subnet or preferred subnet (Multi-AZ) is displayed in the **Subnet** or **Preferred subnet** panel.

Accessing a Single-AZ file system using a client located in a different Availability Zone results in data transfer charges. There are no data transfer charges for accessing a Multi-AZ file system from any Availability Zone in the same region.

Access from a different VPC

The process of accessing your data from an AWS Region outside of the file system's VPC differs between Single-AZ and Multi-AZ file systems, as Multi-AZ file systems utilize a floating IP address. The following sections describe how to access your file systems from a different VPC depending on deployment type.

Accessing Single-AZ file systems

You can access your FSx for OpenZFS file system from compute instances in a different VPC, AWS account, or AWS Region from that associated with your file system by 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 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 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 Virtual Private Cloud 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, see Work with transit gateways in the *Amazon VPC Transit Gateways*.

Accessing Multi-AZ file systems

The NFS endpoints on FSx for OpenZFS Multi-AZ file systems use floating IP addresses so that connected clients seamlessly transition between the preferred and standby file servers during a failover event. For more information about failovers, see Failover process for FSx for OpenZFS.

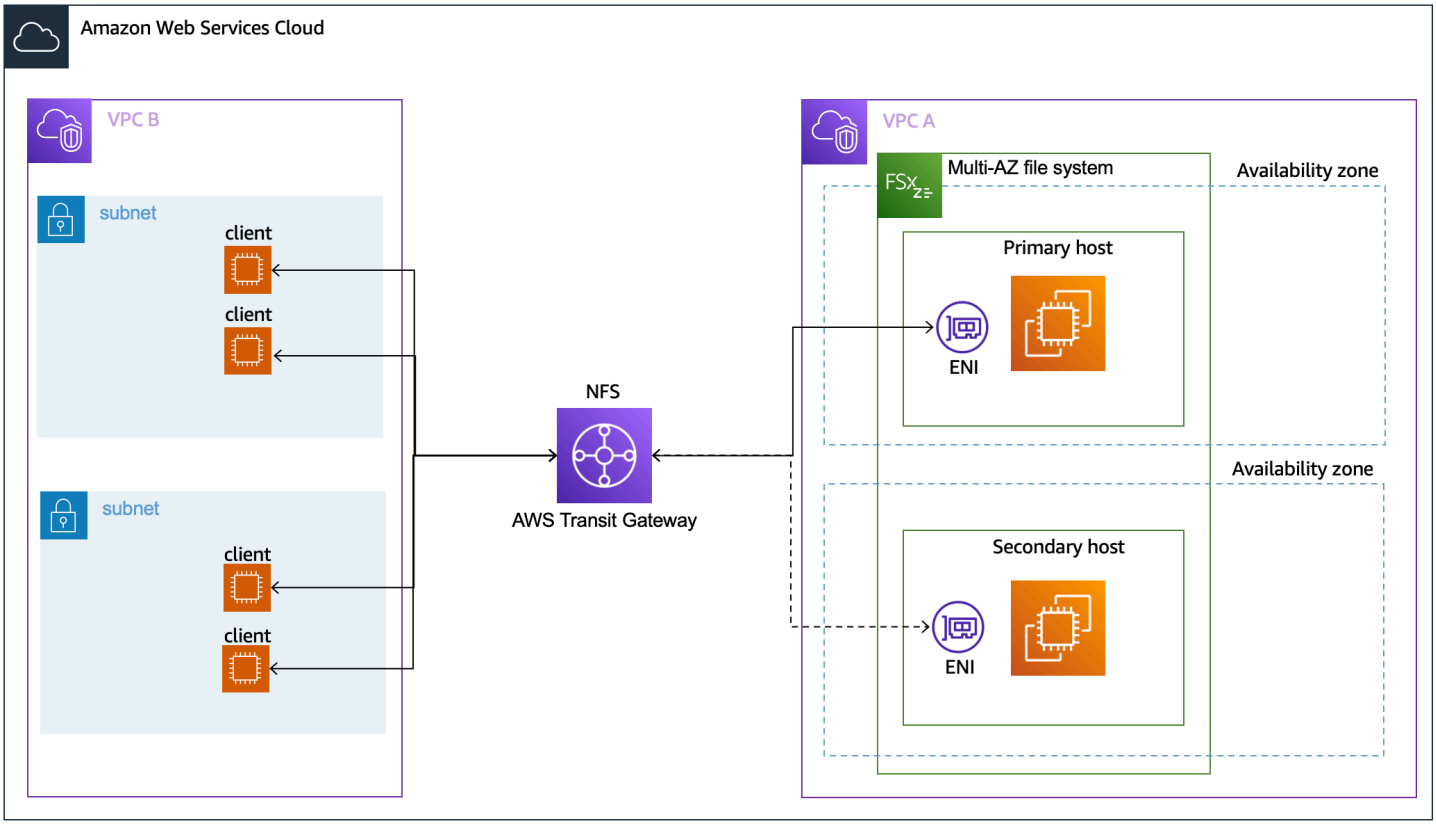
When you create a file system, you can optionally specify the endpoint IP address range in which these floating IP addresses are created. By default, the Amazon FSx API selects a CIDR block of 16 available addresses from within the VPC's CIDR ranges. Additionally, you can optionally specify the VPC route tables in which rules for routing traffic to the correct file server will be created. By default, the Amazon FSx API selects the VPC's default route table.

Only AWS Transit Gateway supports routing to floating IP addresses, which is also known as transitive peering. VPC Peering, AWS Direct Connect, and AWS VPN don't support transitive

peering. Therefore, you are required to use Transit Gateway in order to access these interfaces from networks that are outside of your file system's VPC.

When you access your Multi-AZ file system from outside of the file system's VPC, FSx for OpenZFS will manage routing configurations as long as the file system's `EndpointIpAddressRange` is within the CIDR range of the file system's VPC and does not overlap with the CIDR range of any subnets in the VPC. However, if you access your Multi-AZ file system from outside of the file system's VPC, and the file system's `EndpointIpAddressRange` is outside of the CIDR range of the file system's VPC, you will need to set up additional routing in Transit Gateway. For information on how to configure Transit Gateway to access your FSx for OpenZFS file system, see Configuring routing using AWS Transit Gateway.

The following diagram illustrates using Transit Gateway for NFS access to a Multi-AZ file system that is in a different VPC than the clients that are accessing it.



Note

Ensure that all of the route tables you're using are associated with your Multi-AZ file system. Doing so helps prevent loss of availability during a failover. For information about

associating your Amazon VPC route tables with your file system, see Updating an Amazon FSx for OpenZFS file system.

Configuring routing using AWS Transit Gateway

If you have a Multi-AZ file system with an `EndpointIPAddressRange` that's outside your VPC's CIDR range, you need to set up additional routing in your AWS Transit Gateway to access your file system from peered or on-premises networks. No additional Transit Gateway configuration is required for Single-AZ file systems or Multi-AZ file systems with an `EndpointIPAddressRange` that's within your VPC's IP address range.

Important

To access a Multi-AZ file system using a Transit Gateway, each of the Transit Gateway's attachments must be created in a subnet whose route table is associated with your file system.

To configure routing using AWS Transit Gateway

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose the FSx for OpenZFS file system for which you are configuring access from a peered network.
3. In **Network & security** copy the **Endpoint IP address range**.
4. Add a route to Transit Gateway that routes traffic destined for this IP address range to your file system's VPC. For more information, see Work with transit gateways in the *Amazon VPC Transit Gateways*.
5. Confirm that you can access your FSx for OpenZFS file system from the peered network.

To add the route table to your file system, see Updating an Amazon FSx for OpenZFS file system.

Note

DNS records for the NFS endpoints are only resolvable from within the same VPC as the file system. In order to mount a volume or connect to a management port from another

network, you need to use the endpoint's IP address. These IP addresses do not change over time.

Accessing your data from on-premises

FSx for OpenZFS supports the use of AWS Direct Connect or AWS VPN to access your file systems from your on-premises compute instances. Using AWS Direct Connect, you access your file system over a dedicated network connection from your on-premises environment. Using AWS VPN, you 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 about AWS Direct Connect, see What is AWS Direct Connect? in the *AWS Direct Connect User Guide*. For more information on setting up AWS VPN connections, see VPN connections in the *Amazon VPC User Guide*.

Topics

- Accessing Multi-AZ file systems

Accessing Multi-AZ file systems

Amazon FSx requires that you use AWS Transit Gateway to access Multi-AZ file systems from an on-premises network. In order to support failover across AZs for Multi-AZ file systems, Amazon FSx uses floating IP addresses for the interfaces used for NFS endpoints. Because the NFS endpoints use floating IPs, you must use AWS Transit Gateway in conjunction with AWS Direct Connect or AWS VPN to access these interfaces from an on-premises network. The floating IP addresses used for these interfaces are within the `EndpointIpAddressRange` you specify when creating your Multi-AZ file system. By default, the Amazon FSx API selects a CIDR block of 16 available addresses from within the VPC's CIDR ranges. The floating IP addresses are used to enable a seamless transition of your clients to the standby file system in the event a failover is required. For more information, see Failover process for FSx for OpenZFS.

If you have a Multi-AZ file system with an `EndpointIPAddressRange` that's outside your VPC's CIDR range, you need to set up additional routing in your AWS Transit Gateway to access your file system from peered or on-premises networks. For information, see Configuring routing using AWS Transit Gateway.

Accessing your data using AWS container services

In addition to Amazon EC2, you can also access your Amazon FSx for OpenZFS file systems through Amazon Elastic Container Service and Amazon Elastic Kubernetes Service. The following section provides instructions on how to mount your file system from an Amazon ECS Docker container on an Amazon EC2 Linux instance using a bind mount. To use Amazon EKS clusters to manage the life cycle of your file systems and volumes, see the Amazon FSx for OpenZFS CSI Driver.

> **Note**
>
> Using the CSI Driver is not supported for file systems using the Intelligent-Tiering storage class.

Topics

- Mounting your file system from an Amazon ECS container

Mounting your file system from an Amazon ECS container

You can access your Amazon FSx for OpenZFS file systems from an Amazon Elastic Container Service (Amazon ECS) Docker container on an Amazon EC2 Linux instance by mounting volumes using a bind mount. For more information, see Bind mounts in the *Amazon Elastic Container Service Developer Guide*.

To mount a volume on an Amazon ECS Linux container

1. Create an ECS cluster using the EC2 Linux + Networking cluster template for your Linux containers. For more information, see Clusters in the *Amazon ECS Developer Guide*.
2. Create a directory on the EC2 instance for mounting the volume as follows:

   ```
   sudo mkdir /fsxopenzfs
   ```

3. Mount your FSx for OpenZFS volume on the Linux EC2 instance by either using a user-data script during instance launch, or by running the following commands:

   ```
   sudo mount -t nfs -o nfsvers=NFS_version file-system-dns-name:/volume-path
    /localpath
   ```

The following example uses sample values in the mount command.

```
sudo mount -t nfs -o nfsvers=4.1 fs-01234567890abcdef1.fsx.us-
east-1.amazonaws.com:/fsx/vol1 /fsxopenzfs
```

You can also use the file system's IP address instead of its DNS name.

```
sudo mount -t nfs -o nfsvers=4.1 198.51.100.1:/fsx/vol1 /fsxopenzfs
```

4. When creating your Amazon ECS task definitions, add the following `volumes` and `mountPoints` container properties in the JSON container definition. Replace the `sourcePath` with the mount point and directory in your FSx for OpenZFS file system.

```
{
    "volumes": [
        {
            "name": "openzfs-volume",
            "host": {
                "sourcePath": "mountpoint"
            }
        }
    ],
    "mountPoints": [
        {
            "containerPath": "container_local_path",
            "sourceVolume": "openzfs-volume"
        }
    ],
    .
    .
    .
}
```

Availability and durability for Amazon FSx for OpenZFS

FSx for OpenZFS supports three file system deployment types—Multi-AZ (HA), Single-AZ (HA), and Single-AZ (non-HA)—that offer different levels of availability and durability. 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. For information on which deployment types and storage classes are supported in each AWS Region, see Availability by AWS Region.

Topics

- Choosing a deployment type based on availability and durability
- Failover process for FSx for OpenZFS
- Working with file system resources

Choosing a deployment type based on availability and durability

FSx for OpenZFS offers two high-availability (HA) deployment options and two non-HA deployment options. Each option provides different levels of availability and durability, that is described in detail in the following sections. Use this information to help you decide which file system deployment type provides the level of availability and durability that is best suited for your workload.

Multi-AZ (HA)

Multi-AZ (HA) file systems are composed of a high-availability pair of file servers spread across two Availability Zones—a preferred Availability Zone and a standby Availability Zone. Each of the Availability Zones also contains a set of storage volumes. Relative to Single-AZ (non-HA and HA) file systems, Multi-AZ (HA) file systems provide enhanced durability by replicating data synchronously both within each individual Availability Zone and between the two Availability Zones. Multi-AZ (HA) file systems also offer enhanced availability by automatically failing over to the standby Availability Zone during planned system maintenance, and in cases of unplanned service disruption, typically within 60 seconds. This allows you to continue accessing your data, and helps to protect your data against instance failure and Availability Zone disruption.

Multi-AZ (HA) file systems are recommended for business critical production workloads and database workloads that require storage that is highly available and durable across Availability Zones in the same AWS Region.

Single-AZ (HA)

Single-AZ (HA) file systems are composed of two file servers connected to a single set of storage volumes within a single Availability Zone. During planned maintenance events or unplanned service disruptions, Single-AZ (HA) file systems fail over from the primary file server to the standby file server, typically within 60 seconds. Amazon FSx continuously monitors the file servers for any hardware failure and automatically replaces a server in the event of failure while serving data from the standby file server. Once the primary server is restored, the Single-AZ (HA) file systems fail back to the primary server.

Single-AZ (HA) file systems are recommended as a cost-effective solution for workloads that need high availability, but don't need storage redundancy across Availability Zones, including analytics, and EDA workloads.

Single-AZ (non-HA)

Single-AZ (non-HA) are composed of a single file server instance and a set of storage volumes within one Availability Zone. Amazon FSx continuously monitors for hardware failures, and automatically recovers from failure events by replacing the failed infrastructure component. Single-AZ (non-HA) 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. Single-AZ (non-HA) file systems are designed as a cost-efficient solution for workloads that can tolerate temporary loss of availability in the event of planned file system maintenance, unplanned service disruption or, in rare cases, the loss of data updates since the most recent backup.

> **Note**
>
> For all Single-AZ file systems, file system failure may be unrecoverable in rare cases, such as storage component failures. In these cases, you can recover your file system from the most recent backup.

In addition to differences in availability and durability, different deployment types also offer varying levels of performance that may impact your decision. Note that both Single-AZ (non-HA

and HA) deployment types, FSx for OpenZFS also support Single-AZ 1 and Single-AZ 2. Single-AZ 2 offers higher levels of performance than the maximum offered by Single-AZ 1. For more information on performance differences between deployment types, see Performance for Amazon FSx for OpenZFS.

You can also migrate between deployment types by restoring from a backup, using **rsync**, or using on-demand replication. For more information, see Migrating between deployment types and storage classes.

Failover process for FSx for OpenZFS

Multi-AZ (HA) and Single-AZ (HA) file systems automatically fail over from the preferred file server to the standby file server under the following conditions:

- The preferred file server becomes unavailable.
- The file system's throughput capacity is changed.
- The preferred file server undergoes planned maintenance.

Multi-AZ (HA) file systems will also fail over if an Availability Zone disruption occurs.

When failing over from one file server to another, the new active file server automatically begins serving all file system read and write requests. A failover typically takes less than 60 seconds from the detection of the failure on the active file server to the promotion of the standby file server to active status. Upon completion of the failover, you continue to have access to your data without manual intervention. When the preferred file server is fully recovered and becomes available, Amazon FSx automatically fails back to it, usually in less than 60 seconds.

Testing failover on Multi-AZ (HA) and Single-AZ (HA) file systems

You can test failover on your Multi-AZ (HA) or Single-AZ (HA) 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 servers sequentially. File systems automatically fail over to the secondary server while Amazon FSx replaces the preferred file server first. After the update, 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. For more information about modifying your file system's throughput capacity and monitoring the progress of the request, see Modifying throughput capacity.

Working with file system resources

Each Amazon FSx for OpenZFS file system has a number of resources, including subnets, elastic network interfaces, IP addresses, and backups, that allow Amazon FSx to offer greater availability and durability. The sections below provide information on how these resources work, as well as recommendations for how to configure and manage them.

Subnets

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

When you create a Multi-AZ (HA) file system, you specify two subnets, one for the preferred file server, and one for the standby file server. The two subnets you choose must be in different Availability Zones within the same AWS Region. For more information about Amazon VPC, see What is Amazon VPC? in the *Amazon VPC user guide*.

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

For Multi-AZ (HA) file systems, Amazon FSx provisions two ENIs—one in each of the subnets that you associate with your file system. Clients communicate with your Amazon FSx file system using the elastic network interface that's attached to the file server that serves the data. Network interfaces are considered to be within the service scope of Amazon FSx, despite being part of your account's VPC. For Single-AZ (HA) file systems, Amazon FSx provisions two ENIs in the subnet that you associate with your file system. For Single-AZ (non-HA) file systems, Amazon FSx provisions one elastic network interface (ENI) in the subnet that you associate with your file system.

Warning

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

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

File system deployment type	Number of subnets	Number of elastic network interfaces	Number of IP addresses
Multi-AZ (HA)	2	2	3
Single-AZ (HA)	1	2	3
Single-AZ (non-HA)	1	1	1

Once a file system is created, its IP addresses don't change until the file system is deleted. For Multi-AZ (HA) and Single-AZ (HA) file systems, the number of IP addresses includes a floating IP address, which allows connected clients to transition between the preferred and standby file servers during a failover event. For more information, see Accessing your data.

Important

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

Backups

FSx for OpenZFS offers a native backups feature that's designed to support archival, data retention, and compliance needs. A backup is a secondary, offline copy of your file system. Amazon FSx backups are crash-consistent and incremental, which means that only the changes from your most

recent backup are saved. This saves on backup storage costs by not duplicating data. By default, Amazon FSx takes an automatic daily backup of your file system during a backup window that you specify. You can create additional backups at any time using the AWS Management Console, AWS Command Line Interface, or Amazon FSx API. For more information, see Protecting your data with backups.

Performance for Amazon FSx for OpenZFS

Amazon FSx for OpenZFS provides simple, high-performance file storage. In this section, we provide an overview of FSx for OpenZFS performance for all deployment types, and describe how your file system configuration impacts key performance dimensions. We also include some important tips and recommendations for maximizing the performance of your file system.

Topics

- File system performance
- Choosing a deployment type based on performance
- Choosing a storage class based on performance
- Migrating between deployment types and storage classes
- General tips for maximizing performance
- Monitoring performance
- How FSx for OpenZFS file systems work with SSD storage
- How FSx for OpenZFS file systems work with Intelligent-Tiering

File system performance

File system performance is typically measured in latency, throughput, and I/O operations per second (IOPS). Amazon FSx for OpenZFS offers three deployment options, Multi-AZ (HA), Single-AZ (HA), and Single-AZ (non-HA). Both Single-AZ (non-HA and HA) deployment types, also support Single-AZ 1 and Single-AZ 2. Single-AZ 2 offers higher levels of performance than the maximum offered by Single-AZ 1. Each deployment option offers a different performance profile. In this section, we document the performance you can expect for frequently accessed data from the in-memory or NVMe caches and data accessed from disk for both deployment types. We also document the baseline performance you can always deliver, as well as the burst performance you can drive for short periods of time.

The specific level of performance a file system can provide is defined by its *provisioned throughput capacity*, which determines the size of the file server hosting the file system. Provisioned throughput capacity is equivalent to the baseline disk throughput supported by your file server. For data access from disks, your file system's performance is also dependent on the number of *provisioned SSD disk IOPS* configured for the file system's underlying disks. Note that the actual

level of performance you can drive for your workload depends on a variety of factors. For more information, see Tips for maximizing performance.

Choosing a deployment type based on performance

Both Single-AZ (non-HA) and Single-AZ (HA) offer two tiers of performance with Single-AZ 1 and Single-AZ 2. Single-AZ 2 (HA) is recommended for most use cases, given the higher level of both performance and availablity that it provides. Single-AZ 2 file systems offer double the performance scalability as compared to Single-AZ 1, delivering up to 400,000 IOPS and 10 GBps throughput for both reads and writes to persistent SSD storage.

In addition, Single-AZ 2 file systems include an up to 2.5 TB high-speed NVMe read cache that automatically caches your most recently-accessed data, making that data accessible at millions of IOPS and with latencies of a few hundred microseconds. Single-AZ 2 file systems are suitable for high-performance workloads such as media processing and rendering, financial analytics, and machine learning. Single-AZ 2 file systems are also appropriate for read-heavy workloads with frequently accessed datasets.

In addition to Single-AZ file systems, Amazon FSx for OpenZFS also offers Multi-AZ file systems that offer higher levels of availability and durability in addition to the same levels of performance as Single-AZ 2. For more information on Multi-AZ (HA) file systems and choosing between deployment types, see Availability and durability for Amazon FSx for OpenZFS.

For information on which deployment types are supported in each AWS Region, see Availability by AWS Region

Choosing a storage class based on performance

Your file system's performance also depends on its storage class. FSx for OpenZFS offers two storage classes, Intelligent-Tiering (elastic) and SSD (provisioned).

With Intelligent-Tiering, your file system has fully elastic, low-cost storage and access to a built-in SSD-backed write log for low-latency writes and an optional provisioned SSD read cache for low-latency reads. Intelligent-Tiering is recommended for most Network Attached Storage (NAS) datasets to simplify storage management and reduce costs while offering performance levels comparable to file systems using the SSD (provisioned) storage class for most workloads. Intelligent-Tiering is suitable for home directories, analytics, and other project-based workloads.

For more information on performance for file systems that use Intelligent-Tiering, see How FSx for OpenZFS file systems work with Intelligent-Tiering.

With SSD (provisioned) storage, your file system provides low-latency access to your full dataset. SSD (provisioned) is recommend for datasets that require the performance of all-flash storage across all data. SSD (provisioned) is suitable for databases and non-cache-friendly electronic design automation workloads. For more information on performance for file systems that use SSD (provisioned) storage, see How FSx for OpenZFS file systems work with SSD storage.

For information on which storage classes are supported in each AWS Region, see Availability by AWS Region

Migrating between deployment types and storage classes

Once you create your file system, you cannot change its deployment type or storage class. However, there are several options that you can use to migrate data from your pre-existing file system to a new file system with your desired deployment type or storage class.

- **Restoring from backup** - You can create a new Single-AZ 2 file system by restoring from a backup of your Single-AZ 1 file system and choosing the desired deployment type. You can also create a Single-AZ (HA) file system from a Single-AZ (non-HA) file system. You cannot create a Multi-AZ file system from a Single-AZ backup or migrate between storage classes by restoring from a backup. For more information, see Restoring backups.
- **Using on-demand replication** - You can use on-demand replication to synchronize data between file systems with different deployment types or storage classes. For more information, see Protecting your data with on-demand replication.

For more information on how to migrate your data, see Migrating your existing file storage to Amazon FSx for OpenZFS.

General tips for maximizing performance

FSx for OpenZFS file systems are designed to deliver the maximum performance of your file system across your clients in aggregate, whether you are supporting data access from a single client, or thousands of clients. The following sections provide some practical tips on how to maximize client performance.

Client considerations

Amazon EC2 instances

When launching the Amazon EC2 instances that will work with your FSx for OpenZFS file system, ensure that they can support the level of performance your file system needs to deliver. Ensure they have the compute, memory, and network capacity sufficient to drive the throughput, IOPS, and latencies provided by your FSx for OpenZFS file system.

To determine your EC2 instance's compute and memory capacity, see Instance types in the *Amazon EC2 User Guide for Linux Instances*. To determine its network capacity, see Amazon EC2 instance network bandwidth in the same guide. The performance characteristics of FSx for OpenZFS file systems don't depend on the use of Amazon EC2–optimized instances.

NFS nconnect

With FSx for OpenZFS, NFS clients can use the `nconnect` mount option to have multiple TCP connections (up to 16) associated with a single NFS mount. Such an NFS client multiplexes file operations onto multiple TCP connections (multi-flow) in a round-robin fashion to obtain improved performance beyond single TCP connection (single-flow) limits. For more information on single-flow limits, see Amazon EC2 instance network bandwidth in the *Amazon EC2 User Guide for Linux Instances*.

The following command demonstrates how to use the `nconnect` mount option to mount an FSx for OpenZFS volume with a maximum of 16 simultaneous connections:

```
sudo mount -t nfs -o nconnect=16 filesystem_dns_name:/vol_path /localpath
```

The `nconnect` mount option is supported for all NFS versions (v3, v4.0, v4.1, v4.2). NFS `nconnect` is supported by default in Linux kernel versions 5.3 and above, including the latest Ubuntu 18.04 LTS. In addition, RHEL8.3 supports `nconnect` by way of a backport into the `4.18.0-240.e18` kernel and newer.

NFS v3

FSx for OpenZFS file systems flexibly support multiple versions of the NFS protocol (v3, v4.0, v4.1, v4.2). While more recent versions of NFS can better support simultaneous access from many clients (due to a more robust file-locking mechanism) and client-side caching, NFS v3 may still provide improved latency, throughput, and IOPS performance for performance-sensitive workloads. You

can mount using NFS v3 from Linux, Windows, or macOS EC2 instances. For more information, see Step 2: Mount your file system from an Amazon EC2 instance.

The following example illustrates how to specify NFS v3 when mounting an FSx for OpenZFS volume:

```
sudo mount -t nfs -o nfsvers=3 fs-dns-name:/vol_path /local_path
```

NFS delegations

To improve the ability of NFS clients to cache data locally, NFS v4 introduced NFS delegations, or the ability of the server to delegate certain responsibilities to the client. If the client is granted a read delegation, it is assured that no other client has the ability to write to the file for the duration of the delegation, meaning that the client can read from its local copy instead of having to go back to the file server.

FSx for OpenZFS file systems support NFS v4 file read delegations. To take advantage of this capability, ensure your clients are mounting with NFS v4.0 or higher.

Request model

When you mount your file system, asynchronous writes are enabled by default (that is, `-o async`). With asynchronous writes, pending write operations are buffered on the client before they are written to your Amazon FSx file system, enabling lower latencies for these operations. A client that has enabled synchronous writes (that is, `-o sync`), or one that opens files using an option that bypasses the cache (for example, `O_DIRECT`), issues synchronous requests, which means that every operation incurs a round-trip between your client and the file server. We recommend using the default asynchronous write option to maximize client performance.

Other recommended mount options

To improve the performance of your file system, you can also configure the following options when mounting your file system:

- `rsize=1048576` – Sets the maximum number of bytes of data that the NFS client can receive for each network READ request to 1048576 bytes (1 MB). Due to lower memory capacity on file systems with 64 MBps and 128 MBps of provisioned throughput, these file systems will only accept a maximum `rsize` of 262144 and 524288 bytes, respectively.
- `wsize=1048576` – Sets the maximum number of bytes of data that the NFS client can send for each network WRITE request to 1048576 bytes (1 MB). Due to lower memory capacity on file

systems with 64 MBps and 128 MBps of provisioned throughput, these file systems will only accept a maximum `wsize` of 262144 and 524288 bytes, respectively.

- `timeo=600` – Sets the timeout value that the NFS client uses to wait for a response before it retries an NFS request to 600 deciseconds (60 seconds).
- `_netdev` – When present in **/etc/fstab**, prevents the client from attempting to mount the FSx for OpenZFS volume until the network has been enabled.

The following example uses sample values.

```
sudo mount -t nfs -o rsize=1048576,wsize=1048576,timeo=600
 fs-01234567890abcdef1.fsx.us-east-1.amazonaws.com:/fsx/vol1 /fsx
```

File system and volume configurations

Storage capacity utilization

As the amount of used storage space gets closer to the total available storage capacity, Amazon FSx (like other file systems) spends more time finding suitable places to store new files and their metadata. This leads to higher latency for operations that modify files, which can negatively impact overall performance. To avoid this performance impact, we recommended keeping storage utilization of SSD (provisioned) file systems below 80% of the total capacity. If needed, you can increase your maximum storage capacity at anytime, without disruption to your end users or applications. For more information, see Modifying provisioned SSD storage capacity and IOPS.

You do not need to modify storage capacity if your file system uses the Intelligent-Tiering storage class. For more information, see How FSx for OpenZFS file systems work with Intelligent-Tiering.

Provisioned throughput capacity and in-memory cache

In addition to defining the throughput and IOPS that a file system can deliver, a file system's provisioned throughput capacity also determines the amount of in-memory cache on your file server. Increasing your file system's throughput capacity improves workload performance in two ways.

First, it increases the throughput and IOPS you can drive from disk (disk I/O) and from in-memory cache. Second, by increasing the amount of in-memory cache, you can store more data in your file server's in-memory cache, which drives higher cached performance for larger workloads.

Some request- or metadata-intensive workloads will also benefit from a larger file server in-memory cache. These types of workloads can generate and store a large volume of metadata in the in-memory cache. To ensure the size of your file server's in-memory cache is not a bottleneck for your file system performance, we recommend provisioning at least 128 MBps of throughput capacity for these types of workloads.

NFS export options (sync and async)

On the file server side, the `sync` or `async` NFS export option can impact performance. (This is distinct from the similarly-named option you use when mounting your FSx for OpenZFS volume on your client.) This option determines whether your file server will acknowledge client I/O requests as complete when they are written to the file server's in-memory cache (`async`), or only after they are committed to the file server's persistent disks (`sync`). `sync` is the default option and is generally recommended for most workloads.

If you have performance-intensive workloads that can use an FSx for OpenZFS volume as temporary storage for shorter-term data processing or workloads that are resilient to data loss, you can use the `async` option to achieve substantially higher performance. Because an FSx for OpenZFS volume exported with the `async` option will acknowledge client writes before they are committed to durable disk storage, clients can write data to the file server at a significantly faster rate. However, this performance comes at the cost of losing data from acknowledged writes that have not yet been committed to the server's disks, in the event of a file server crash.

Data compression

For read-heavy workloads, compression can significantly improve the overall throughput performance of your file system because it reduces the amount of data that needs to be sent between the underlying storage and the file server. FSx for OpenZFS volumes support the following data compression algorithms.

- *Zstandard compression* delivers very high levels of on-disk data compression, with higher read throughput and reduced write throughput performance than LZ4 compression.
- *LZ4 compression* delivers higher write throughput performance, but achieves lower levels of data compression than Zstandard compression.

With data compression, you can improve your read throughput on data accessed from disk up to the same levels you deliver for frequently accessed cached data. The specific improvement depends upon the amount by which compression can reduce the size of your dataset. Your effective

throughput will be roughly equivalent to the product of your provisioned disk throughput and your compression ratio (defined as the ratio of the size of the compressed data to the size of the uncompressed data). For the highest provisioned throughput level (4096 MBps), common Z-Standard compression ratios of 2-3x can increase your effective read throughput by up to 8-12 GBps.

You can change a volume's data compression to improve performance. Changing this property affects only newly-written data on the volume.

ZFS record size

The ZFS record size specifies a suggested block size for files in the volume. This property is designed solely for use with databases and other workloads that access files in fixed-size records. ZFS automatically tunes block sizes according to internal algorithms optimized for typical access patterns. When you create a volume, the default record size for file systems using the Intelligent-Tiering storage class is 1024 KiB. The default for all other file systems is 128 KiB. General purpose workflows perform well using the default record size, and we don't recommend changing it, as it may adversely affect performance.

For database workflows that create very large files but access them in small random chunks, specifying a record size greater than or equal to the record size of the database can result in significant performance gains. For databases that use a fixed disk block or record size for I/O, set the ZFS record size to match it. See Dataset record size in the OpenZFS documentation for more information.

Streaming workflows such as multimedia and video can benefit from setting a larger record size than the default value. For more information about setting the record size on a volume, see Managing Amazon FSx for OpenZFS volumes.

You can change a volume's record size to make performance improvements. Changing the volume record size affects only files created afterward; existing files are unaffected.

Monitoring performance

Every minute, FSx for OpenZFS emits usage metrics to Amazon CloudWatch and you can use these metrics to help identify opportunities to improve the performance your clients can drive from your file system.

You can investigate aggregate file system performance with the `Sum` statistic of each metric. For example, the `Sum` of the `DataReadBytes` statistic reports the total read throughput by file system

or volume, and the `Sum` of the `DataWriteBytes` statistic reports the total write throughput by file system or volume.

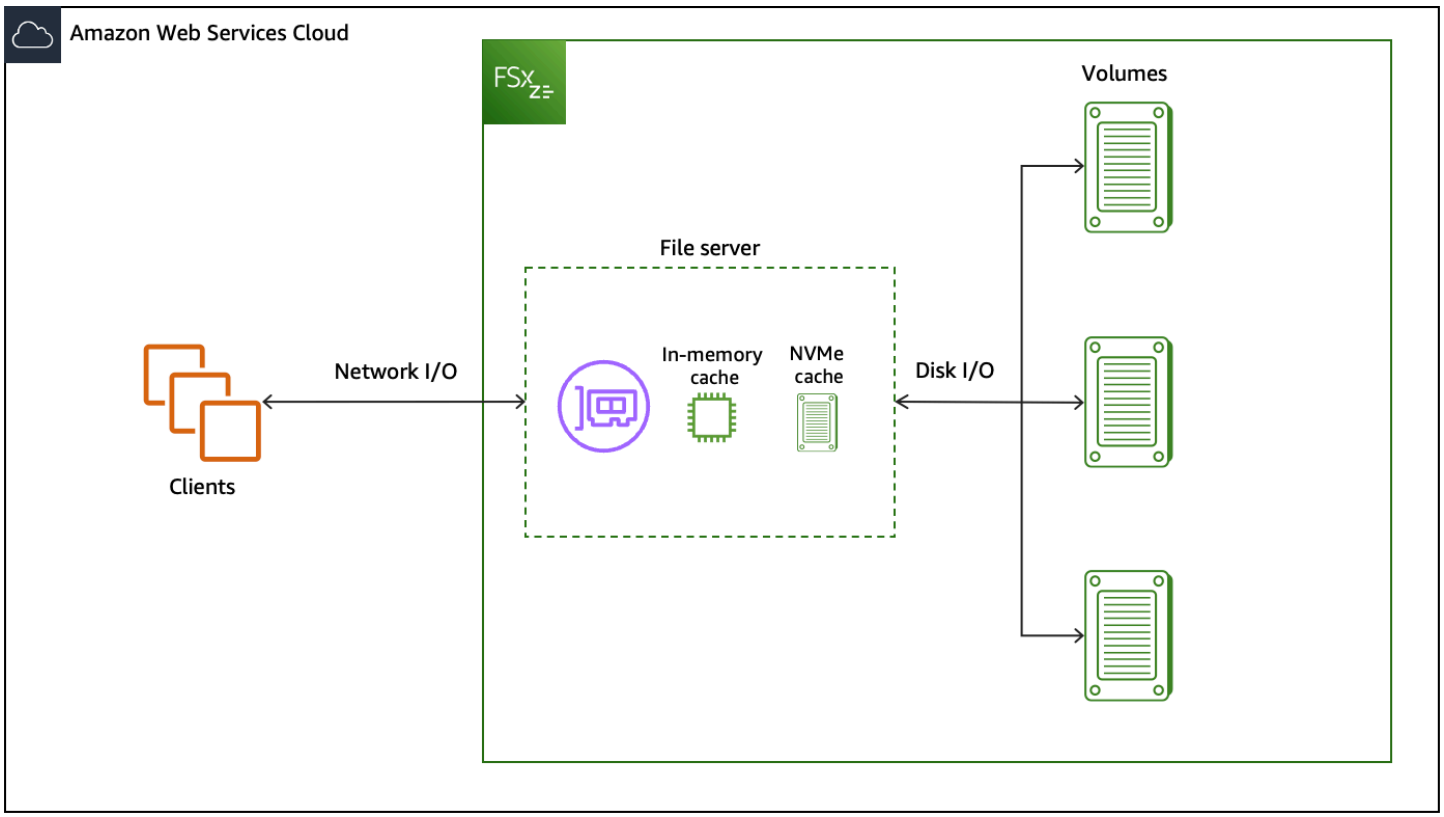
For more information on monitoring your file system's performance, see Monitoring with Amazon CloudWatch.

How FSx for OpenZFS file systems work with SSD storage

FSx for OpenZFS file systems that use the SSD (provisioned) storage class consist of a file server that clients communicate with and a set of disks attached to that file server. Each file server employs a fast, in-memory cache to enhance performance for the most frequently accessed data. In addition to the in-memory cache, Single-AZ 2 file systems also provide an additional Non-volatile Memory Express (NVMe) cache for storing up to terrabytes of frequently accessed data. FSx for OpenZFS utilizes the Adaptive Replacement Cache (ARC) and L2ARC that are built into the OpenZFS file system, which improves the portion of data access driven from the in-memory and NVMe caches.

When a client accesses data that's stored in either the in-memory or NVMe caches, the data is served directly to the requesting client as *network I/O*, without the file server needing to read it from disk. When a client accesses data that is not in either of these caches, it is read from disk as *disk I/O* and then served to the client as network I/O; data read from disk is also subject to the IOPS and bandwidth limits of the underlying disks.

FSx for OpenZFS file systems can serve network I/O about three times faster than disk I/O, which means that clients can drive greater throughput and IOPS with lower latencies for frequently accessed data in cache. The following diagram illustrates how data is accessed from an FSx for OpenZFS file system, with the NVMe cache applying to all Single-AZ 2 file systems.



File-based workloads are typically spiky, characterized by short, intense periods of high I/O with plenty of idle time between bursts. To support spiky workloads, in addition to the *baseline* speeds that a file system can sustain 24/7, Amazon FSx provides the capability to *burst* to higher speeds for periods of time for both network I/O and disk I/O operations. Amazon FSx uses a network I/O credit mechanism to allocate throughput and IOPS based on average utilization — file systems accrue credits when their throughput and IOPS usage is below their baseline limits, and can use these credits when they perform I/O operations.

Topics

- Data access from cache
- Data access from disk
- SSD IOPS and performance

Data access from cache

For read access directly from the in-memory ARC or NVMe L2ARC cache, performance is primarily defined by two components: the performance supported by the client-server network I/O

connection, and the size of the cache. The following tables show the cached read performance of all Single-AZ 1, all Single-AZ 2, and Multi-AZ (HA) file systems, based on AWS Region.

> **Note**
>
> Single-AZ 1 (HA) and Single-AZ 2 (HA) file systems are only available in a certain subset of AWS Regions. For more information on which AWS Regions support Single-AZ 1 (HA) and Single-AZ 2 (HA) file systems, see Availability by AWS Region.

Single-AZ 1 (US West (N. California), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Middle East (UAE), AWS GovCloud (US-West), AWS GovCloud (US-East))

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Maximum network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
64	3	97	1,562	Tens of thousands of IOPS
128	11.2	195	1,562	
256	22.4	390	1,562	
512	44.8	781	1,562	Hundreds of thousands of IOPS
1,024	89.6	1,562	–	
2,048	179.2	3,125	–	
3,072	268.8	4,687	–	
4,096	358.4	6,250	–	Up to 1 million IOPS

Single-AZ 1 (All other AWS Regions)

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Maximum network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
64	0.25	200	3,200	Tens of thousands of IOPS
128	1.0	400	3,200	
256	3.0	800	3,200	
512	11.2	1,600	3,200	Hundreds of thousands of IOPS
1,024	22.4	3,200	–	
2,048	44.8	6,400	–	
3,072	67.2	9,600	–	
4,096	89.6	12,800	–	1 million IOPS

Single-AZ 2 (All AWS Regions)

Provisioned throughput capacity (MBps)	In-memory cache (GB)	NVMe L2ARC cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
			Baseline	Burst	
160	3	40	375	3,125	Tens of thousands of IOPS
320	11.2	80	775	3,750	

Provisioned throughput capacity (MBps)	In-memory cache (GB)	NVMe L2ARC cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
640	22.4	160	1,550	5,000	Hundreds of thousands of IOPS
1,280	44.8	320	3,125	6,250	
2,560	89.6	640	6,250	–	
3,840	134.4	960	9,375	–	
5,120	179.2	1,280	12,500	–	1+ million IOPS
7,680	268.8	1,920	18,750	–	
10,240	358.4	2,560	21,000	–	

Multi-AZ (US West (N. California), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Middle East (UAE) AWS GovCloud (US-West), AWS GovCloud (US-East))

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
160	11.2	195	1,562	Tens of thousands of IOPS
320	22.4	390	1,562	
640	44.8	781	1,562	Hundreds of thousands of IOPS
1,280	89.6	1,562	–	
2,560	179.2	3,125	–	
3,840	268.8	4,687	–	

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
5,120	358.4	6,250	–	Up to 1 million IOPS

Multi-AZ (US East (N. Virginia), US East (Ohio), US West (Oregon), Europe (Ireland), Europe (Frankfurt), Canada (Central), Asia Pacific (Mumbai), Asia Pacific (Tokyo), Asia Pacific (Sydney), Asia Pacific (Singapore))

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
160	3	375	3,125	Tens of thousands of IOPS
320	11.2	775	3,750	
640	22.4	1,550	5,000	Hundreds of thousands of IOPS
1,280	44.8	3,125	6,250	
2,560	89.6	6,250	–	
3,840	134.4	9,375	–	
5,120	179.2	12,500	–	1+ million IOPS
7,680	268.8	18,750	–	
10,240	358.4	21,000	–	

> **Note**
>
> For Multi-AZ file systems created in Canada (Central) and Asia Pacific (Mumbai) prior to July 9th, 2024, refer to Table 6 for performance details.

Multi-AZ (All other AWS Regions)

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
160	1.0	400	3,200	Tens of thousands of IOPS
320	3	800	3,200	
640	11.2	1,600	3,400	Hundreds of thousands of IOPS
1,280	22.4	3,200	–	
2,560	44.8	6,400	–	
3,840	67.2	9,600	–	
5,120	89.6	12,800	–	1+ million IOPS

Data access from disk

For read and write access from the disks attached to the file server, performance depends on the performance supported by the server's disk I/O connection. Similar to data accessed from cache, the performance of this connection is determined by the provisioned throughput capacity of the file system, which is equivalent to the baseline throughput capacity of your file server.

Single-AZ 1 (All AWS Regions)

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)		Maximum disk IOPS	
	Baseline	Burst	Baseline	Burst
64	64	1,024	2,500	40,000
128	128	1,024	5,000	40,000
256	256	1,024	10,000	40,000
512	512	1,024	20,000	40,000
1,024	1,024	–	40,000	–
2,048	2,048	–	80,000	–
3,072	3,072	–	120,000	–
4,096	4,096	–	160,000	–

Single-AZ 2 (All AWS Regions)

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)		Maximum disk IOPS	
	Baseline	Burst	Baseline	Burst
160	160	3,125	6,250	100,000
320	320	3,125	12,500	100,000
640	640	3,125	25,000	100,000
1,280	1,280	3,125	50,000	100,000

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)		Maximum disk IOPS	
2,560	2,560	–	100,000	–
3,840	3,840	–	150,000	–
5,120	5,120	–	200,000	–
7,680	7,680	–	300,000	–
10,240	10,240*	–	400,000	–

Multi-AZ (US West (N. California), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Middle East (UAE), AWS GovCloud (US-West), AWS GovCloud (US-East))

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)*		Maximum disk IOPS	
	Baseline	Burst	Baseline	Burst
160	160	1,250	6,000	40,000
320	320	1,250	12,000	40,000
640	640	1,250	20,000	40,000
1,280	1,280	–	40,000	–
2,560	2,560	–	80,000	–
3,840	3,840	–	120,000	–
5,120	5,120	–	160,000	–

Note

*Deployment hardware differences in these regions may cause disk throughput capacity to vary by up to 5% from the values shown in this table.

Multi-AZ (US East (N. Virginia), US East (Ohio), US West (Oregon), Europe (Ireland), Europe (Frankfurt), Canada (Central), Asia Pacific (Mumbai), Asia Pacific (Tokyo), Asia Pacific (Sydney), Asia Pacific (Singapore))

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)		Maximum disk IOPS	
	Baseline	Burst	Baseline	Burst
160	160	3,125	6,250	100,000
320	320	3,125	12,500	100,000
640	640	3,125	25,000	100,000
1,280	1,280	3,125	50,000	100,000
2,560	2,560	–	100,000	–
3,840	3,840	–	150,000	–
5,120	5,120	–	200,000	–
7,680	7,680	–	300,000	–
10,240	10,240*	–	400,000	–

Note

*If you have a Multi-AZ (HA) file system with a throughput capacity of 10,240 MBps, performance will be limited to 7,500 MBps for write traffic only. Otherwise, for read traffic on all Multi-AZ (HA) 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.

Note

For Multi-AZ file systems created in Canada (Central) and Asia Pacific (Mumbai) prior to July 9th, 2024, refer to Table 11 for performance details.

Multi-AZ (All other AWS Regions)

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)*		Maximum disk IOPS	
	Baseline	Burst	Baseline	Burst
160	160	1,187	5,000	40,000
320	320	1,187	10,000	40,000
640	640	1,187	20,000	40,000
1,280	1,280	–	40,000	–
2,560	2,560	–	80,000	–
3,840	3,840	–	120,000	–
5,120	5,120	–	160,000	–

Note

*Deployment hardware differences in these regions may cause disk throughput capacity to vary by up to 5% from the values shown in this table.

The previous tables show your file system's throughput capacity for uncompressed data. However, because data compression reduces the amount of data that needs to be transferred as disk I/O, you can often deliver higher levels of throughput for compressed data. For example, if your data is compressed to be 50% smaller (that is, a compression ratio of 2), then you can drive up to 2x the throughput than you could if the data were uncompressed. For more information, see Data compression.

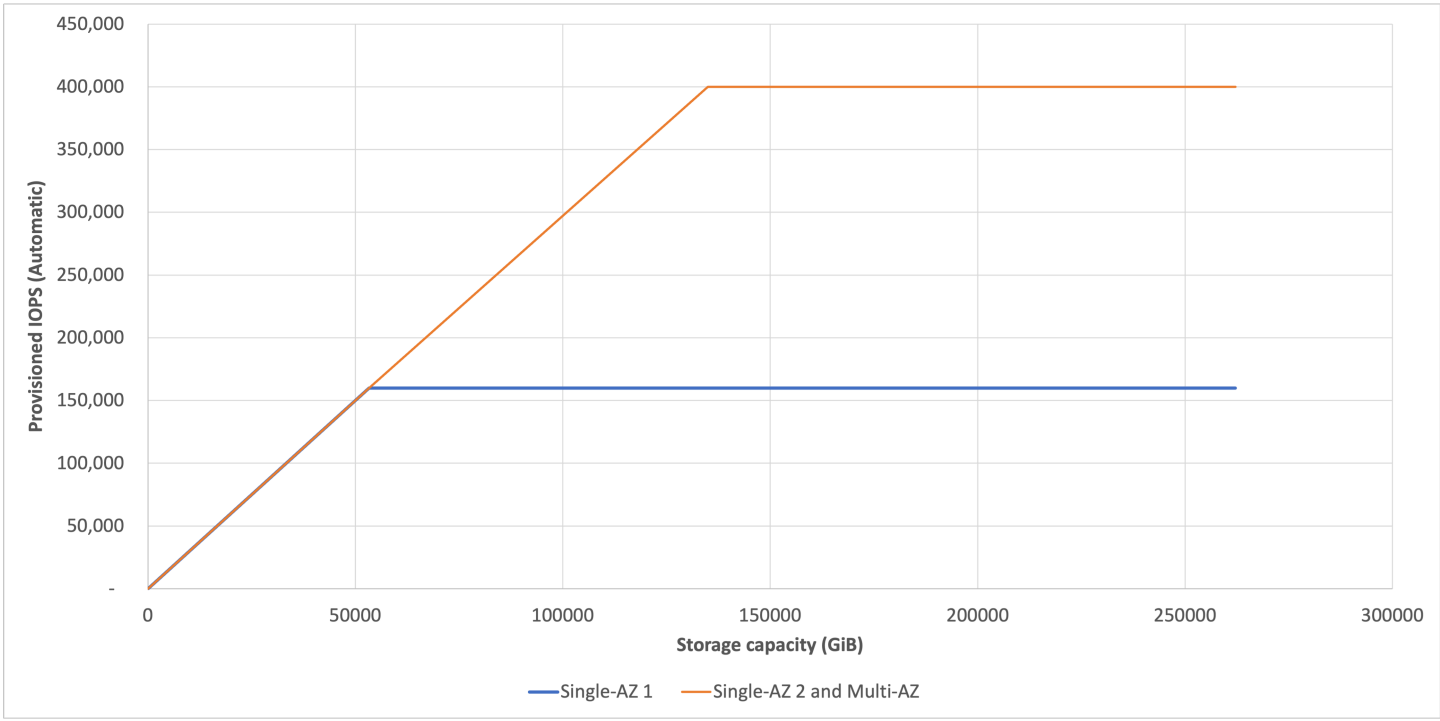
SSD IOPS and performance

Data accessed from disk is also subject to the performance of those underlying disks, which is determined by the number of provisioned SSD IOPS configured on the file system. The maximum IOPS levels you can achieve are defined by the lower of either the maximum IOPS supported by your file server's disk I/O connection, or the maximum SSD disk IOPS supported by your disks. In order to drive the maximum performance supported by the server-disk connection, you should configure your file system's provisioned SSD IOPS to match the maximum IOPS in the table above.

If you select `Automatic` provisioned SSD IOPS, Amazon FSx will provision 3 IOPS per GB of storage capacity up to the maximum for your file system, which is the highest IOPS level supported by the disk I/O connection documented above. If you select `User-provisioned`, you can configure any level of SSD IOPS from the minimum of 3 IOPS per GB of storage, up to the maximum for your file system, as long as you don't exceed 1000 IOPS per GiB*.

> **Note**
>
> *File systems in the following AWS Regions have a maximum IOPS to storage ratio of 50 IOPS per GiB: Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Middle East (UAE), Middle East (Bahrain), Asia Pacific (Osaka), Europe (Milan), Europe (Paris), South America (São Paulo) Region, Israel (Tel Aviv), Asia Pacific (Hong Kong), Asia Pacific (Seoul), Asia Pacific (Mumbai), Canada (Central), Europe (Stockholm), and Europe (London).

The following graph illustrates the maximum IOPS for Single-AZ 1 (non-HA and HA), Single-AZ 2 (non-HA and HA), and Multi-AZ (HA) depending on storage capacity.



How FSx for OpenZFS file systems work with Intelligent-Tiering

The FSx for OpenZFS Intelligent-Tiering storage class offers elastic, low-cost storage for workloads that traditionally run on Network Attached Storage (NAS) file systems. File systems using the Intelligent-Tiering storage class consist of a high-availability pair of file servers that clients communicate with and an in-memory cache within each file server that enhances performance for frequently accessed data. Instead of a set of volumes attached to each file server, however, file systems using the Intelligent-Tiering storage class utilize fully elastic, intelligently tiered, regional block storage which automatically grows and shrinks to fit your workload as it changes. File systems using the Intelligent-Tiering storage class also provide an optional provisioned SSD read cache for low-latency access to frequently accessed data and a built-in SSD-backed write log for low-latency, durable writes. As most workloads tend to be read-heavy and actively work with only a small subset of the overall dataset at any given time, the hybrid model of Intelligent-Tiering storage and an SSD read cache allows file systems using the Intelligent-Tiering storage class to provide storage that performs at levels comparable to SSD (provisioned) file systems for most workloads, while delivering up to 85% cost savings relative to the SSD (provisioned) storage class.

When reading and writing data to an Intelligent-Tiering file system, especially data that hasn't been accessed recently or frequently enough to be in the file server's in-memory cache, performance depends on the size of the SSD read cache. Data access from Intelligent-Tiering storage has time-to-first-byte latencies of roughly tens of milliseconds as well as per-request costs,

while accesses from the SSD read cache return with sub-millisecond latency and no per-request costs.

You can configure your SSD read cache with one of three sizing mode options: Automatic, Custom, or None. With Automatic, Amazon FSx automatically selects an SSD read cache size based on provisioned throughput. With Custom, you can customize the size of your SSD read cache and scale it up or down at any time based on your workload's needs. When configuring the size of the SSD read cache for your file system, you should consider both the size of your frequently accessed dataset within the workload and the workload's sensitivity to higher latency for reads of less-frequently accessed data. Choose None if you do not want to use an SSD read cache with your file system. You can switch between SSD read cache sizing modes after your file system has been created. For more information on how to modify your SSD read cache, see Modifying provisioned SSD read cache.

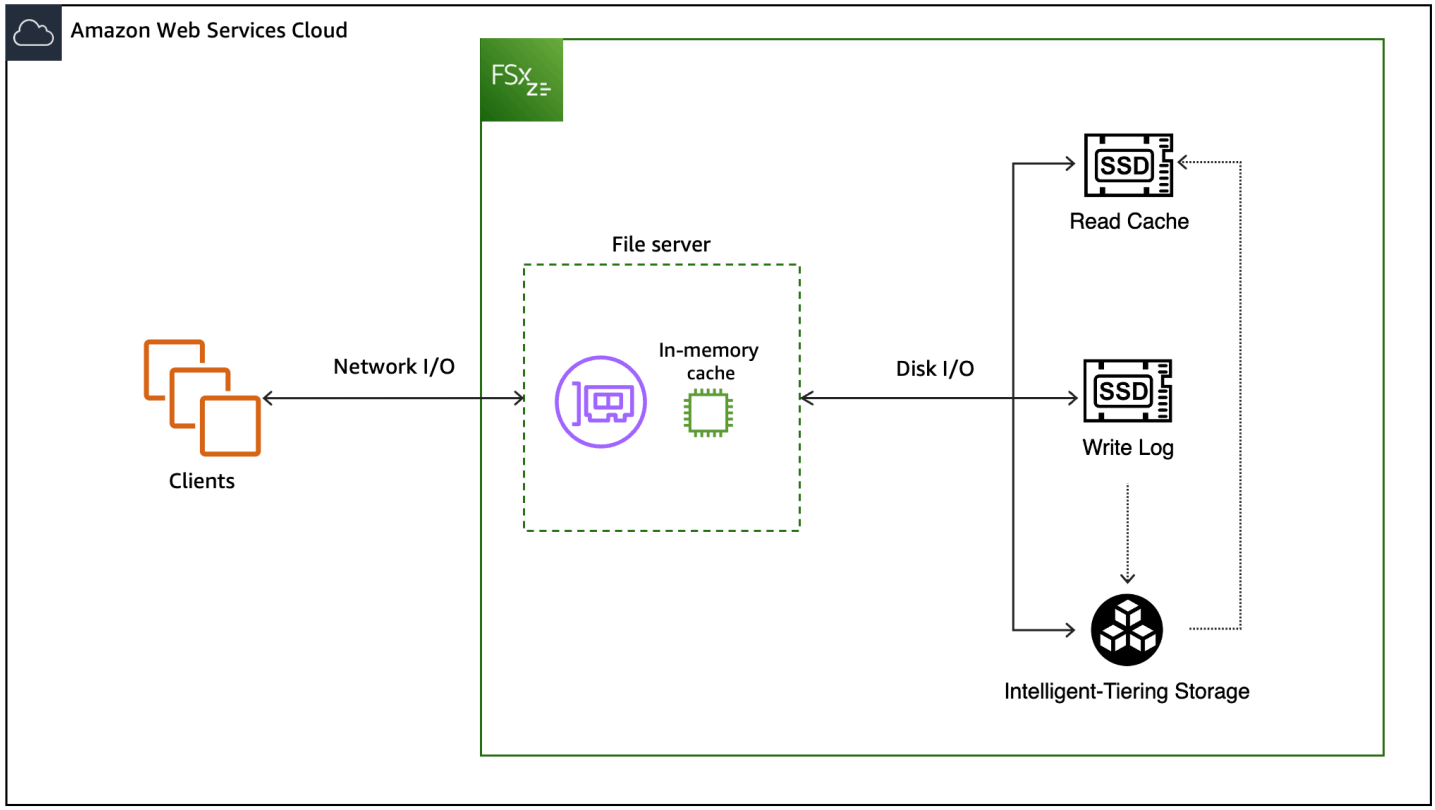
To help determine whether your SSD read cache is sized appropriately, Amazon FSx publishes a cache hit ratio metric, which reports the percentage of reads that are served from the cache. You can view the cache hit ratio metric in the Amazon FSx console on your file system's **Monitoring** tab or with CloudWatch. For most workloads, a cache hit ratio of 80% or more provides the right balance of performance and cost optimization. For more information on how to use CloudWatch to monitor your SSD read cache, see Using Amazon FSx for OpenZFS CloudWatch metrics.

With the Intelligent-Tiering storage class, you do not need to provision IOPS. Instead, you pay for data access based on usage, in the form of read and write requests. A request is a single read or write operation between the file system and the Intelligent-Tiering storage. FSx for OpenZFS automatically optimizes the conversion between read and write I/O to the file system and requests to optimize performance and reduce costs.

A write request occurs when FSx for OpenZFS writes a block of data to Intelligent-Tiering storage. When you write data to the file system, FSx for OpenZFS immediately sends the data to the built-in SSD-based write log. Write requests are then aggregated and written to Intelligent-Tiering storage from the SSD write log, increasing throughput and lowering request costs. Reads can be served from the file server's in-memory cache, SSD read cache, or directly from Intelligent-Tiering storage. When a read is served from Intelligent-Tiering storage, a read request occurs for each block of retrieved data. When you read data sequentially, FSx for OpenZFS will prefetch data to improve performance.

Data from the in-memory cache on file systems using the Intelligent-Tiering storage class is served directly to the requesting client as *network I/O*. When a client accesses data that is not in the in-

memory cache, it is read from either the SSD read cache or Intelligent-Tiering storage as *disk I/O* and then served to the client as network I/O. The following diagram illustrates how data is accessed from an Intelligent-Tiering file system.



Topics

- Data access from in-memory cache
- Data access from SSD cache and Intelligent-Tiering
- Performance considerations

Data access from in-memory cache

For read access directly from the in-memory ARC cache, performance is primarily defined by two components: the performance supported by the client-server network I/O connection, and the size of the cache. The following table shows the in-memory cached read performance of file systems using the Intelligent-Tiering storage class.

Multi-AZ (US East (N. Virginia), US East (Ohio), US West (Oregon), Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Mumbai), Canada (Central), Europe (Frankfurt), Europe (Ireland))

Provisioned throughput capacity (MBps)	In-memory cache (GB)	Network throughput capacity (MBps)		Maximum network IOPS
		Baseline	Burst	
160	3	375	3,125	Tens of thousands of IOPS
320	11.2	775	3,750	
640	22.4	1,550	5,000	Hundreds of thousands of IOPS
1,280	44.8	3,125	6,250	
2,560	89.6	6,250	–	
3,840	134.4	9,375	–	
5,120	179.2	12,500	–	1+ million IOPS
7,680	268.8	18,750	–	
10,240	358.4	21,000	–	

Data access from SSD cache and Intelligent-Tiering

For read and write access from the SSD read cache and Intelligent-Tiering, performance depends on the performance supported by the server's disk I/O connection. Similar to data accessed from cache, the performance of this connection is determined by the provisioned throughput capacity of the file system, which is equivalent to the baseline throughput capacity of your file server.

Multi-AZ (US East (N. Virginia), US East (Ohio), US West (Oregon), Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Mumbai), Canada (Central), Europe (Frankfurt), Europe (Ireland))

Provisioned throughput capacity (MBps)	Maximum disk throughput capacity (MBps)*	Maximum disk IOPS	
		Baseline	Burst
160	160	6,250	100,000
320	320	12,500	100,000
640	640	25,000	100,000
1,280	1,280	50,000	100,000
2,560	2,560	100,000	–
3,840	3,840	150,000	–
5,120	5,120	200,000	–
7,680	7,680	300,000	–
10,240	10,240	400,000	–

Note

*The maximum disk throughput values above refer to read throughput. Maximum write throughput is around 75% of the maximum read throughput for each throughput capacity level up to 3,840. For throughput capacity levels above 3,840, the maximum write throughput is 3,200 MBps.

The previous tables show your file system's throughput capacity for uncompressed data. However, because data compression reduces the amount of data that needs to be transferred as disk I/O, you can often deliver higher levels of throughput for compressed data. For example, if your data

is compressed to be 50% smaller (that is, a compression ratio of 2), then you can drive up to 2x the throughput than you could if the data were uncompressed. For more information, see Data compression.

Performance considerations

Here are a few important performance considerations when working with file systems using the Intelligent-Tiering storage class:

- Due to the higher latency and per-request costs of data access from Intelligent-Tiering storage, you should configure your workloads and file systems with I/O size in mind. Workloads reading data with smaller I/O sizes will require higher concurrency and incur more request costs to achieve the same throughput on data not in the cache as workloads using larger I/O sizes.
- For workloads with working-sets larger than the size of the SSD cache, consider maximizing application I/O size and FSx volume record size. Because writes to the file system are only stored on either the in-memory cache (for asynchronous writes) or the SSD write cache (for synchronous writes) before being served to the client, this should be less of a consideration for write I/O.
- You may be more likely to see higher latencies after a maintenance event. This is due to the in-memory cache being erased during maintenance windows for file systems using the Intelligent-Tiering storage class. For more information, see Modifying file system maintenance windows.
- The maximum disk IOPS your clients can drive with an Intelligent-Tiering file system depends on the specific access patterns of your workload and whether you have provisioned an SSD read cache. For workloads with random access, clients can typically drive much higher IOPS if the data is cached in the SSD read cache than if the data is not in the cache.
- Random read requests from Intelligent-Tiering storage will have higher latencies than sequential read requests, as the data will be served just-in-time instead of ahead of time through pre-fetch. We recommend configuring your data access pattern sequentially when possible to allow for pre-fetching data and higher performance.
- While you can select a throughput capacity as low as 160 megabytes per second (MBps), we recommend using higher throughput capacity levels for workloads that are write-heavy.

Managing Amazon FSx for OpenZFS file system resources

A *file system* is the primary FSx for OpenZFS resource. The sections below provide information on how to create, view, update, and delete FSx for OpenZFS file systems. These sections also include information on how to view file system status and configure your file system's weekly maintenance window.

Topics

- Creating an Amazon FSx for OpenZFS file system
- Viewing an Amazon FSx for OpenZFS file system
- Updating an Amazon FSx for OpenZFS file system
- Deleting an Amazon FSx for OpenZFS file system

Creating an Amazon FSx for OpenZFS file system

This section contains instructions on how to create a file system using the AWS CLI and the Amazon FSx API, as well as details on the file system properties that you can configure. For information on how to create a file system using the Amazon FSx console, see Step 1: Create a file system.

Topics

- Creating a file system (AWS CLI and Amazon FSx API)
- Configurable file system properties

Creating a file system (AWS CLI and Amazon FSx API)

To create an FSx for OpenZFS file system (CLI and API)

Use the create-file-system CLI command (or the equivalent CreateFileSystem API operation). The following example creates an FSx for OpenZFS file system with a `SINGLE_AZ_1` deployment type.

```
aws fsx create-file-system\
  --region us-east-1 \
   --file-system-type OPENZFS \
   --storage-capacity 10000 \
   --storage-type SSD \
   --security-group-ids sg-0123456789abcdef3,sg-0123abcd4567ef89a \
```

```
    --subnet-ids subnet-1234567890abcdef4 \
    --tags Key=creator,Value=allison \
    --open-zfs-configuration '{
       "AutomaticBackupRetentionDays": 30,
       "CopyTagsToBackups": true,
       "DailyAutomaticBackupStartTime": "02:00",
       "DeploymentType": "SINGLE_AZ_1",
       "DiskIopsConfiguration": {
          "Iops": 250,
          "Mode": "USER_PROVISIONED"
       },
       "RootVolumeConfiguration": {
          "CopyTagsToSnapshots": true,
          "DataCompressionType": "LZ4",
          "NfsExports": [
             {
                "ClientConfigurations": [
                   {
                      "Clients": "*",
                      "Options": [ "rw","root_squash","crossmnt" ]
                   }
                ]
             }
          ],
          "ReadOnly": false,
          "RecordSizeKiB": 128,
          "UserAndGroupQuotas": [
             {
                "Id": 1001,
                "StorageCapacityQuotaGiB": 2000,
                "Type": "GROUP"
             }
          ]
       },
       "ThroughputCapacity": 128
    }'
```

After successfully creating the file system, Amazon FSx returns the file system's description in JSON format.

Configurable file system properties

When you create a file system, you specify the following file system properties:

- **Deployment type** – The deployment type of your file system—Multi-AZ (HA), Single-AZ (HA), or Single-AZ (non-HA). Multi-AZ (HA) file systems provide additional resiliency by replicating your data and provide high availability by automatically failing over between multiple Availability Zones within the same AWS Region. Single-AZ (HA) file systems deploy primary and standby file servers within the same Availability Zone to ensure continuous availability during failover and failback. Single-AZ (non-HA) file systems replicate your data and provide automatic self-healing within a single Availability Zone. Both Single-AZ (HA) and Single-AZ (non-HA) offer Single-AZ 1 and Single-AZ 2. For more information, see Availability and durability.
- **Storage class** – The storage class of your file system. Choose either Intelligent-Tiering (elastic) or SSD (provisioned). We recommend Intelligent-Tiering for elastic storage that is suitable for most workloads. Intelligent-Tiering also comes with an optional SSD read cache for frequently accessed data. SSD (provisioned) is best for workloads that are latency sensitve and not cache-friendly. If you select SSD (provisioned), you will also need to specify an SSD storage capacity for your file system, from 64 to 524,288 GiB.
- **Provisioned SSD IOPS** – The maximum number of read and write operations for your file system. You can use the default setting of 3 IOPS per GB of SSD storage, or you can provision the SSD IOPS to a maximum of 160,000 SSD IOPS per file system for Single-AZ 1 and 400,000 SSD IOPS per file system for Single-AZ 2 and Multi-AZ*. You pay for additional SSD IOPS that you provision above the default 3 IOPS per GB of SSD storage.

  > **Note**
  >
  > *The maximum SSD IOPS you can provision for Multi-AZ file systems depends on the AWS Region your file system is located in. For more information, see Data access from disk.

- **Throughput capacity** – The sustained speed at which the file server that hosts your file system can serve data, in megabytes per second (MBps). You can use the default Amazon FSx-provisioned value or you can specify a different value. You pay for additional throughput capacity that you provision above the Amazon FSx default value.

  You can increase the amount of throughput capacity as needed at any time after you create the file system. For more information, see Modifying throughput capacity.
- **Network and security** – The VPC and subnets for the management and data access endpoints that your file system creates. For Multi-AZ file systems, you also define an IP address range and route tables. The maximum number of route tables that you can specify is 15.

- **Encryption** – Amazon FSx automatically encrypts the data in your file system at rest using the Amazon FSx service AWS Key Management Service key for your AWS account by default. You can choose to use a different KMS key.

Viewing an Amazon FSx for OpenZFS file system

This section contains instructions on how to view the details of your FSx for OpenZFS file system using the Amazon FSx console, the AWS CLI, and the Amazon FSx API, as well as information on the different file system statuses.

Topics

- Viewing file system details (Amazon FSx console, AWS CLI, and Amazon FSx API)
- File system status

Viewing file system details (Amazon FSx console, AWS CLI, and Amazon FSx API)

To view a file system's details (Amazon FSx console, CLI, and API):

- **Using the console** – Choose a file system to view the **File systems** detail page. The **Summary** panel shows the file system ID, lifecycle status, deployment type, Availability Zone, storage type, storage capacity, throughput capacity, provisioned IOPS, and creation time. The tabs provide detailed information and configuration functions for the file system's features, such as backups and volumes.
- **Using the CLI or API** – Use the describe-file-systems CLI command or the DescribeFileSystems API operation.

File system status

Depending on what actions you have taken on it, your file system will have one of the following statuses. You can find the status of an Amazon FSx file system by viewing the file system details using the instructions in the previous section.

File system status	Description
AVAILABLE	The file system has been successfully created and is available for use.
CREATING	Amazon FSx is creating a new file system.
DELETING	Amazon FSx is deleting an existing file system.
MISCONFIGURED	The file system is in a misconfigured but recoverable state.
FAILED	1. The file system has failed and Amazon FSx can't recover it. 2. When creating new file system, Amazon FSx was unable to create a new file system.

Updating an Amazon FSx for OpenZFS file system

This section provides information on how to update modifyable properties, including storage capacity, throughput capacity, and maintenance windows, on your file system using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

Topics

- Updating a file system (Amazon FSx console, AWS CLI, and Amazon FSx API)
- Modifiable file system properties
- Modifying provisioned SSD storage capacity and IOPS
- Modifying provisioned SSD read cache
- Modifying throughput capacity
- Modifying file system maintenance windows

Updating a file system (Amazon FSx console, AWS CLI, and Amazon FSx API)

Amazon FSx Console

To update how file system tags are copied

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **File systems**, and then choose the FSx for OpenZFS file system that you want to update.
3. For **Actions**, choose **Update file system**. The **Update file system** dialog box displays.
   - For **Copy tags to backups**, choose whether to copy tags from the file system to any backup that's taken.
   - For **Copy tags to volumes**, choose whether to copy tags from the file system to any volume that you create.
4. Choose **Update** to update the file system with your changes.

To update automatic daily backups

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. To display the file system details page, in the left navigation pane, choose **File systems**, and then choose the FSx for OpenZFS file system that you want to update.
3. Choose the **Backups** tab, and then choose Update.
4. Modify the automatic daily backup settings for this file system, and then choose **Save**.

To update the file system's VPC route tables

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. To display the file system details page, in the left navigation pane, choose **File systems**, and then choose the FSx for OpenZFS file system that you want to update.
3. For **Actions**, choose **Update route tables**. This option is only available for Multi-AZ file systems.
4. In the **Manage route tables** dialog box. do one of the following:

- To associate a new VPC route table, select a route table from the **Associate new route tables** dropdown list, and then choose **Associate**.
- To disassociate an existing VPC route table, select a route table from the **Current route tables** pane, and then choose **Disassociate**.

5. Choose **Close**.

AWS CLI and Amazon FSx API

To update a file system (CLI and Amazon FSx API)

- To update the configuration of an FSx for OpenZFS file system, 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 \
    --open-zfs-configuration
 AutomaticBackupRetentionDays=30,DailyAutomaticBackupStartTime=01:00, \
      WeeklyMaintenanceStartTime=1:01:30,AddRouteTableIds=rtb-0123abcd
```

Modifiable file system properties

This section provides information on how to update the following FSx for OpenZFS file system properites. You can update an FSx for OpenZFS file system's configuration using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

- **Automatic daily backups** – Back up your file system automatically on a daily basis. Modify the backup window and the backup retention period. For more information about backups, see Working with automatic daily backups.
- **Copy tags to backups** – Copy file system tags to file system backups.
- **Copy tags to volumes** – Copy file system tags to the volumes that are attached to the file system.
- **Provisioned SSD IOPS** – Set a fixed number of IOPS or have Amazon FSx automatically maintain 3 SSD IOPS per GiB of storage capacity. For information on how to increase SSD IOPS, see Updating SSD storage capacity and provisioned IOPS.

- **Storage capacity** – For file systems with SSD (provisioned) storage, storage capacity cannot be decreased after creation, only increased. For information on how to increase SSD (provisioned) storage capacity, see Updating SSD storage capacity and provisioned IOPS. For file systems using the Intelligent-Tiering storage class, storage capacity will automatically grow and strink to fit your data needs as they change.
- **SSD read cache** – For file systems using the Intelligent-Tiering storage class, you can modify the sizing mode of the optional provisioned SSD read cache at any time. For information on how to modify your SSD read cache, see Modifying provisioned SSD read cache.
- **Throughput capacity** – You can increase or decrease your file system's throughput capacity at any point. For information on how to update throughput capacity, see Modifying throughput capacity.
- **Weekly maintenance window** – Set the day of the week and time that Amazon FSx performs file system maintenance and updates. For information on how to change the weekly maintenance window, see Changing the weekly maintenance window.
- **Amazon VPC route tables** – For Multi-AZ file systems, FSx for OpenZFS creates an endpoint for accessing your file system in a VPC route table. You can associate new route tables that you create with your existing Multi-AZ file systems—allowing you to configure which clients can access your data even as your network evolves. You can also disassociate (remove) existing route tables from your file system.

Modifying provisioned SSD storage capacity and IOPS

When you need additional storage for your dataset, you can increase the solid state drive (SSD) storage capacity of your Amazon FSx for OpenZFS file system without any disruption to your end users or applications by using the Amazon FSx console, Amazon FSx API, or AWS Command Line Interface (AWS CLI).

You can also change the provisioned SSD IOPS for your file system when you increase SSD storage capacity, or as an independent action. To specify the amount of provisioned SSD IOPS for your file system, use one of two IOPS modes:

- Use **Automatic** mode if you want Amazon FSx to automatically scale your SSD IOPS.
- Use **User-provisioned** mode if you want to provision a specific amount of SSD IOPS.

For more information about these modes, see Considerations when updating storage and IOPS.

When you increase the SSD storage capacity of your Amazon FSx file system, the new capacity is available for use within minutes. You can update the SSD storage capacity or SSD IOPS at anytime, as long as storage capacity increases are at least 6 hours apart. These updates do not impact the availability of your file system in any way. You will be billed for the new SSD storage capacity after it becomes available to you. For more information, see Amazon FSx for OpenZFS Pricing.

You can track the progress of an SSD storage capacity increase or SSD IOPS update at any time by using the Amazon FSx console, CLI, and API. For more information, see Monitoring storage capacity and IOPS updates.

Once the increased SSD capacity is available, if the file system's root volume storage capacity quota is set to the same size as the file system, FSx will automatically update the volume's storage capacity quota to match the newly-increased file system capacity. Otherwise, you need to manually increase the `storage capacity quota` of the root volume, and any other volumes in your file system. For more information, see Updating an Amazon FSx for OpenZFS volume.

Topics

- Considerations when updating storage and IOPS
- When to increase storage capacity
- Updating SSD storage capacity and provisioned IOPS
- Monitoring storage capacity and IOPS updates

Considerations when updating storage and IOPS

Here are a few important considerations when modifying your SSD storage capacity and provisioned IOPS:

- **Storage capacity increase only** – You can only *increase* the amount of SSD storage capacity for a file system; you cannot decrease the storage capacity.
- **Storage capacity minimum increase** – Each SSD storage capacity increase must be a minimum of 10 percent of the file system's current SSD storage capacity, up to the maximum allowed value of 512 Tebibytes (TiB)*.

> **Note**
>
> *The maximum storage capacity of your file system depends on the AWS Region in which it is located. For more information, see Resource quotas for each file system.

- **Time between increases** – You can't make further SSD storage capacity increases on a file system until 6 hours after the last increase was requested.
- **Allocating increased storage capacity** – If the file system's root volume storage capacity quota is set to the same size as the file system, FSx will automatically update the volume's storage capacity quota to match the newly-increased file system capacity. Otherwise, you will need to manually increase storage on the root volume and any other volumes in your file system.
- **Provisioned IOPS modes** – For a provisioned IOPS change, you must specify a mode. The two IOPS modes are the following:
  - **Automatic** mode – Amazon FSx automatically scales your SSD IOPS to maintain 3 SSD IOPS per GiB of storage capacity, up to the maximum number of IOPS for your file system.
  - **User-provisioned** mode – You specify the number of SSD IOPS, which must be greater than or equal to 3 IOPS per GiB of storage capacity. If the amount of SSD IOPS is not at least 3 IOPS per GiB, the request will fail. You can optionally provision a higher level of IOPS. If you do so, you pay for the average IOPS provisioned above 3 IOPS per GiB per file system.

    Note

    For file systems that are already configured with **User-provisioned** SSD IOPS, you must specify a value for **User-provisioned** SSD IOPS when you are updating your file system, or the update request will fail.

When to increase storage capacity

If you are running out of SSD storage, we recommend that you increase the storage capacity of your file system. You can monitor SSD storage capacity on the file system using these file system-level Amazon CloudWatch metrics.

- `StorageCapacity` measure the total amount of file system SSD storage capacity.
- `UsedStorageCapacity` measures the amount of used SSD storage capacity.

You can use these metrics to measure storage capacity and create alarms. The following are some examples:

- Per cent storage capacity used = `UsedStorageCapacity` ÷ `StorageCapacity`
- Per cent storage capacity available = `StorageCapacity` ÷ `UsedStorageCapacity`

- Amount of free storage capacity = `StorageCapacity` – `UsedStorageCapacity`

You can create a CloudWatch alarm on a metric and get notified when it drops below a specific threshold. For more information, see Monitoring with Amazon CloudWatch.

Updating SSD storage capacity and provisioned IOPS

You can increase a file system's SSD storage capacity and modify your provisioned SSD IOPS by using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

To update SSD storage capacity and provisioned IOPS for a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for OpenZFS file system that you want to update SSD storage capacity and SSD IOPS for.
3. On the **Summary** panel, choose **Update** next to the file system's **SSD storage capacity** value.

   The **Update SSD storage capacity and IOPS** dialog box appears.
4. To increase SSD storage capacity, select **Modify storage capacity**.
5. For **Input type**, choose one of the following:

   - To enter the new SSD storage capacity as a percentage change from the current value, choose **Percentage**.

     For **Desired % increase**, enter the percentage by which you want to increase storage capacity. This value must be at least 10 percent.
   - To enter the new value in GiB, choose **Absolute**.

     For **Desired storage capacity**, enter the new value for SSD storage capacity value in GiB, up to the maximum allowed value of 512 TiB*.

     > **Note**
     >
     > *The maximum storage capacity of your file system depends on the AWS Region in which it is located. For more information, see Resource quotas for each file system.
6. For **Provisioned SSD IOPS**, you have two options to modify the number of provisioned SSD IOPS for your file system:

- If you want Amazon FSx to automatically scale your SSD IOPS to maintain 3 provisioned SSD IOPS per GiB of primary storage capacity, up to a maximum of 160,000 for Single-AZ 1 (non-HA and HA) and 400,000 for Single-AZ 2 (non-HA) and Multi-AZ (HA)*, choose **Automatic**.
- If you want to specify the number of SSD IOPS, choose **User-provisioned**. Enter an absolute number of IOPS that is at least 3 times the amount of GiB of your primary storage tier, and less than or equal to the maximum number of IOPS for your file system.

7. Choose **Update**.

To update SSD storage capacity and provisioned IOPS for a file system (CLI)

To update the SSD storage capacity and provisioned IOPS for an FSx for OpenZFS file system, use the AWS CLI command update-file-system (UpdateFileSystem is the equivalent API action). Set the following parameters:

- Set `--file-system-id` to the ID of the file system that you are updating.
- To increase your SSD primary storage capacity, set `--storage-capacity` to a value that is at least 10 percent greater than the current value.
- To modify your provisioned SSD IOPS, use the `--open-zfs-configuration DiskIopsConfiguration` property. This property has two parameters, `Iops` and `Mode`:
  - If you want to specify the number of provisioned SSD IOPS, use `Iops=`*`number_of_IOPS`*, up to a maximum of 160,000 for Single-AZ 1 (non-HA and HA) and 400,000 for Single-AZ 2 (non-HA and HA) and Multi-AZ (HA)*, and `Mode=USER_PROVISIONED`. The SSD IOPS value must be greater than or equal to 3 times the requested SSD storage capacity. If you're not increasing the storage capacity, the IOPs value must be greater than or equal to 3 times the current SSD storage capacity.

    Note

    *The maximum SSD IOPS you can provision for Multi-AZ file systems depends on the AWS Region your file system is located in. For more information, see Data access from disk.

  - If you want Amazon FSx to automatically increase your SSD IOPS, use `Mode=AUTOMATIC` and don't use the `Iops` parameter. Amazon FSx will automatically maintain 3 provisioned SSD IOPS per GiB of your primary storage capacity, up to a maximum of 160,000 for Single-AZ 1 (non-HA and HA) and 400,000 for Single-AZ 2 (non-HA and HA) and Multi-AZ (HA).

The following example requests an increase of 2000 GiB to the file system's SSD storage capacity. It also requests 7000 provisioned SSD IOPS.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --storage-capacity 2000 \
    --open-zfs-configuration 'DiskIopsConfiguration={Iops=7000,Mode=USER_PROVISIONED}'
```

To monitor the progress of the update, use the describe-file-systems AWS CLI command. Look for the `AdministrativeActions` section in the output.

For more information, see AdministrativeAction in the *Amazon FSx for OpenZFS API Reference*.

Monitoring storage capacity and IOPS updates

You can monitor the progress of an SSD storage capacity and IOPS update by using the Amazon FSx console, the API, or the AWS CLI.

Monitoring updates in the console

You can monitor file system updates in the **Updates** tab on the **File system details** page.

For SSD storage capacity and IOPS updates, you can view the following information:

Update type

Supported types are **Storage capacity**, **IOPS Mode**, and **SSD IOPS**. The **IOPS Mode** and **SSD IOPS** values are listed for all storage capacity and IOPS scaling requests.

Target value

The updated value for the file system's SSD storage capacity or IOPs.

Status

The current status of the update. 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 update finished successfully.
- **Failed** – The update request failed. Choose the question mark (**?**) to see details on why the request failed.

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 SSD storage capacity increase requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API operation. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you increase a file system's SSD storage capacity, a `FILE_SYSTEM_UPDATE` `AdministrativeActions` is generated.

The following example shows an excerpt of the response of a `describe-file-systems` CLI command. The file system has a pending administrative action to increase the SSD storage capacity to 2000 GiB and the provisioned SSD IOPS to 7000.

```
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1586797629.095,
        "Status": "PENDING",
        "TargetFileSystemValues": {
            "StorageCapacity": 2000,
            "OpenZFSConfiguration": {
                "DiskIopsConfiguration": {
                    "Mode": "USER_PROVISIONED",
                    "Iops": 7000
                }
            }
        }
    }
]
```

Amazon FSx processes the `FILE_SYSTEM_UPDATE` action, increasing the file system's storage capacity. When the new storage is available to the file system, the `FILE_SYSTEM_UPDATE` status changes to `COMPLETED`. The storage capacity shows the new larger value. This behavior is shown in the following excerpt of the response of a `describe-file-systems` CLI command.

```
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
```

```
        "RequestTime": 1586799169.445,
        "Status": "UPDATED_OPTIMIZING",
        "TargetFileSystemValues": {
            "StorageCapacity": 2000,
            "OpenZFSConfiguration": {
                "DiskIopsConfiguration": {
                    "Mode": "USER_PROVISIONED",
                    "Iops": 7000
                }
            }
        }
    }
]
```

If the storage capacity or IOPS update request fails, the status of the `FILE_SYSTEM_UPDATE` action changes to `FAILED`, as shown in the following example. The `FailureDetails` property provides information about the failure.

```
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1586373915.697,
        "Status": "FAILED",
        "TargetFileSystemValues": {
            "StorageCapacity": 2000,
            "OpenZFSConfiguration": {
                "DiskIopsConfiguration": {
                    "Mode": "USER_PROVISIONED",
                    "Iops": 7000
                }
            }
        },
        "FailureDetails": {
            "Message": "failure-message"
        }
    }
]
```

Modifying provisioned SSD read cache

When you create a file system with the Intelligent-Tiering storage class, you have the option to also provision an SSD read cache that provides SSD latencies for reads of frequently accessed data, up

to 3 IOPS per GiB. The SSD read cache has three sizing modes: Automatic, Custom, and None. After your file system is created, you can modify your SSD read cache's sizing mode at any time.

Topics

- Considerations when updating SSD read cache sizing mode
- Updating the provisioned SSD read cache
- Monitoring SSD read cache updates

Considerations when updating SSD read cache sizing mode

Here are a few important considerations when modifying your SSD read cache:

- Any time you modify the SSD read cache, all of its contents will be erased. This means that you may see a decrease in performance levels until the SSD read cache is populated again.
- You can both increase and decrease the size of the SSD read cache. However, you can only do this once every six hours. There is no time restriction when adding or removing the SSD read cache from your file system.
- You must increase or decrease the size of your SSD read cache by a minimum of 10% every time you modify it.

Updating the provisioned SSD read cache

You can update your SSD read cache using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

To update the SSD read cache for an Intelligent-Tiering file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for OpenZFS file system that you want to update the SSD read cache for.
3. On the **Summary** panel, choose **Update** next to the file system's **SSD read cache configuration** value.

   The **Update SSD read cache** panel appears.
4. Select the new sizing mode that you would like for your SSD read cache, as follows:

- Choose **Automatic (Proportional to throughput capacity)** to have your SSD read cache automatically sized based on your throughput capacity.
- Choose **Custom (User-provisioned)** if you know the approximate size of your dataset and would like to customize your SSD read cache. If you select Custom, you will also need to specify the **Desired read cache capacity** in GiB.
- Choose **None** if you do not want to use an SSD read cache with your Intelligent-Tiering file system.

5. Choose **Update**.

To update the SSD read cache for an Intelligent-Tiering file system (CLI & API)

To update the SSD read cache for an Intelligent-Tiering file system, use the AWS CLI command update-file-system or the equivalent UpdateFileSystem API action. Set the following parameters:

- Set `--file-system-id` to the ID of the file system that you are updating.
- To modify your SSD read cache, use the `--open-zfs-configuration ReadCacheConfiguration` property. This property has two parameters, `SizeGiB` and `SizingMode`:
  - **SizeGiB** - Sets the size of your SSD read cache in GiB when using `USER_PROVISIONED` mode.
  - **SizingMode** - Sets the sizing mode of your SSD read cache.
    - Set to `NO_CACHE` if you do not want to use an SSD read cache with your Intelligent-Tiering file system.
    - Set to `USER_PROVISIONED` to specify the exact size of your SSD read cache.
    - Set to `PROPORTIONAL_TO_THROUGHPUT_CAPACITY` to have your SSD read cache automatically sized based on your throughput capacity.

The following example updates the SSD read cache sizing mode to **USER_PROVISIONED** and sets the size to 524288 GiB.

```
aws fsx update-file-system \
    --open-zfs-configuration
 'ReadCacheConfiguration={SizeGiB=524288,SizingMode=USER_PROVISIONED}'
```

To monitor the progress of the update, use the describe-file-systems AWS CLI command. Look for the `AdministrativeActions` section in the output.

For more information, see AdministrativeAction in the *Amazon FSx for OpenZFS API Reference*.

Monitoring SSD read cache updates

You can monitor the progress of an SSD read cache update by using the Amazon FSx console, the API, or the AWS CLI.

Monitoring updates in the console

You can monitor file system updates in the **Updates** tab on the **File system details** page.

For SSD read cache updates, you can view the following information:

Update type

Supported types are **SSD read cache sizing mode** and **SSD read cache size**.

Target value

The updated value for the file system's SSD read cache sizing mode or SSD read cache size.

Status

The current status of the update. 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 update finished successfully.
- **Failed** – The update request failed. Choose the question mark (**?**) to see details on why the request failed.

Request time

The time that Amazon FSx received the update action request.

Monitoring SSD read cache updates with the AWS CLI and API

You can view and monitor file system SSD read cache update requests using the describe-file-systems AWS CLI command and the DescribeFileSystems API operation. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you update a file system's SSD read cache, a `FILE_SYSTEM_UPDATE` `AdministrativeActions` is generated.

The following example shows an excerpt of the response of a `describe-file-systems` CLI command. The file system has a pending administrative action to change the SSD read cache sizing mode to `USER_PROVISIONED` and the SSD read cache size to 524288.

```
"AdministrativeActions": [
    {
        "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
        "RequestTime": 1586797629.095,
        "Status": "PENDING",
        "TargetFileSystemValues": {
            "OpenZFSConfiguration": {
                "ReadCacheConfiguration": {
                     "SizingMode": "USER_PROVISIONED"
                     "SizeGiB": 524288,
                }
            }
        }
    }
]
```

When the new SSD read cache configuration is available to the file system, the `FILE_SYSTEM_UPDATE` status changes to `COMPLETED`. If the SSD read cache update request fails, the status of the `FILE_SYSTEM_UPDATE` action changes to `FAILED`.

Modifying throughput capacity

Every FSx for OpenZFS file system has a throughput capacity that is configured when you create the file system. You can modify your file system's throughput capacity at any time, as needed. Throughput capacity is one factor that determines the speed at which the file server hosting the file system can serve file data. Higher levels of throughput capacity also come with higher levels of I/O operations per second (IOPS) and more memory for caching of data on the file server. For more information, see Performance for Amazon FSx for OpenZFS.

When you modify your file system's throughput capacity, behind the scenes, Amazon FSx switches out the file system's file server. For Single-AZ (HA) and Multi-AZ (HA) file systems, it results in an automatic failover and failback while Amazon FSx switches out the preferred and secondary file servers. Single-AZ (non-HA) file systems will be unavailable for a few minutes during throughput capacity scaling. You are billed for the new amount of throughput capacity once it is available to your file system.

File systems offer varying levels of throughput capacity, depending on your deployment type. The values of throughput capacity (in MBps) for all Single-AZ deployment types and Multi-AZ (HA) are as follows:

- **Single-AZ 2 (non-HA and HA) and Multi-AZ (HA)** – 160, 320, 640, 1280, 2560, 3840, 5120, 7680, 10240
- **Single-AZ 1(HA)** – 128, 256, 512, 1024, 2048, 3072, 4096
- **Single-AZ 1(non-HA)** – 64, 128, 256, 512, 1024, 2048, 3072, 4096

Topics

- When to modify throughput capacity
- Modifying throughput capacity

When to modify throughput capacity

Amazon FSx integrates with Amazon CloudWatch, enabling you to monitor your file system's ongoing throughput usage levels. The performance (throughput and IOPS) that you can drive through your file system depends on your specific workload's characteristics, in addition to your file system's throughput capacity, storage capacity, and storage type. You can use CloudWatch metrics to determine which of these dimensions to change to improve performance. For more information, see Monitoring with Amazon CloudWatch.

Modifying throughput capacity

You can modify a file system's throughput capacity using the Amazon FSx console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx API.

To modify a file system's throughput capacity (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the FSx for OpenZFS file system that you want to increase the throughput capacity for.
3. For **Actions**, choose **Update throughput capacity**. 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 **Desired throughput capacity** from the list.
5. Choose **Update** to initiate the throughput capacity update.

> **Note**
>
> Your file system may experience a very brief period of unavailability during the update.

To modify a file system's throughput capacity (CLI)

- To modify a file system's throughput capacity, 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 \
    --open-zfs-configuration '{"ThroughputCapacity":512}'
```

Modifying file system maintenance windows

Amazon FSx for OpenZFS performs routine software patching for the OpenZFS software it manages. The maintenance window is your opportunity to control what day and time of the week this software patching occurs. You can choose the maintenance window during file system creation, or at a later time. If you have no time preference, a 30-minute default window is assigned.

FSx for OpenZFS allows you to adjust your maintenance window as needed to accommodate your workload and operational requirements. You can move your maintenance window as frequently as required, provided that a maintenance window 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 OpenZFS will proceed with maintenance on the file system to ensure its security and reliability.

Patching occurs infrequently, typically once every month during the maintenance window that you specify. Patching should require only a portion of your 30-minute maintenance window. We recommend scheduling your maintenance window during idle periods when there is minimal load on your file system. While patching is in progress, you should expect that your Single-AZ (non-HA) file systems will be unavailable, typically for less than 15 minutes. Your Single-AZ (HA) and Multi-AZ (HA) file systems will remain available and automatically fail over and fail back between the preferred file server and the standby file server. For more information, see Failover process for FSx for OpenZFS.

For file systems that use the Intelligent-Tiering storage class, the in-memory cache will be erased during maintenance, leading to higher latencies, particularly on metadata, until after maintenance has been completed.

> **Note**
>
> To ensure data integrity during maintenance activity, FSx for OpenZFS completes pending write operations to the underlying storage volumes that host your file system before maintenance begins.

You can use the Amazon FSx Management Console, AWS CLI, AWS API, or one of the AWS SDKs to change the maintenance window for your file systems.

Changing the weekly maintenance window

To change the weekly maintenance window (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose **File systems** in the left hand navigation column.
3. Choose the file system that you want to change the weekly maintenance window for. The file system details page displays.
4. Choose **Administration** to display the file system administration **Settings** panel.
5. Choose **Update** to display the **Change maintenance window** window.
6. Enter the new day and time that you want the weekly maintenance window to start.
7. Choose **Save** to save your changes. The new maintenance start time is displayed in the file system administration **Settings** panel.

To change the weekly maintenance window using the update-file-system CLI command, see Updating a file system (Amazon FSx console, AWS CLI, and Amazon FSx API).

Deleting an Amazon FSx for OpenZFS file system

This section contains instructions on how to delete a file system using the AWS CLI and the Amazon FSx API. For information on how to delete a file system using the Amazon FSx console, see Step 3: Clean up your resources.

Topics

- Deleting a file system

Deleting a file system

To delete a file system (AWS CLI and Amazon FSx API)

- Use the delete-file-system CLI command or the DeleteFileSystem API operation. The following is an example using the CLI to delete

> **Note**
>
> To delete a file system which still has child volumes present, you must include `DELETE_CHILD_VOLUMES_AND_SNAPSHOTS` in the `Options` property, otherwise the delete request will fail.

```
aws fsx delete-file-system \
    --file-system-id fs-1234567890abcdef0
    --open-zfs-configuration '{
      "FinalBackupTags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Options": [ "DELETE_CHILD_VOLUMES_AND_SNAPSHOTS" ],
      "SkipFinalBackup": boolean
   }'
```

Managing Amazon FSx for OpenZFS volumes

An FSx for OpenZFS file system can contain one or more *volumes*, which are isolated data containers for files and directories. Every FSx for OpenZFS file system has one (and only one) *root volume*, which is created at file system creation time. All other volumes created on a file system are children of the root volume.

After the file system is created, you can create volumes as needed. Each customer-created volume is a child of another volume. This means that all volumes are children or descendants of the root volume. For example, you could create 10 volumes with each one being a child of the root volume (that is, all 10 volumes are siblings) or you could create a hierarchy of 10 volumes in which each volume is a child of the previous volume.

You use volumes to logically separate an individual file system into multiple namespaces. This allows you to independently manage the storage capacity, compression, NFS exports, record size, and user and group quotas at the volume level.

You access volumes from Linux, Windows, or macOS clients over the Network File System (NFS) protocol (v3, v4.0, v4.1, or v4.2). FSx for OpenZFS presents data to your users and applications as a local directory or drive.

Topics

- Creating an Amazon FSx for OpenZFS volume
- Viewing an Amazon FSx for OpenZFS volume
- Updating an Amazon FSx for OpenZFS volume
- Deleting an Amazon FSx for OpenZFS volume

Creating an Amazon FSx for OpenZFS volume

This section provides information on the volume properties that you can configure, as well as instructions on how to create a volume using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

Topics

- Creating a volume

- Configurable volume properties

Creating a volume

You can create new volumes, or create a volume from an existing volume snapshot using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

To create a volume (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **File systems**, and then choose the FSx for OpenZFS file system that you want to create a volume for.
3. Choose the **Volumes** tab.
4. Choose **Create volume**.

   The **Create volume** dialog box appears.
5. In the **File system** field, choose the file system to create the volume on.
6. In the **Parent volume ID** field, choose the ID of the parent volume, which can be the root volume or another volume.
7. In the **Volume name** field, provide a name for the volume. You can use a maximum of 203 alphanumeric characters, plus the underscore (_) special character. The name must be unique among all the volume names on the same parent volume on the same file system.
8. For **Storage capacity quota - optional**, you can set a quota that will be the maximum storage size for the volume. This quota cannot be larger than the parent volume quota. To set no quota and allow this volume to consume any available capacity in your file system, set this property to `-1`.
9. For **Storage capacity reservation - optional**, you can enter the reservation for the volume. This reserves dedicated space on the file system storage pool, meaning the space available to all other volumes is reduced by the amount specified. It cannot be set to a value that's greater than the parent volume quota or the remaining reservation space on the file system storage. To set no reservation and allow this volume to consume any available capacity in your file system, set this property to `0` or `-1`.
10. For **Data compression type**, choose the type of compression to use for your volume, either **Zstandard**, **LZ4**, or **No compression**. Zstandard compression provides more data compression and higher read throughput than LZ4 compression. LZ4 compression provides

less compression and higher write throughput performance than Zstandard compression. For more information about the storage and performance benefits of the volume data compression options, see Data compression.

11. For **Copy tags to snapshots**, enable or disable the option to copy tags on the root volume to snapshots.
12. For **NFS exports**, there is a default client configuration setting which you can modify or remove. Client configurations define which clients can access the volume and their permissions.

    To provide additional client configurations:

    a. In the **Client addresses** field, specify which clients can access the volume. Enter an asterisk (*) for any client, a specific IP address, or a CIDR range of IP addresses.

    b. In the **NFS options** field, enter a comma-delimited set of exports options. For example, enter `rw` to allow read and write permissions to the volume.

    c. Choose **Add client configuration**.

    d. Repeat the procedure to add another client configuration.

    For more information, see NFS exports.
13. For **Record size**, choose whether to use the default suggested record size of 128 KiB, or to set a **User-configured** suggested record size for the volume. Generally, workloads that write in fixed small or large record sizes may benefit from setting a custom record size, like database workloads (small record size) or media streaming workloads (large record size). We recommend using the default setting for the majority of use cases. For more information about the record size setting, see Configurable volume properties.
14. For **User and group quotas**, you can set a storage quota for a user or group:

    a. For **Quota type**, choose USER or GROUP.

    b. For **User or group ID**, choose a number that is the ID of the user or group.

    c. For **Usage quota**, choose a number that is the storage quota of the user or group.

    d. Choose **Add quota**.

    e. Repeat the procedure to add a quota for another user or group.
15. To create a volume from an existing volume snapshot, use **Source snapshot ID - optional**, to specify the ID of a snapshot from which to create a volume. Then choose a **Source snapshot copy strategy** option for the type of volume you're creating:

- **Clone** creates a clone volume. The snapshot will provide the seed content for the volume.
- **Full copy** creates a volume that will contain data copied from the snapshot.

16. Choose **Confirm** to create the volume.

You can monitor the progress on the **File systems** detail page, in the **Status** column of the **Volumes** pane. The volume is ready for use when its status is **Created**.

To create a volume (CLI)

- To create an FSx for OpenZFS volume, use the create-volume CLI command (or the equivalent CreateVolume API operation). The following example creates a new volume by cloning an existing snapshot.

```
aws fsx create-volume \
 --name vol2 \
    --volume-type OPENZFS \
    --tags Key=creator,Value=Liu \
    --open-zfs-configuration '{
      "CopyTagsToSnapshots": true,
      "DataCompressionType": "LZ4",
      "NfsExports": [
         {
            "ClientConfigurations": [
               {
                  "Clients": "*",
                  "Options": [ "rw","root_squash","crossmnt" ]
               }
            ]
         }
      ],
      "OriginSnapshot": {
         "CopyStrategy": "CLONE",
         "SnapshotARN": "arn:aws:fsx:us-east-2:111122223333:snapshot/
fsvol-0123456789abcdef0/fsvolsnap-1234567890abcdef0"
      },
      "ParentVolumeId": "fsvol-abcdef01234567890",
      "ReadOnly": false,
      "RecordSizeKiB": 128,
      "StorageCapacityQuotaGiB": 10000,
      "StorageCapacityReservationGiB": -1,
      "UserAndGroupQuotas": [
```

```
        {
           "Id": 1004,
           "StorageCapacityQuotaGiB": 2000,
           "Type": "GROUP"
        }
    ]
}'
```

After successfully creating the volume, Amazon FSx returns its description in JSON format.

Configurable volume properties

When you create a volume, you can configure the following properties to customize and control the storage aspects of the volume.

- **Volume name** provides a name for the volume. Please ensure that the specified name does not conflict with an existing file or directory on the parent volume.
- **Data compression type** reduces the storage capacity that your data consumes and can also help increase your effective throughput. You can choose either **Zstandard**, **LZ4**, or **No compression**. Zstandard compression provides a higher level of data compression and higher read throughput performance than LZ4 compression. LZ4 compression provides a lower level of compression and higher write throughput performance than Zstandard compression. For more information about the storage and performance benefits of the volume data compression options, see Data compression.
- **Storage capacity quota** sets a volume quota, which is the maximum storage size for the volume. A volume quota limits the amount of storage space that the volume can consume to the configured amount, but does not guarantee the space will be available. To guarantee quota space, you must also set **Storage capacity reservation**.

  By setting quotas without setting a **Storage capacity reservation**, you can create space-efficient *thin-provisioned* volumes where capacity is allocated only as storage is being consumed. With thin-provisioned volumes, you can assign quotas that are collectively larger than the existing capacity of the file system or quota of a parent volume. If your file system is nearing capacity or a parent volume is nearing its quota, note that your users or applications may not be able to write in a child volume even though the volume has not reached its quota limit.
- **Storage capacity reservation** guarantees a specified amount of storage space to always be available for the volume. The reservation reserves a configured amount of storage space from the parent volume. Only the volume with the reservation can use that storage space, regardless

of the volume quotas that other volumes may have. Note that unlike volume quotas, you can't reserve an amount of storage space that doesn't exist in its immediate parent. Set a reservation if an application must have a certain amount of storage space or it will fail.

- **Record size** sets the suggested block size for a volume in a ZFS dataset. Choose whether to use the default record size of 128 KiB, or to set a custom record size for the volume. We recommend using the default setting for the majority of use cases. For more information about the record size setting, see ZFS record size. Generally, workloads that write in fixed small or large record sizes may benefit from setting a custom record size, like database workloads (small record size) or media streaming workloads (large record size). See the OpenZFS documentation for more information about Dataset record size and ZFS datasets.
- **NFS exports** use NFS-level export policies to define how the file system should be exported over the NFS protocol. The NFS exports setting defines which clients can access the volume and what permissions they have. For more information, see NFS exports.
- **User and group quotas** configures individual user and/or group quotas for volumes, which sets a limit on the amount of storage space they can consume on the volume. To determine quota usage, add up the total size of files owned by the user or group specified in the quota. Only data in the volume on which the quota is applied counts toward the user or group's volume quota utilization. Other files and directories that exist only in snapshots or child volumes do not count toward quota usage.
- **Source snapshot ID** specifies a snapshot from which to create a volume. Then use **Source snapshot copy strategy** to specify the type of volume to create:
  - **Clone** creates a clone volume. A clone volume is a writable copy that is initialized with the same data as the snapshot from which it was created. Because clone volumes reference the data from the snapshot, clone volumes are created almost instantly, and initially consume no additional capacity. They only consume the capacity required for the incremental changes made to the source snapshot, providing an easy way to support multiple users or applications in parallel from a shared dataset. However, a clone volume maintains a dependency on its source snapshot, so you cannot delete this source snapshot while the clone volume is in use.
  - **Full copy** creates a full-copy volume. A full-copy volume is a writable copy that is initialized with the same data as the snapshot from which it was created. Unlike a clone volume, it does not maintain any dependency on its source snapshot. Because a full-copy volume requires copying all of the source snapshot data to a new volume, creation time will depend on the size of the source snapshot. While this data is being copied, your full-copy volume will be read only. Once a full-copy volume is created, it is identical to a standard FSx for OpenZFS volume. Files in the source snapshot will maintain their original record size regardless of the record size of

the destination volume. Files will be compressed according to the compression property on the destination volume.

For more information on snapshots, see Protecting your data with snapshots.

NFS exports

NFS exports are NFS-level export policies that configure which clients can access the volume and the options that are available. Each volume has its own NFS exports setting, so a client may be able to mount one volume on the file system but not a different volume.

When creating a volume from the console, you provide the NFS exports information in an array of client configurations, each of which has **Client addresses** and **NFS options** fields.

The **Client addresses** field specifies which hosts can mount over the NFS protocol and contains one of these settings:

- `*` is a wildcard that means anyone who can route to the file server can mount it.
- The IP address of a client's computer (such as `10.0.0.1`) that means a client from that specific IP address can mount the file system.
- A CIDR block range (such as `192.0.2.0/24`) that means any client from that address range can mount the file system.

> **Note**
>
> If an IP address is permitted to mount a parent volume, it is also automatically permitted to mount any of the child volumes.

The **NFS options** field lists a set of exports options available on the volume. Following are descriptions of the most common NFS options. For a more comprehensive list of exports options, see the exports(5) - Linux man page on the `die.net` web site.

- `rw` allows both read and write requests on this NFS volume from the specified Client addresses.
- `ro` allows only read requests on this NFS volume. The specified Client addresses can't write to the volume.

- `crossmnt` allows clients to inherit access to any child volumes within this volume (if configured along with the `no_subtree_check` option, which is included by default). This option is required to provide file-level access to your snapshots in the `.zfs/snapshot` directory of each volume.
- `all_squash` maps all User IDs (UIDs) and group IDs (GIDs) to the anonymous user.
- `root_squash` maps requests from UID/GID 0 to the anonymous UID/GID. It prevents remote root user from having superuser (root) privileges on remote NFS-mounted volumes. `root_squash` is the default unless overridden by `all_squash` or `no_root_squash`.
- `no_root_squash` turns off root squashing.
- `anonuid` and `anongid` explicitly set the UID and GID of the anonymous account. Valid values are 0 - 2147483647, inclusive.
- `sync` replies to client requests only after the changes have been committed to stable storage (that is, disk drives). `sync` is the default unless overridden by `async`
- `async` replies to client requests (such as write requests) after the changes have been committed to memory, but before any changes made by that request have been committed to stable storage (that is, disk drives). This setting can improve performance for latency-intensive or IOPS-intensive workloads. For more information, see Performance for Amazon FSx for OpenZFS.

> **Warning**
>
> Use of the `async` option can cause data to be lost or corrupted if a write request is acknowledged but the server crashes before the write request is fully written to disk.

When you create a volume, the default for **Client addresses** is an asterisk (*) and the default for **NFS options** is `rw,crossmnt`.

Viewing an Amazon FSx for OpenZFS volume

You can view the FSx for OpenZFS volumes that are currently on your file system using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

To view the volumes on your file system:

- **Using the console** – Choose a file system to view the **File systems** detail page. Choose the **Volumes** tab to list all the volumes on the file system, and then choose the volume you want to view.

- **Using the CLI or API** – Use the describe-volumes CLI command or the DescribeVolumes API operation.

Updating an Amazon FSx for OpenZFS volume

This section provides information on how to update a volume using the Amazon FSx console, the AWS CLI, and the Amazon FSx API, and details on the volume properties that you can modify.

Topics

- Updating a volume (Amazon FSx console, AWS CLI, and Amazon FSx API)
- Modifiable volume properties

Updating a volume (Amazon FSx console, AWS CLI, and Amazon FSx API)

To update a volume configuration (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems** and choose the FSx for OpenZFS file system that you want to update a volume for.
3. Choose the **Volumes** tab.
4. Choose the volume that you want to update.
5. For **Actions**, choose **Update volume**.

   The **Update volume** dialog box displays with the volume's current settings.
6. For **Storage capacity quota - optional**, you can set a quota that will be the maximum storage size for the volume. This quota cannot be larger than the parent volume quota. To set no quota and allow this volume to consume any available capacity in your file system, set this property to `-1`.
7. For **Storage capacity reservation - optional**, you can enter the reservation for the volume. This reserves dedicated space on the file system storage pool, meaning the space available to all other volumes is reduced by the amount specified. It cannot be set to a value that's greater than the parent volume quota or the remaining reservation space on the file system storage. To set no reservation and allow this volume to consume any available capacity in your file system, set this property to `0` or `-1`.

8. For **Data compression type**, choose the type of compression to use for your volume—either **Zstandard**, **LZ4**, or **No compression**. Zstandard compression provides more data compression and higher read throughput than LZ4 compression. LZ4 compression provides less compression and higher write throughput performance than Zstandard compression. Changing this property affects only newly-written data. For more information about the storage and performance benefits of volume data compression options, see Data compression.
9. For **NFS exports**, you can modify or remove the existing client configurations that define which clients can access the volume and what permissions they have.

   You can provide additional client configurations for the volume:

   a. In the **Client addresses** field, specify which clients can access the volume. Enter an asterisk (*) for any client, a specific IP address, or a CIDR range of IP addresses.

   b. In the **NFS options** field, enter a comma-delimited set of exports options. For example, enter `rw` to allow read and write permissions to the volume.

   c. Choose **Add client configuration**.

   d. Repeat the procedure to add another client configuration.

   For more information, see NFS exports.
10. For **Record size**, choose whether to use the default suggested record size of 128 KiB, or to set a **User-configured** suggested record size for the volume. Generally, workloads that write in fixed small or large record sizes may benefit from setting a custom record size, like database workloads (small record size) or media streaming workloads (large record size). We recommend the default setting for the majority of use cases. For more information about the record size setting, see Configurable volume properties.
11. For **User and group quotas**, you can change or set a storage quota for a user or group:

    a. For **Quota type**, choose USER or GROUP.

    b. For **User or group ID**, enter a number that is the ID of the user or group.

    c. For **Usage quota**, enter a number that is the storage quota in GiB of the user or group.

    d. To create additional user or group quotas, choose **Add quota** and repeat the procedure to change or add a quota for another user or group.
12. For **Record size**, enter the desired maximum size of a logical block in a ZFS dataset. Valid values are 4, 8, 16, 32, 64, 128, 256, 512, or 1024 KiB. The default is 128 KiB. Most file systems will use the default value. Database workflows can benefit from a smaller record size, while

streaming workflows can benefit from a larger record size. For more information about Record size and file system performance, see ZFS record size.

13. Choose **Update** to update the volume.

To update a volume configuration (CLI)

- To update the configuration of an FSx for OpenZFS volume, use the update-volume CLI command (or the equivalent UpdateVolume API operation), as shown in the following example. In this example, the `StorageCapacityQuota` and `StorageCapacityReservationGiB` properties are unset, turning off the quota capacity reservation for the volume.

```
aws fsx update-volume \
    --volume-id fsxvol-0123456789abcdef0 \
    --open-zfs-configuration '{
      "DataCompressionType": "ZSTD",
      "NfsExports": [
         {
            "ClientConfigurations": [
               {
                  "Clients": "192.0.2.0/24",
                  "Options": [ "crossmnt" ]
               }
            ]
         }
      ],
      "RecordSizeKiB": 128,
      "StorageCapacityQuotaGiB": -1,
      "StorageCapacityReservationGiB": -1,
      "UserAndGroupQuotas": [
         {
            "Id": 1102,
            "StorageCapacityQuotaGiB": 2000,
            "Type": "GROUP"
         }
      ]
   }'
```

Modifiable volume properties

You can update the following FSx for OpenZFS volume properties:

- **Storage capacity quota** – You can change an existing quota setting, set a new quota, or unset the capacity quota.
- **Storage capacity reservation** – You can change an existing storage capacity setting, enable storage capacity reservation, or unset the capacity reservation.
- **Data compression type** – You can enable compression, change the compression type, and turn compression off. Changing this property affects only newly-written data.
- **NFS exports** – You can manage access to the volume by modifying **Client addresses** and **NFS options**. For more information, see NFS exports.
- **User and group quotas** – You can change or remove existing quotas and add new quotas.
- **Volume record size** – You can change the record size for the volume. Changing the volume record size affects only files created afterward; existing files are unaffected. For more information, see ZFS record size.

Deleting an Amazon FSx for OpenZFS volume

This section provides information on how to delete an FSx for OpenZFS volume using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

When you delete a volume, the operation has the following results:

- Deletes all data in the volume
- Deletes all snapshots
- Deletes all child volumes and their snapshots

> **Note**
>
> You cannot delete a root volume. You also cannot delete any volumes from a client mount (for example, with a `rm -r /fsx/vol1` command).

Before you delete a volume, make sure that no applications are accessing the data in the volume that you want to delete.

To delete a volume (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **File systems**, and then choose the FSx for OpenZFS file system that you want to delete a volume from.
3. Choose the **Volumes** tab.
4. Choose the volume that you want to delete.
5. For **Actions**, choose **Delete volume**.
6. In the delete dialog, do the following:

   a. Acknowledge that you understand the results of deleting the volume.

   b. Confirm the volume deletion by entering **delete** in the **Confirm delete** field.
7. Choose **Delete volume(s)**.

To delete a volume (CLI)

- To delete an FSx for OpenZFS volume, use the delete-volume CLI command (or the equivalent DeleteVolume API operation), as shown in the following example.

```
aws fsx delete-volume --volume-id fsxvol-0123456789abcdef1
```

Tag your Amazon FSx for OpenZFS resources

To help you manage your file systems and other Amazon FSx resources, you can assign your own metadata to each resource in the form of tags. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Topics

- Tag basics
- Tagging your resources
- Tag restrictions
- Permissions and tag

Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's Amazon FSx file systems that helps you track each instance's owner and stack level.

We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper Tagging Best Practices.

Tags don't have any semantic meaning to Amazon FSx and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

If you're using the Amazon FSx API, the AWS CLI, or an AWS SDK, you can use the `TagResource` API action to apply tags to existing resources. Additionally, some resource-creating actions enable

you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, we roll back the resource creation process. This ensures that resources are either created with tags or not created at all, and that no resources are left untagged at any time. By tagging resources at the time of creation, you can eliminate the need to run custom tagging scripts after resource creation. For more information about enabling users to tag resources on creation, see Grant permission to tag resources during creation.

Tagging your resources

You can tag Amazon FSx resources that exist in your account. If you're using the Amazon FSx console, you can apply tags to resources by using the Tags tab on the relevant resource screen. When you create resources, you can apply the Name key with a value, and you can apply tags of your choice when creating a new file system. The console may organize resources according to the Name tag, but this tag doesn't have any semantic meaning to the Amazon FSx service.

You can apply tag-based resource-level permissions in your IAM policies to the Amazon FSx API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `TagResource` and `UntagResource` Amazon FSx API actions in your IAM policies to control which tag keys and values are set on your existing resources.

For more information about tagging your resources for billing, see Using cost allocation tags in the *AWS Billing User Guide*.

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8

- Maximum value length – 256 Unicode characters in UTF-8
- The allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.

You can't delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete a file system that you tagged with a tag key called `DeleteMe`, you must use the `DeleteFileSystem` action with the file system resource identifier, such as fs-1234567890abcdef0.

When you tag public or shared resources, the tags you assign are available only to your AWS account; no other AWS account will have access to those tags. For tag-based access control to shared resources, each AWS account must assign its own set of tags to control access to the resource.

Permissions and tag

For more information about the permissions required to tag Amazon FSx resources at creation, see Grant permission to tag resources during creation.

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.

Protecting your Amazon FSx for OpenZFS data

Amazon FSx for OpenZFS provides you with the following options to further protect the data stored on your file systems:

- **Built-in Amazon FSx backups** – Supports your backup retention and compliance needs within Amazon FSx, offering both automatic daily backups and user-initiated backups.
- **Snapshots** – Enables your users to easily undo file changes and compare file versions by restoring files to previous versions.
- **On-demand data replication** – Makes it easy to replicate datasets across file systems, within or across AWS Regions and accounts.
- **AWS Backup** – Creates backups of your FSx for OpenZFS file system that work as part of a centralized and automated backup solution across AWS services in the cloud and on premises.

Topics

- Protecting your data with backups
- Protecting your data with snapshots
- Protecting your data with on-demand replication

Protecting your data with backups

With FSx for OpenZFS, backups are file-system-consistent, highly durable, and incremental. To ensure high durability, Amazon FSx stores backups in Amazon Simple Storage Service (Amazon S3).

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

Creating regular backups for your file system is a best practice that complements the replication that FSx for OpenZFS performs for your file system. Amazon FSx backups help support your

backup retention and compliance needs. Working with Amazon FSx backups is easy, whether it's creating backups, copying a backup, restoring a file system from a backup, or deleting a backup.

Topics

- Working with automatic daily backups
- Working with user-initiated backups
- Working with AWS Backup
- Copying backups
- Restoring backups
- Deleting backups

Working with automatic daily backups

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

Automatic daily backups are kept for a certain period of time, known as a retention period. 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 1–90 days. Automatic daily backups are deleted when the file system is deleted.

> **Note**
>
> While automatic daily backups have a maximum retention period of 90 days, user-initiated backups are kept forever, unless you delete them. For more information about user-initiated backups, see Working with user-initiated backups.

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.

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.

Creating user-initiated backups

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

To create a user-initiated file system backup

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

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

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

Working with AWS Backup

AWS Backup is a simple and cost-effective way to protect your data by backing up your Amazon FSx for OpenZFS file systems. AWS Backup is a unified backup service designed to simplify the creation, 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 file systems, 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 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, are incremental relative to any other Amazon FSx backups you take of your file system (user-initiated or automatic), and offer the same restore options as backups taken through the Amazon FSx console. If you use AWS Backup to manage these backups, you gain additional functionality, such as unlimited retention options and the ability to create scheduled backups as frequently as every hour. In addition, AWS Backup and Amazon FSx retain 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 view 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 and how to delete backups, see Working with Amazon FSx file systems in the *AWS Backup Developer Guide*.

Deleting backups

You can't delete backups taken by AWS Backup in the Amazon FSx console, CLI, or API. For information on deleting a backup taken by AWS Backup, see Backup deletion in the *AWS Backup Developer Guide*. 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.

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`.

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.

Backup copy limitations

The following are some limitations when you copy backups:

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

Permissions for cross-Region backup copies

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

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

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

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

Full and incremental copies

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

To copy a backup (Amazon FSx 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 (AWS CLI)

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

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

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

  The response shows the description of the copied backup.

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

Restoring backups

You can use an available backup to create a new file system, effectively restoring a point-in-time snapshot of another file system. You can restore a backup using the Amazon FSx console, AWS CLI, or one of the AWS SDKs. You can also restore backups using the AWS Backup console. For more information about using the AWS Backup console, see Restore an FSx file System in the AWS Backup Developer Guide.

> **Note**
>
> Backups of Multi-AZ FSx for OpenZFS file systems can be restored only by using the Amazon FSx console or CLI.

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.

> **Note**
>
> Restoring from a backup with a given storage class to a file system with a different storage class is not supported.

To restore a file system from backup (Amazon FSx 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 restore from the **Backups** table, and then choose **Restore backup**. Doing so opens the file system creation wizard.
4. For **File system name - optional**, you can enter a name using a maximum of 256 Unicode letters, white space, and numbers, plus these special characters: + - = . _ : /
5. For **Storage capacity**, enter a value that is equal to or greater than the storage capacity of the original file of which the backup was taken, in GiB. The range of valid values is 64–524288 GiB.
6. For **Provisioned SSD IOPS**, you have two options to provision the number of IOPS for your file system:

   - Choose **Automatic** (the default) if you want Amazon FSx to automatically provision 3 IOPS per GiB of SSD storage.
   - Choose **User-provisioned** if you want to specify the number of IOPS. You can provision a maximum of 160,000 SSD IOPS per file system for Single-AZ 1 (non-HA and HA) and a maximum of 400,000 SSD IOPS per file system for Single-AZ 2 (non-HA and HA) and Multi-AZ (HA)*. You pay for SSD IOPS that you provision that exceed 3 IOPS per GiB of SSD storage.

   > **Note**
   >
   > *The maximum SSD IOPS you can provision for Multi-AZ file systems depends on the AWS Region your file system is located in. For more information, see Data access from disk.

7. For **Throughput capacity**, you have two options to provide your desired throughput capacity in megabytes per second (MBps). **Throughput capacity** is the sustained speed at which the file server that hosts your file system can serve data.

- Choose **Recommended throughput capacity** (the default) if you want Amazon FSx to automatically choose the throughput capacity. The recommended value is based on the amount of storage capacity that you chose.
- Choose **Specify throughput capacity** if you want to specify the throughput capacity value, and choose a value of 64, 128, 256, 512, 1024, 2048, 3072, or 4096 MBps. You pay for additional throughput capacity that you provision above the recommended amount.

You can increase the amount of throughput capacity as needed at any time after you create the file system. For more information, see Modifying throughput capacity.

8. In the **Network & security** section, provide networking and security group information:

- For **Virtual Private Cloud (VPC)**, choose the Amazon VPC that you want to associate with your file system.
- For **VPC Security Groups**, the ID for the default security group for your VPC should be already added.
- For **Subnet**, choose any value from the list of available subnets.

9. In the **Encryption** section, for **Encryption key**, choose the AWS Key Management Service (AWS KMS) encryption key that protects your file system's data at rest.

> **Note**
>
> When restoring a file system with a storage capacity greater than 144 TiB, you must use the same KMS key that you used when creating the backup. If you want to restore the file system with a new KMS key, you must first copy the backup using the new KMS key, and then restore the file system with this backup.

10. In **Root volume configuration**, you can set the following options for the file system's root volume:

- For **Data compression type**, choose the type of compression to use for your volume, either **Zstandard**, **LZ4**, or **No compression**. Zstandard compression provides more data compression and higher read throughput than LZ4 compression. LZ4 compression provides less compression and higher write throughput performance than Zstandard compression. For more information about the storage and performance benefits of the volume data compression options, see Data compression.

- For **Copy tags to snapshots**, enable or disable the option to copy tags to the volume's snapshot.
- For **NFS exports**, there is a default client configuration setting which you can modify or remove. Client configurations define which clients can access the volume and their permissions.

  To provide additional client configurations:

  1. In the **Client addresses** field, specify which clients can access the volume. Enter an asterisk (*) for any client, a specific IP address, or a CIDR range of IP addresses.
  2. In the **NFS options** field, enter a comma-delimited set of exports options. For example, enter `rw` to allow read and write permissions to the volume for the specified **Client addresses**.
  3. Choose **Add client configuration**.
  4. Repeat the procedure to add another client configuration.

  For more information, see NFS exports.
- For **Record size**, choose whether to use the default suggested record size of 128 KiB, or to set a custom suggested record size for the volume. Generally, workloads that write in fixed small or large record sizes may benefit from setting a custom record size, like database workloads (small record size) or media streaming workloads (large record size). We recommend using the default setting for the majority of use cases. For more information about the record size setting, see Configurable volume properties.
- For **User and group quotas**, you can set a storage quota for a user or group:

  1. For **Quota type**, choose USER or GROUP.
  2. For **User or group ID**, choose a number that is the ID of the user or group.
  3. For **Usage quota**, choose a number that is the storage quota of the user or group.
  4. Choose **Add quota**.
  5. Repeat the procedure to add a quota for another user or group.

11. In **Backup and maintenance - *optional***, you can set the following options:

- For **Daily automatic backup**, choose **Enabled** for automatic daily backups. This option is enabled by default.
- For **Daily automatic backup window**, set 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.

- For **Automatic backup retention period**, set a period from 1–90 days that you want to retain automatic backups.
- For **Weekly maintenance window**, you can set 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.

12. For **Tags - *optional***, you can enter a key and value to add tags to your file system. A tag is a case-sensitive key-value pair that helps you manage, filter, and search for your file system.

    Choose **Next**.

13. Review the file system configuration shown on the **Create file system** page. For your reference, note which file system settings you can modify after the file system is created.
14. Choose **Create file system**.

Deleting backups

Deleting a backup is a permanent, unrecoverable action. Any data in a deleted backup is also deleted. Do not delete a backup unless you're sure you won't need that backup again in the future.

> **Note**
>
> You can't delete backups taken by AWS Backup in the Amazon FSx console, CLI, or API. For information on deleting a backup taken by AWS Backup, see Deleting Backups in the*AWS Backup Developer Guide*.

To delete a backup

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose **Backups** from the left side navigation.
3. From the **Backups** table, choose the backup that you want to delete.
4. For **Actions**, choose **Delete backup**.
5. In the **Delete backups** dialog box that opens, confirm that the ID of the backup identifies the backup that you want to delete.

6. Confirm that the check box is checked for the backup that you want to delete.
7. Choose **Delete backups**.

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

Protecting your data with snapshots

A *snapshot* is a read-only image of an FSx for OpenZFS volume at a point in time. Snapshots offer protection against accidental deletion or modification of files in your volumes. With snapshots, your users can easily view and restore individual folders and files from an earlier snapshot. Doing this enables users to easily undo changes and compare file versions.

Because snapshots are stored alongside your file system's data, they consume the file system's storage capacity. However, snapshots consume storage capacity only for the changed portions of files since the last snapshot.

Snapshots are stored in the `.zfs/snapshot` directory at the root of a volume.

Topics

- Using snapshots to create volumes
- Creating a snapshot
- Deleting a snapshot
- Viewing a snapshot
- Restoring a volume from a snapshot
- Restoring individual files and folders
- Setting up a custom snapshot schedule

Using snapshots to create volumes

You can use a snapshot to create a clone volume or a full-copy volume.

A clone volume is a writable copy that is initialized with the same data as the snapshot from which it was created. Clone volumes provide an easy way to support multiple users or applications in parallel from a shared dataset, as well as quickly test new changes to your databases or applications. Clone volumes are created almost instantly and initially consume no additional storage capacity. (They only consume capacity for incremental changes to the source snapshot.)

However, each clone volume maintains a dependency on its source snapshot, so you cannot delete this source snapshot while the clone volume is in use. To split a clone from its source snapshot and remove this dependency, you must create a new full-copy volume from that clone.

A full-copy volume is also initialized with the same data as its source snapshot, but is a fully independent writable copy. However, because a full-copy volume requires transferring all of the data from the source snapshot, it can take a significantly longer time to create than a clone volume. Once a full-copy volume is created, it is identical to a standard FSx for OpenZFS volume and does not maintain any relationship to its source snapshot.

For more information on creating clone and full-copy volumes, see Managing Amazon FSx for OpenZFS volumes.

Creating a snapshot

You can create an FSx for OpenZFS snapshot using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

To create a snapshot (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, under **OpenZFS**, choose **Snapshots**. Then in the **Snapshots** pane, choose **Create snapshot**.

   You can also create a snapshot directly by navigating to the **Snapshots** tab of an individual volume and choosing **Create snapshot**.
3. In the **File system** field of the **Create snapshot** dialog box, choose the file system to create the snapshot on.
4. In the **Volume** field, choose an existing volume you want to take a snapshot of.
5. In the **Snapshot name** field, provide a name for the snapshot. You can use a maximum of 203 alphanumeric characters, and the special characters . - _ :
6. Choose **Confirm** to create the snapshot.

The snapshot is ready for use when its status is **Available**.

To create a snapshot (CLI)

- To create an FSx for OpenZFS snapshot, use the create-snapshot CLI command (or the equivalent CreateSnapshot API operation), as shown in the following example.

```
aws fsx create-snapshot \
    --volume-id fsvol-123 \
    --name snapshot2
```

The command example uses the following parameters:

- `volume-id` - The ID of the volume that you are taking a snapshot of.
- `name` - The name of the source snapshot.

After successfully creating the snapshot, Amazon FSx returns its description in JSON format.

Deleting a snapshot

You can delete an FSx for OpenZFS snapshot using the Amazon FSx console, the AWS CLI, and the Amazon FSx API. After deletion, the snapshot no longer exists, and its data is gone. Deleting a snapshot doesn't affect snapshots stored in a file system backup.

> **Note**
>
> A snapshot can't be deleted if it was previously cloned and that clone is still available. Before you can delete the snapshot, you must first delete all of the snapshot's clones.

To delete a snapshot (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, under **OpenZFS**,choose **Snapshots**.
3. In the **Snapshots** page, choose the snapshot that you want to delete.
4. In the **Summary** page for the snapshot, choose **Delete**.

To delete a snapshot (CLI)

- To delete an FSx for OpenZFS volume, use the delete-volume CLI command (or the equivalent DeleteVolume API operation), as shown in the following example.

```
aws fsx delete-snapshot --snapshot-id fsvolsnap-1234
```

Viewing a snapshot

You can see the FSx for OpenZFS volumes that are currently on your file system using the Amazon FSx console, the AWS CLI, and the Amazon FSx API and SDKs.

To view the snapshots on your file system:

- **Using the console** – Choose a file system to view the **File systems** detail page. Choose the **Volumes** tab to list all the volumes on the file system, and then choose a volume. The volume's **Summary** page has a **Snapshots** tab that lists the snapshots for the volume.
- **Using the CLI or API** – Use the describe-snapshots CLI command or the DescribeSnapshots API operation.

Restoring a volume from a snapshot

You can return a volume to a state saved by a specified snapshot. You use the restore-volume-from-snapshot CLI command (or the equivalent RestoreVolumeFromSnapshot API operation), as shown in the following example.

```
aws fsx restore-volume-from-snapshot \
    --volume-id fsvol-12345 \
    --snapshot-id fsvolsnap-67890 \
    --options DELETE_INTERMEDIATE_SNAPSHOTS DELETE_CLONED_VOLUMES
```

The command example uses the following parameters:

- `volume-id` - The ID of the volume that you are restoring.
- `snapshot-id` - The ID of the source snapshot. Specifies the snapshot you are restoring from.
- `DELETE_INTERMEDIATE_SNAPSHOTS` - Deletes snapshots between the current state and the specified snapshot. `restore-volume-from-snapshot` will fail if there are intermediate snapshots and this option isn't used.
- `DELETE_CLONED_VOLUMES` - Deletes any volumes cloned from this volume. `restore-volume-from-snapshot` will fail if there are any cloned volumes and this option isn't used.

Restoring individual files and folders

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

Linux, macOS, and Windows clients can view snapshots in the `.zfs/snapshot` directory hidden at the root of a volume. The `.zfs` directory is hidden, so it will not appear in any `ls` or `dir` results. You must also specify the `crossmnt` option in the `NFS exports` configuration of your FSx for OpenZFS volume to enable your clients to navigate to this directory.

To restore a file from a snapshot (Linux, macOS, and Windows clients)

1. If the original file still exists and you do not want it overwritten by the file in a snapshot, then use your Linux, macOS, or Windows client to rename the original file or move it to a different directory.
2. In the `.zfs/snapshot` directory, locate the snapshot that contains the version of the file that you want to restore.
3. Copy the file from the `.zfs/snapshot` directory to the directory in which the file originally existed.

Data in snapshots is read only. If you want to make modifications to files and folders in a snapshot, you must save a copy of the files and folders to a writable location and make modifications to these copies.

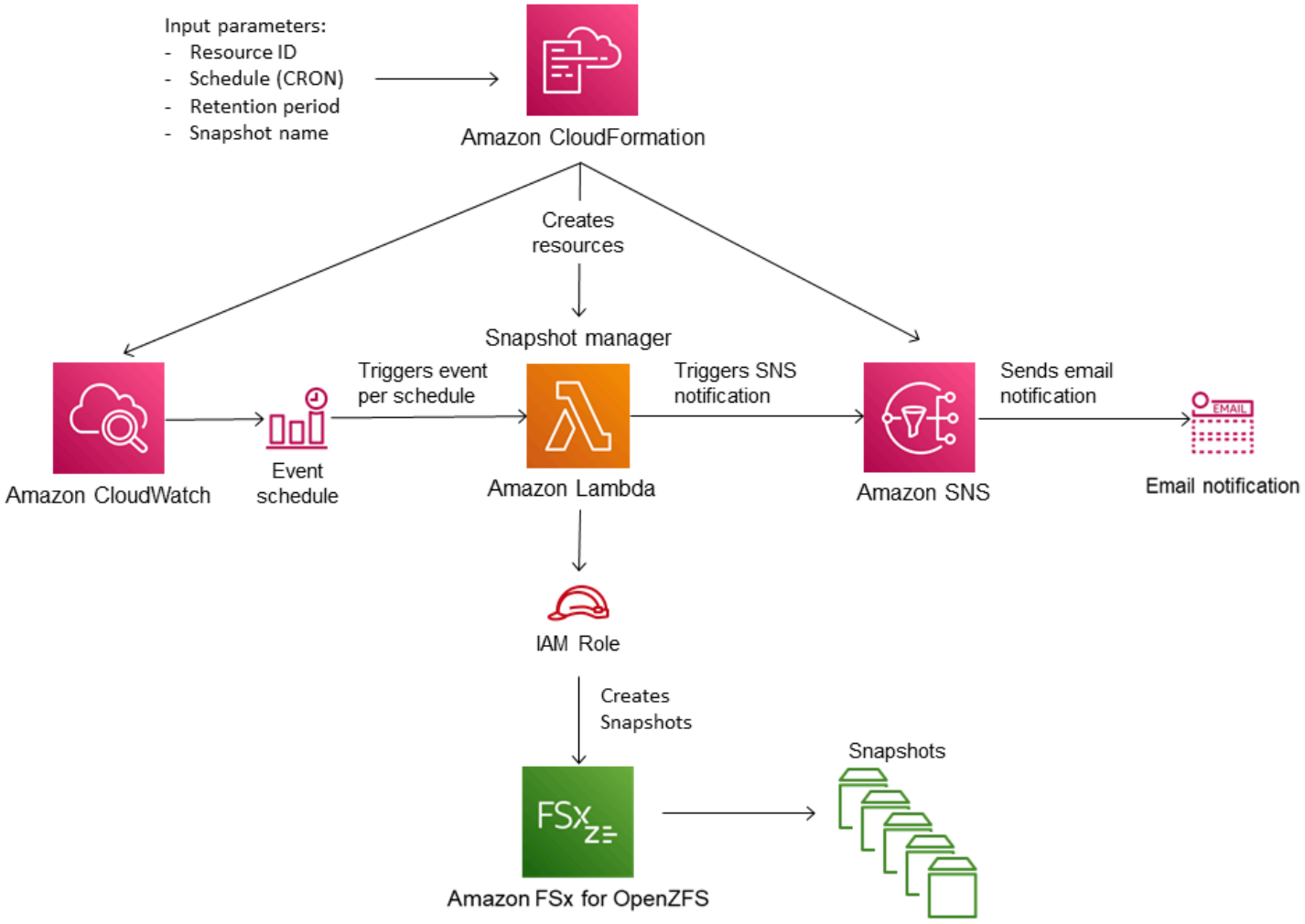
Setting up a custom snapshot schedule

You can set up a automated custom snapshot schedule for FSx for OpenZFS volumes using the resources and configuration template provided in this topic. The custom snapshot scheduling solution performs user-initiated snapshots of your Amazon FSx volumes on a custom schedule that you define. For example, you can configure a custom schedule to take a snapshot every hour and automatically delete snapshots that are older than two days.

For more information on CRON schedule patterns, see Schedule expressions for rules in the *Amazon CloudWatch Events User Guide*.

Architecture overview

Deploying this solution builds the following resources in the AWS Cloud:



The diagram illustrates the following custom snapshot schedule workflow:

1. The solution AWS CloudFormation template deploys an CloudWatch Event, an AWS Lambda function, an Amazon Simple Notification Service (Amazon SNS) queue, and an IAM role. The IAM role gives the Lambda function permission to invoke the necessary Amazon FSx API operations.
2. The CloudWatch event runs on a schedule you define as a CRON pattern, during the initial deployment. This event invokes the solution's snapshot manager Lambda function that invokes the Amazon FSx `CreateSnapshot` API operation to initiate a snapshot.
3. The snapshot manager retrieves a list of existing user-initiated snapshots for the specified volume using `DescribeSnapshots`. It then deletes snapshots older than the retention period, which you specify during the initial deployment.

4. The snapshot manager sends a notification message to the Amazon SNS queue on a successful snapshot if you choose the option to be notified during the initial deployment. A notification is always sent in the event of a failure.

Required permissions

The following permissions are required to use the custom snapshot schedule AWS CloudFormation template:

- `AWSCloudFormationFullAccess`
- `AmazonS3FullAccess`
- `AmazonEventBridgeFullAccess`
- `IAMFullAccess`
- `AmazonSNSFullAccess`
- `AWSKeyManagementServicePowerUser`
- `AWSLambda_FullAccess`

AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Amazon FSx custom snapshot scheduling solution. To use this solution, download the fsx-openzfs-scheduled-snapshot.template AWS CloudFormation template.

Automated deployment

The following procedure configures and deploys this custom snapshot scheduling solution. It takes about five minutes to deploy. Before you start, you must have the ID of a volume on an Amazon FSx file system running in an Amazon Virtual Private Cloud (Amazon VPC) in your AWS account. For more information on creating these resources, see Creating an Amazon FSx for OpenZFS volume.

> **Note**
>
> Implementing this solution incurs billing for the associated AWS services. For more information, see the pricing details pages for those services.

To launch the custom snapshot solution stack

1. Download the fsx-openzfs-scheduled-snapshot.template AWS CloudFormation template. For more information on creating an AWS CloudFormation stack, see Creating a stack on the AWS CloudFormation console in the *AWS CloudFormation User Guide*.

   > **Note**
   >
   > By default, this template launches in the US East (N. Virginia) AWS Region. Amazon FSx for OpenZFS is currently only available in specific AWS Regions. You must launch this solution in an AWS Region where FSx for OpenZFS is available. For more information, see Amazon FSx endpoints and quotas in the *AWS General Reference*.

2. For **Parameters**, review the parameters for the template and modify them for the needs of your file system volumes. This solution uses the following default values.

Parameter	Default	Description
FSx for OpenZFS resource ID	No default value	The file system ID or volume ID on which the snapshot schedule will apply. If you provide a file system ID, the schedule will take snapshots of all volumes within that file system.
CRON schedule pattern for snapshots	0 0/6 * * ? * [Every 6 hours]	The schedule to run the CloudWatch event, triggering a new snapshot and deleting old snapshots outside of the retention period.
Snapshot retention (days)	7	The number of days to keep user-initiated snapshots. The Lambda function deletes user-initiated

Parameter	Default	Description
		snapshots older than this number of days.
Name for snapshots	User-scheduled_snapshot	The name for these snapshots, which appears in the **Snapshot Name** column of the Amazon FSx Management Console.
Snapshot Notification	Yes	Choose whether to be notified when snapshots are successfully initiated. A notification is always sent if there's an error.
Email address	No default value	The email address to use in subscribing to the SNS notifications.

3. Choose **Next**.
4. For **Options**, choose **Next**.
5. For **Review**, review and confirm the settings. Select the check box acknowledging that the template creates IAM resources.
6. Choose **Create** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in about five minutes.

Additional options

You can use the Lambda function created by this solution to perform custom scheduled snapshots of more than one FSx for OpenZFS volume. The volume ID is passed to the Amazon FSx function in the input JSON for the CloudWatch event. The default JSON passed to the Lambda function is as follows, where the values for `VolumeId` and `SuccessNotification` are passed from the parameters specified when launching the AWS CloudFormation stack.

```
{
 "start-snapshot": "true",
 "purge-snapshots": "true",
 "volume-id": "${VolumeId}",
 "notify_on_success": "${SuccessNotification}"
}
```

To schedule snapshots for an additional FSx for OpenZFS volume, create another CloudWatch event rule. You do so using the Schedule event source, with the Lambda function created by this solution as the target. Choose **Constant (JSON text)** under **Configure Input**. For the JSON input, simply substitute the volume ID of the FSx for OpenZFS volume to back up in place of `${VolumeId}`. Also, substitute either `Yes` or `No` in place of `${SuccessNotification}` in the JSON above.

Any additional CloudWatch Event rules you create manually aren't part of the AWS CloudFormation stack for the Amazon FSx custom scheduled snapshot solution. Thus, they aren't removed if you delete the stack.

Protecting your data with on-demand replication

Amazon FSx for OpenZFS supports on-demand data replication, enabling you to transfer snapshots of data between file systems within and across AWS Regions and accounts. You can use on-demand data replication for a variety of tasks such as:

- Synchronizing or distributing data to your development or test environments.
- Establishing and maintaining read replicas to provide scale-out read performance.
- Maintaining a passive standby file system for use in disaster recovery cases.

With on-demand data replication, Amazon FSx automatically establishes and maintains network connectivity between file systems to handle interruptions and resume data transfer as needed. Amazon FSx also encrypts data in transit and at rest and integrates with AWS RAM to authorize accesss to volumes for data replication across AWS accounts. For more information, see Shareable AWS resources in the *AWS RAM User Guide*.

On-demand data replication is available for all deployment types in AWS Regions where Amazon FSx for OpenZFS is available. For more information, see Availability by AWS Region.

Topics

- Prerequisites for using on-demand data replication
- Performance considerations for on-demand data replication
- Using on-demand data replication
- Monitoring progress of on-demand data replication
- Setting up ongoing periodic data replication

Prerequisites for using on-demand data replication

Before using on-demand data replication, make sure that you have met the following prerequisites.

- Single-AZ 1 (non-HA and HA) file systems must have a provisioned throughput capacity of 256 MBps or above. It is also recommended that Single-AZ 1 (non-HA and HA) file systems have a provisioned SSD IOPS level of 6,000 or above.
- Single-AZ 2 (non-HA and HA) and Multi-AZ (HA) file systems must have a provisioned throughput capacity of 160 MBps or above. It is also recommended that Single-AZ 2 (non-HA) and Multi-AZ (HA) file systems have a provisioned SSD IOPS level of 6,000 or above.
- Users or roles must have permission to take the CreateVolume and CopySnapshotAndUpdateVolume actions in an AWS account. You can control these permissions by using AWS Identity and Access Management (IAM) policies. For more information, see Actions, resources, and condition keys for Amazon FSx in the *Service Authorization Reference*.
- To replicate data across file systems in different AWS accounts, the source account must have, at minimum, permission to take the **fsx:PutResourcePolicy**, **fsx:GetResourcePolicy**, and **fsx:DeleteResourcePolicy** actions. The source account must also have permissions to share resources on AWS RAM. To grant these permissions, you can directly attach the AmazonFSxFullAccess, AmazonFSxConsoleFullAccess, and AWSResourceAccessManagerFullAccess AWS managed policies to your IAM roles, groups, and users. The destination account must have the AWSResourceAccessManagerResourceShareParticipantAccess AWS managed policy attached to its IAM roles, groups, and users.

Performance considerations for on-demand data replication

On-demand data replication shares provisioned throughput with other file system clients. To accommodate data replication activity without impacting other workloads, we recommend provisioning twice the level of throughput capacity that your workload normally needs. You can use Amazon CloudWatch metrics with FSx for OpenZFS to monitor your file system's performance

utilization and scale up your file system's performance as needed to avoid slowing down your ongoing workloads. For more information, see Using Amazon FSx for OpenZFS CloudWatch metrics.

Using on-demand data replication

On-demand data replication only transfers data from the indicated source snapshot, which does not include data from child volumes. To transfer data from child volumes, you must initiate additional data replication jobs using source snapshots from the child volumes.

Each file system can only be used as the source file system or the destination file system for one on-demand data replication task at a time. You must wait until the first on-demand replication task is completed or cancelled before initating another request. You can only have a maximum of twenty concurrent cross-file system replication jobs per account, per AWS Region.

Replicating data across file systems on the same account

You can create or update a replica volume across file systems that are on the same AWS account by using the Amazon FSx Console, API, or CLI.

To update a volume from a snapshot (Console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the left navigation pane, choose **Volumes**, and then choose the volume that you would like to use as your destination volume.
3. For **Actions**, choose **Update volume with snapshot**. The **Copy snapshot and update volume** panel displays.
4. Choose the source region of the snapshot
5. Choose the snapshot that you would like to update the volume from.
6. For **Source snapshot copy strategy**, choose **Incremental copy** or **Full copy**. An incremental copy returns the destination volume to the most recent common ancestor that it shares with the source volume and then updates the destination volume, transferring only the data that is not already included in the most recent common ancestor. A full copy will remove any clones, snapshots, and intermediate data on the destination volume and transfer all of the data from the source volume. During incremental copy, your destination volume will be read-only. During full copy, your destination volume will be unmounted and automatically remounted after the transfer is completed.

7. If the destination volume has any **intermediate clones**, **dependent snapshots**, or **intermediate data**, select the checkboxes to delete them. If you are using incremental copy, you must delete all descendent data for the update to succeed.
8. Choose **Update** to update the volume.

To update a volume from a snapshot (CLI)

- To update an FSx for OpenZFS volume with a snapshot, use the copy-snapshot-and-update-volume CLI command, or the equivalent CopySnapshotAndUpdateVolume API command, and specify the following properties:
  - `--volume-id` – The ID of the volume that you would like to update.
  - `--source-snapshot-arn` – The ARN of the source snapshot.
  - `--options` – Any intermediate clones, dependent snapshots, or intermediate data that need to be deleted. Valid values are `DELETE_INTERMEDIATE_SNAPSHOTS`, `DELETE_CLONED_VOLUMES`, and `DELETE_INTERMEDIATE_DATA`.
  - `--copy-strategy` – Strategy used to copy data from the source volume. Value values are `FULL_COPY` and `INCREMENTAL_COPY`.

The following example shows how to update a volume with a snapshot using incremental copy and deleting all intermediate clones, dependent snapshots, and intermediate data.

```
aws fsx copy-snapshot-and-update-volume \
     --volume-id fsvol-1234567890abcdef0 \
     --source-snapshot-arn arn:aws:fsx:555555555555:snapshot/fsvol-1234567890abcdef0/
fsvolsnap-021345abcdef6789\
     --options DELETE_INTERMEDIATE_SNAPSHOTS DELETE_CLONED_VOLUMES
 DELETE_INTERMEDIATE_DATA\
     --copy-strategy INCREMENTAL_COPY
```

The example above returns the following response.

```
{
    "VolumeId": "fsvol-1234567890abcdef0",
    "Lifecycle": "AVAILABLE",
    "AdministrativeActions": [
    {
        "AdministrativeActionType": "VOLUME_UPDATE_WITH_SNAPSHOT",
        "FailureDetails": {
```

```
            "Message": "string"
        },
        "ProgressPercent": 80,
        "RequestTime": 2023-11-03T09:26:55-07:00,
        "Status": "IN_PROGRESS",
        "TargetVolumeValues": {
            "OpenZFSConfiguration": {
            "RecordSizeKiB": 128,
            "DataCompressionType": "ZSTD",
            "DeleteIntermediateSnaphots": false,
            "DeleteClonedVolumes": false,
            "DeleteIntermediateData": true,
            "SourceSnapshotARN": "arn:aws:fsx:us-east-1:854733241892:snapshot/
fsvol-018a3d05b4d9fc768/fsvolsnap-03b43bd1942a51637",
            "DestinationSnapshot": "fsvolsnap-0f753e290e20cc974" }"
        }
    }]
}
```


Replicating data across file systems on different AWS accounts using AWS RAM

FSx for OpenZFS integrates with AWS Resource Access Manager (RAM) to allow you to replicate data across file systems that are on different AWS accounts. In the AWS Resource Access Manager (RAM) console, the owner of the source account must first enable resource sharing, and then share the source FSx for OpenZFS volume with the destination account. For more information on enabling and creating a resource share, see Enable resource sharing within AWS Organizations and Creating a resource share in the *AWS RAM User Guide*.

You will receive a shared resource invitation when the source volume has been shared with your account. Once you accept the invitation, all snapshots associated with the source volume will appear in the list of snapshots that you can replicate to a volume in the FSx for OpenZFS console. For more information, see To update a volume from a snapshot (Console). After you've created a replica volume, you can continue to update it with any of the subsequent snapshots in the source volume, as long as the source volume continues to be shared.

Monitoring progress of on-demand data replication

You can monitor the progress of your data replication using the AWS Management Console on the **Volume details** page. When you initiate a replication task, the destination snapshot will enter

the **CREATING** state. Once the data transfer is complete, the destination snapshot will become **AVAILABLE**.

You can also use the AWS CLI or Amazon FSx API to track more detailed progress of your replication by using the describe-volumes AWS CLI command or the DescribeVolumes API operation. to display the `AdministrativeActions` for the destination volume. The `AdministrativeActions` array lists the 10 most recent update actions for each administrative action type. When you initiate an on-demand data replication, a `VOLUME_UPDATE_WITH_SNAPSHOT` action is generated. Progress will be reported using the `ProgressPercent` property.

The following example shows the response for an incremental copy on-demand data replication task.

```
{
    "VolumeId": "fsvol-1234567890abcdef0",
    "Lifecycle": "AVAILABLE",
    "AdministrativeActions": [
    {
        "AdministrativeActionType": "VOLUME_UPDATE_WITH_SNAPSHOT",
        "FailureDetails": {
            "Message": "string"
        },
        "ProgressPercent": 80,
        "RequestTime": 2023-11-03T09:26:55-07:00,
        "Status": "IN_PROGRESS",
        "TotalTransferBytes": 107483152368,
        "RemainingTransferBytes": 0
        "TargetVolumeValues": {
            "OpenZFSConfiguration": {
                "SourceSnapshotARN": "stringarn:aws:fsx:555555555555:snapshot/
fsvol-1234567890abcdef0/fsvolsnap-021345abcdef6789",
                "DestinationSnapshot": "fsvolsnap-021345abcdef6789"
            }
        }
    }]
}
```

When Amazon FSx processes the request successfully, the status changes to `COMPLETED`. If the on-demand data replication task fails, the status changes to `FAILED`, and the `FailureDetails` property provides information about the failure.

Setting up ongoing periodic data replication

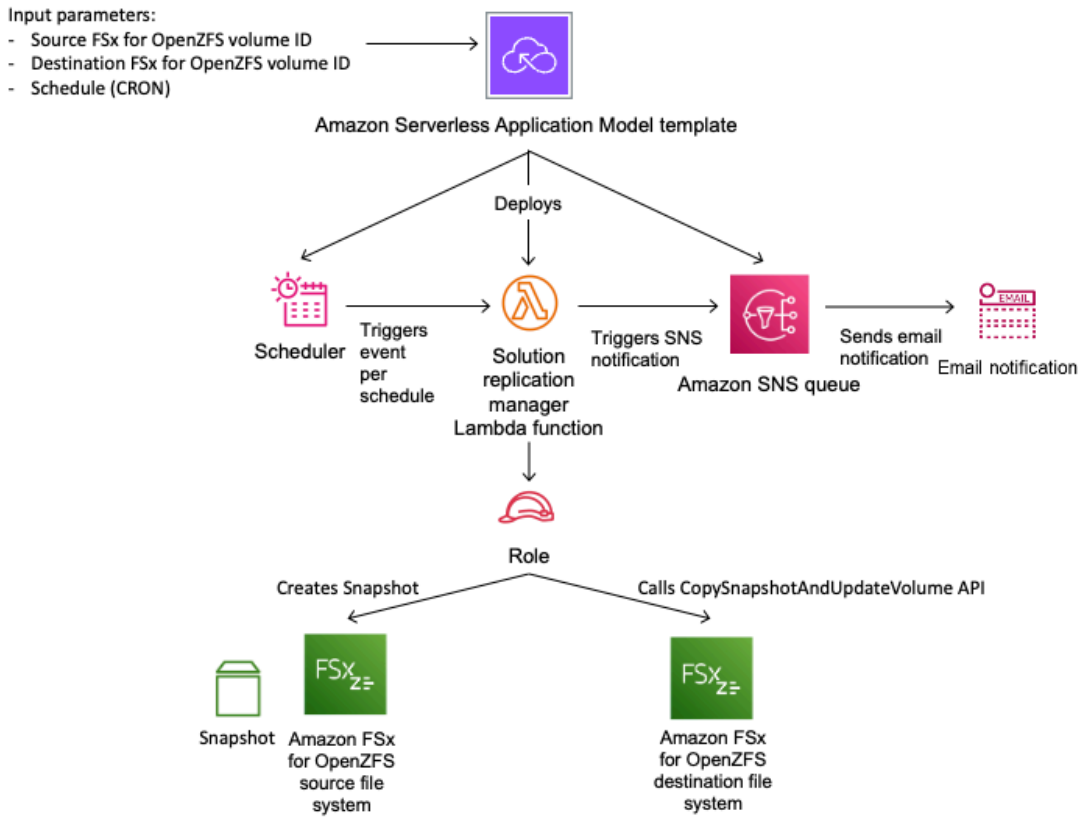
With ongoing periodic data replication, you can set up a schedule that automatically takes a snapshot of a source volume and performs an incremental replication of that snapshot on a destination volume at a certain interval, for example every 15 minutes. You can schedule ongoing periodic data replication between two volumes on FSx for OpenZFS file systems within or across AWS Regions and accounts by using the solution provided in this section.

Topics

- Architecture overview
- Required permissions
- Step 1: Initializing and deploying the application
- Step 2: Monitoring periodic replication

Architecture overview

Deploying this solution builds the following resources in the AWS Cloud.



The diagram illustrates the following periodic replication workflow.

1. AWS Serverless Application Model (SAM) automates the deployment of the FSx for OpenZFS periodic replication solution. For more information about AWS SAM, see What is the AWS Serverless Application Model (AWS SAM)? in the *AWS Serverless Application Model* User Guide.
2. The SAM template deploys an Amazon EventBridge scheduler, an AWS Lambda function, an Amazon SNS queue, and an IAM role. The IAM role gives the Lambda function permission to call the necessary Amazon FSx API operations.
3. The EventBridge scheduler runs on a schedule you specify as a cron pattern during the initial deployment. For more information about cron patterns, see Creating an Amazon EventBridge rule that runs on a schedule in the *Amazon EventBridge* User Guide. The scheduler invokes a Lambda function that calls the Amazon FSx `CreateSnapshot` API operation to create a snapshot of the source volume.
4. Once the snapshot is available, the Lambda function calls the Amazon FSx `CopySnapshotAndUpdateVolume` API operation to start replicating the source snapshot data to the destination volume.

5. The Lambda function sends a notification message to the Amazon SNS queue when replication starts, if you choose to be notified during the initial deployment. A notification is always sent when a snapshot cannot be created or the replication cannot be initiated.

Required permissions

The following permissions are required to use the custom snapshot schedule AWS CloudFormation template.

- `AmazonS3FullAccess`
- `AWSCloudFormationFullAccess`
- `AmazonEventBridgeFullAccess`
- `IAMFullAccess`
- `AmazonSNSFullAccess`
- `AWSKeyManagementServicePowerUser`
- `AWSLambda_FullAccess`

For more information about using IAM to set up permissions, see How Amazon FSx for OpenZFS works with IAM.

Step 1: Initializing and deploying the application

The following procedure configures and deploys the periodic replication solution. It takes about five minutes to deploy. Before you begin this step, make sure that you have the ID of the source and destination volumes that you would like to initiate the replication between. For more information on these resources, see Creating an Amazon FSx for OpenZFS volume, Creating a snapshot, and Using on-demand data replication.

> **Note**
>
> Implementing this solution incurs billing for the associated AWS services. For more information, see the pricing details pages for those services.

To launch the periodic replication solution stack

1. Follow the instructions on the Replicate FSx-OpenZFS volumes across file systems page to download the serverless pattern.
2. For **Parameters**, review the following parameters for the template and modify them for the needs of your periodic replication. This solution uses the following default values.

Parameter	Default	Description
Source volume ID	No default value	The ID of the source volume from which data will be periodically replicated.
Destination volume ID	No default value	The ID of the destination volume that will become a replica of the source volume.
CronSchedule	[0 0/6 **?*] (every six hours)	The schedule to replicate data from the source volume to the destination volume.
SnapshotName	fsx-scheduled-snapshot	The name for the scheduled snapshots that will be taken of the source volume. Appears in the **Snapshot Name** column of the Amazon FSx Console.
Snapshot retention (days)	7	The number of days to keep user-initiated snapshots . The Lambda function deletes user-initiatted snapshots that are kept after this number of days.

Parameter	Default	Description
SuccessNotification	Yes	Choose whether to be notified when the replication is successfully initiated. A notification is always sent when a snapshot fails to create or the replication fails to start.
Email	No default value	The email address that you would like notifications to be sent to.
CopyStrategy	INCREMENTAL_COPY	The **CopyStrategy** parameter for the `CopySnapshotAndUpdateVolume` API operation. For more information, see CopySnapshotAndUpdateVolume in the Amazon FSx API.
Options	None	The **Options** parameter for the `CopySnapshotAndUpdateVolume` API operation. For more information, see CopySnapshotAndUpdateVolume in the Amazon FSx API.

3. In the AWS SAM CLI, run the following command to deploy the resources specified in the SAM template.

```
sam deploy --guided \
--stack-name fsxz-periodic-replication \
--template-file fsx-openzfs-periodic-replication.yaml \
--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_IAM CAPABILITY_NAMED_IAM
```

You will be asked if you would like to update any parameters.

4. Choose **Enter** to deploy the template.

Step 2: Monitoring periodic replication

You can monitor the status of the periodic replication workflow using the Amazon FSx Console, AWS CLI, and API. For more information on how to monitor periodic replication using the Amazon FSx Console, see Monitoring progress of on-demand data replication.

To use the AWS CLI or API to track the progress of your replication, call the describe-volumes CLI command or the DescribeVolumes API operation to view the `AdministrativeActions` array for the destination volume. The following example shows the response for an incremental copy on-demand data replication task.

```
"AdministrativeActions": [
   {
    "AdministrativeActionType": "VOLUME_UPDATE_WITH_SNAPSHOT",
    "ProgressPercent": 100,
    "RequestTime": 1699997847.438,
    "Status": "COMPLETED",
    "TargetVolumeValues": {
    "OpenZFSConfiguration": {
        "RecordSizeKiB": 128,
        "DataCompressionType": "ZSTD",
        "DeleteIntermediateSnaphots": true,
        "DeleteClonedVolumes": false,
        "DeleteIntermediateData": true,
        "SourceSnapshotARN": "arn:aws:fsx:us-east-1:609492434915:snapshot/
fsvol-0e1ab09de954a352f/fsvolsnap-01dda47dcbb24ddd0",
        "DestinationSnapshot": "fsvolsnap-0afef62088c7c9060"
        }
    },
    "TotalTransferBytes": 44144,
    "RemainingTransferBytes": 0
   },
```

Monitoring Amazon FSx for OpenZFS file systems

Monitoring is an important part of maintaining the reliability, availability, and performance of your FSx for OpenZFS file system and your other AWS solutions. Collecting monitoring data from all parts of your AWS solution allows you to more easily debug a multi-point failure if one occurs. You can monitor your FSx for OpenZFS file system, report when something is wrong, and take action automatically when appropriate using the following tools:

- **Amazon CloudWatch** – Monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track storage capacity or other metrics for your Amazon FSx instances and automatically launch new instances when needed.
- **AWS CloudTrail** – Captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

The following sections provide information on how to use both Amazon CloudWatch and AWS CloudTrail with your FSx for OpenZFS file systems.

Topics

- Monitoring with Amazon CloudWatch
- Logging FSx for OpenZFS API calls with AWS CloudTrail

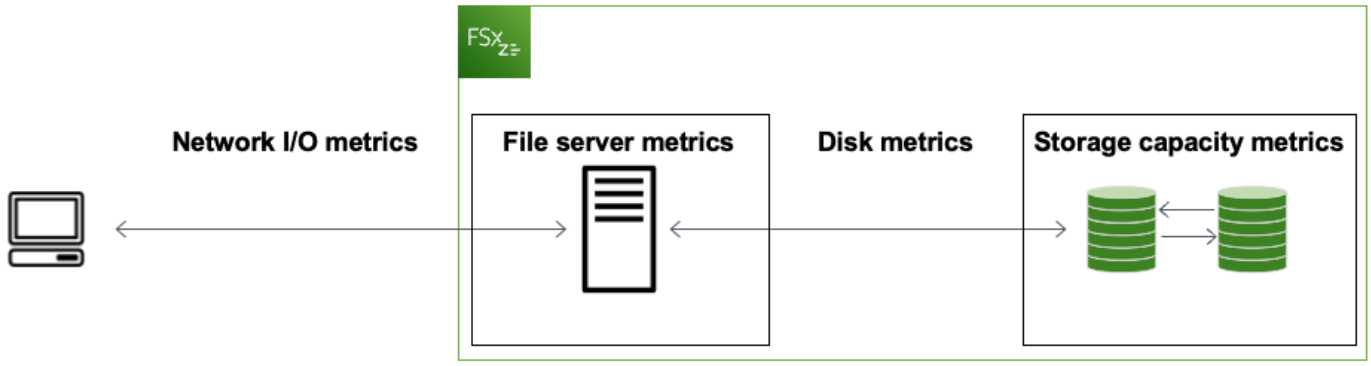
Monitoring with Amazon CloudWatch

You can monitor Amazon FSx using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months so that you can access historical information and gain a better perspective on how your application or service is performing. You can also set alarms that watch for certain thresholds and send notifications or take actions when those thresholds are met. For more information about CloudWatch, see What is Amazon CloudWatch? in the *Amazon CloudWatch User Guide*.

FSx for OpenZFS publishes CloudWatch metrics in the following domains:

- **Network I/O metrics** – Measure activity between clients that access 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 metrics** – Measure activity between the file server and the SSD storage.
- **Storage capacity metrics** – Measure storage usage.

The following diagram illustrates an FSx for OpenZFS file system, its components, and its metric domains.



By default, FSx for OpenZFS sends metric data to CloudWatch at 1-minute intervals. The following are exceptions to the default, and are sent at 5-minute intervals:

- `FileServerDiskThroughputBalance`
- `FileServerDiskIopsBalance`

> **Note**
>
> Metrics might not be published during file system maintenance for Single-AZ (non-HA) file systems, or during failover and failback between the primary and secondary file servers for Single-AZ (HA) and Multi-AZ (HA) file systems.

Topics

- Using Amazon FSx for OpenZFS CloudWatch metrics
- Accessing CloudWatch metrics

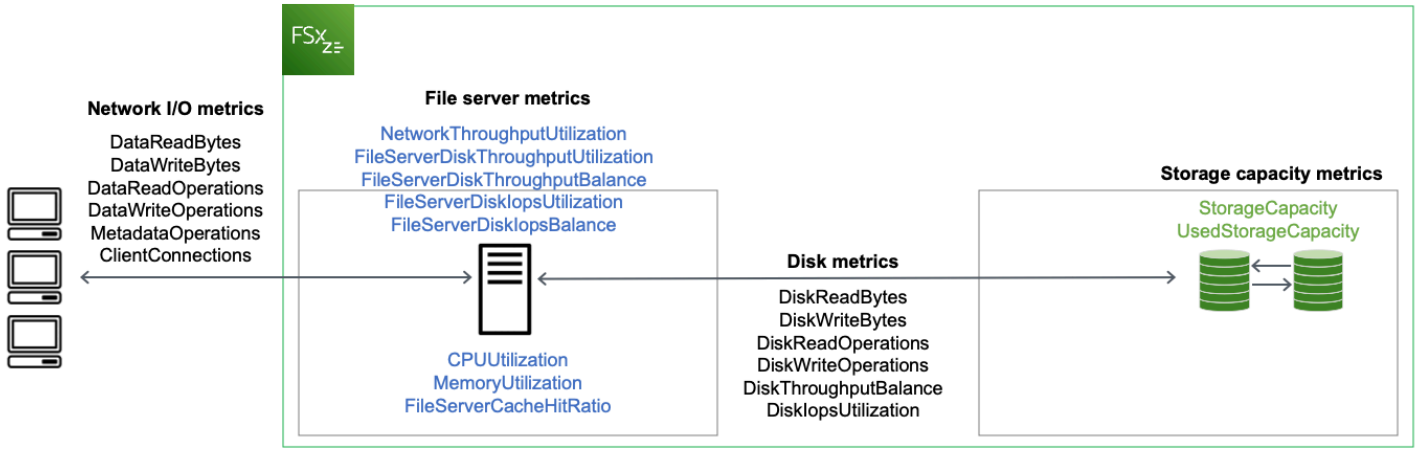
- Amazon FSx for OpenZFS metrics and dimensions
- Performance warnings and recommendations
- Creating CloudWatch alarms to monitor metrics

Using Amazon FSx for OpenZFS CloudWatch metrics

There are two primary architectural components of each Amazon FSx file system:

- The **file server** that serves data to clients that access the file system.
- The **storage volumes** that host the data in your file system.

FSx for OpenZFS reports metrics in CloudWatch that track 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 that are available for monitoring. The key property 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.



You can use the **Monitoring & performance** panel on your file system's dashboard in the Amazon FSx console to view the metrics that are described in the following table. For more information, see Accessing CloudWatch metrics.

Monitoring &performance panel	How do I...	Chart	Relevant metrics
Summary	...determine my file system's total throughput?	Total throughput	SUM(`DataReadBytes` + `DataWriteBytes`)/ Period (in seconds)
	...determine my file system's total IOPS?	Total IOPS	SUM(`DataReadOperations` + `DataWriteOperations` + `MetadataOperations`)/Period (in seconds)
	...determine the number of connections that are established between clients and the file server?	Client connections	`ClientConnections`
Storage	...determine how much primary storage is available?	Available primary storage capacity (bytes)	`StorageCapacity {SSD}` - `UsedStorageCapacity {SSD}`
	...determine the percentage of used primary storage for my file system?	Primary storage capacity utilization (percent)	`StorageCapacity {SSD}` * 100/`UsedStorageCapacity {SSD}`
File server performance	...determine the network throughput for clients that access the file system as a percentage of the file system's provisioned throughput?	Network throughput utilization	`NetworkThroughputUtilization`

Monitoring &performance panel	How do I...	Chart	Relevant metrics
	...determine the disk throughput between the file server and its storage volumes as a percentage of the provisioned limit, which is determined by throughput capacity?	Disk throughput utilization	`FileServerDiskThroughputUtilization`
	...determine the percentage of available burst credits for disk throughput between the file server and its storage volumes?	Disk throughput burst balance	`FileServerDiskThroughputBalance`
	...determine if the file system is approaching the file server's maximum SSD IOPS capacity? Do I need to increase throughput on my file system?	Disk IOPS utilization	`FileServerDiskIopsUtilization`
	...determine if the file system is approaching the file server's maximum SSD IOPS capacity? Do I need to increase my file system' throughput capacity?	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`
	...determine my workload's usage of the file system's in-memory (ARC), NVMe (L2ARC), and Intelligent-Tiering SSD read caches?	Cache hit ratio	`FileServerCacheHitRatio`

Monitoring &performance panel	How do I...	Chart	Relevant metrics
Disk performance	....determine if the file system is approaching the file server's current provisioned SSD IOPS capacity? Do I need to provision more IOPS for my file system?	Disk IOPS utilization (SSD)	`DiskIopsUtilization`
	...determine if the file system is approaching the file server's current provisioned SSD IOPS capacity? Do I need to provision more IOPS for my file system?	Compression ratio	`(UsedStorageCapacity /CompressionRatio ) - UsedStorageCapacity`

> **Note**
>
> We recommend that you provision throughput capacity so that the utilization of any performance-related dimension,—such as typical network, CPU, and memory utilization is less than 50%. This ensures that you have enough spare throughput capacity for unexpected spikes in your workload, as well as for background storage operations.

Accessing CloudWatch metrics

You can access Amazon FSx metrics for CloudWatch in the following ways:

- The Amazon FSx console.
- The CloudWatch console.
- The CloudWatch command line interface (CLI).
- The CloudWatch API.

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

Using the Amazon FSx console

To view metrics using the Amazon FSx console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation pane, choose **File systems**, then choose the file system that has the metrics that you want to view.
3. Choose **Actions > View details**.
4. On the **Summary** page, choose **Monitoring and performance** to see the metrics for your file system.

Using the CloudWatch console

To view metrics using the CloudWatch console

1. Open the CloudWatch console.
2. In the navigation pane, choose **Metrics**.
3. Select the **FSx** namespace.
4. (Optional) To view a metric, enter its name in the search field.
5. (Optional) To explore metrics, select the category that best matches your question. *File system metrics* and *Volume metrics* report summary-level metrics for individual file systems or volumes. *File system detailed metrics* and *Volume detailed metrics* report more granular metrics within a file system or volume. For example, storage capacity that's used by snapshots.

Using the AWS CLI

To access metrics from the AWS CLI

- Use the `list-metrics` command with the `--namespace "AWS/FSx"` namespace. For more information, see the AWS CLI Command Reference.

Using the CloudWatch API

To access metrics from the CloudWatch API

- Call `GetMetricStatistics`. For more information, see Amazon CloudWatch API Reference.

Amazon FSx for OpenZFS metrics and dimensions

Amazon FSx publishes the metrics described in the following tables in the AWS/FSx namespace in Amazon CloudWatch for all FSx for OpenZFS file systems:

Topics

- FSx for OpenZFS network I/O metrics
- FSx for OpenZFS file server metrics
- FSx for OpenZFS disk metrics
- FSx for OpenZFS storage capacity metrics
- FSx for OpenZFS dimensions

FSx for OpenZFS 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 that access the file system. Unit: Bytes Valid statistic: Sum
DataWriteBytes	The number of bytes for write operations for clients that access the file system. Unit: Bytes Valid statistic: Sum
DataReadOperations	The number of read operations for clients that access the file system. Unit: Count Valid statistic: Sum

Metric	Description
DataWriteOperations	The number of write operations for clients that access the file system. Unit: Count Valid statistic: Sum
ClientConnections	The number of active connections between clients and the file server. Unit: Count Valid statistics: Average, Maximum
MetadataOperations	The number of metadata operations for clients that access the file system. Unit: Count Valid statistic: Sum
NfsBadCalls	The number of calls rejected by the NFS server remote procedure call (RPC) mechanism. Unit: Count Valid statistic: Sum

FSx for OpenZFS file server metrics

The AWS/FSx namespace includes the following file server metrics.

Metric	Description
NetworkThroughputUtilization	The network throughput for clients that access the file system, as a percentage of the provisioned limit. For Multi-AZ file systems, this metric includes replication traffic. Unit: Percent

Metric	Description
	Valid statistics: `Average`, `Maximum`
`CPUUtilization`	The percentage utilization of your file server's CPU resources. Unit: Percent Valid statistics: `Average`, `Maximum`
`FileServerDiskThroughputUtilization`	The disk throughput between your file server and its storage volumes. Listed as a percentage of the provisioned limit, which is determined by throughput capacity. Unit: Percent Valid statistics: `Average`, `Maximum`
`FileServerDiskThroughputBalance`	The percentage of available burst credits for disk throughput between your file server and its storage volumes. Unit: Percent Valid statistics: `Average`, `Minimum`
`FileServerDiskIopsUtilization`	The disk IOPS between your file server and storage volumes, as a percentage of the provisioned limit, which is determined by throughput capacity. Unit: Percent Valid statistics: `Average`, `Maximum`
`FileServerDiskIopsBalance`	The percentage of available burst credits for disk IOPS between your file server and its storage volumes. Unit: Percent Valid statistics: `Average`, `Minimum`

Metric	Description
`MemoryUtilization`	The percentage of your file server's memory resources that are utilized. Unit: Percent Valid statistics: `Average`, `Maximum`
`FileServerCacheHitRatio`	The percentage of cache hits. For Single-AZ 2 (non-HA and HA) file systems, this metric reports the cache hit ratio for both the in-memory (ARC) and NVMe (L2ARC) caches. For Single-AZ 1 (non-HA and HA) file systems, this metric reports only the cache hit ratio for the ARC cache. Unit: Percent Valid statistics: `Average`, `Minimum`

FSx for OpenZFS disk metrics

The `AWS/FSx` namespace includes the following disk metrics.

Metric	Description
`DiskReadBytes`	The number of bytes for read operations that access storage volumes. Unit: Bytes Valid statistic: `Sum`
`DiskWriteBytes`	The number of bytes for write operations that access storage volumes. Unit: Bytes Valid statistic: `Sum`
`DiskReadOperations`	The number of read operations for the file server that accesses storage volumes.

Metric	Description
	Unit: Count Valid statistic: Sum
DiskWriteOperations	The number of write operations for the file server that accesses storage volumes. Unit: Count Valid statistic: Sum
DiskThroughputBalance	The percentage of available burst credits for disk throughput for the storage volumes. Unit: Percent Valid statistics: Average, Minimum
DiskIopsUtilization	The disk IOPS between your file server and storage volumes, as a percentage of the provisioned IOPS limit that has been determined by the storage volumes. Unit: Percent Valid statistics: Average, Maximum

FSx for OpenZFS storage capacity metrics

The AWS/FSx namespace includes the following storage capacity metrics.

Metric	Description
StorageCapacity	The total storage capacity, equal to the sum of used and available storage capacity. Unit: Bytes Valid statistics: Average, Minimum

Metric	Description
UsedStorageCapacity	The amount of storage that's used. Unit: Bytes Valid statistics: Average, Minimum
CompressionRatio	The ratio of compressed storage usage to uncompressed storage usage. Valid statistics: Average, Minimum

FSx for OpenZFS dimensions

FSx for OpenZFS provides additional dimensions to further refine the metrics listed in the previous table. FSx for OpenZFS metrics use the FSx namespace and provide metrics at the file system or volume granularity by using the FileSystemId or VolumeId.

Dimension	Description
FileSystemId	This dimension filters the metrics that you request to an individual file system.
VolumeId	This dimension filters the metrics that you request to an individual volume within a file system. This dimension must be used in combination with FileSystemId .
CacheType	This dimension filters the metrics that you request by the type of cache used, either ARC for in-memory caching or L2ARC for SSD and NVMe caching. This dimension must be used in combination with FileSystemId .
DataType	This dimension filters the metrics that you requested to a specific type of stored data. Metrics with DataType set to Snapshot

Dimension	Description
	report information about the snapshots within the volume. Metrics without a `DataType` dimension report aggregated information from the volume, which includes any child volumes and snapshots within the volume.

FSx for OpenZFS metrics use the `FSx` namespace and provide metrics for the dimensions that are listed in the table above.

Performance warnings and recommendations

FSx for OpenZFS displays a warning for CloudWatch metrics when one of these metrics approaches or crosses 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 on the Amazon FSx console. All active or recent Amazon FSx performance warnings and 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 where the metric graph is displayed.

You can create CloudWatch alarms for any of the Amazon FSx metrics. For more information, see Creating CloudWatch alarms to monitor metrics.

Use performance warnings to improve file system performance

Amazon FSx provides actionable recommendations that you can use to optimize your file system's performance. You can take the recommended action if you expect the issue 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 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

If there's a warning for this metric	Do this
File server > Disk IOPS – utilization	
File server > Disk throughput – utilization	
File server > Disk IOPS – burst balance	
File server > Disk throughput – burst balance	
File server > CPU utilization	
Storage capacity utilization	Increase storage capacity
Storage volume > Disk IOPS – utilization (SSD)	Increase SSD IOPS

For more information about file system performance, see Performance for Amazon FSx for OpenZFS.

Creating CloudWatch alarms to monitor metrics

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 that you specify and performs one or more actions based on the value of the metric relative to a given threshold over a specified period of time. The action is a notification that's sent to an Amazon SNS topic or Auto Scaling policy.

Alarms invoke actions for sustained state changes only. CloudWatch alarms don't invoke actions because they are in a particular state. The state must change and remain changed for a specified period of time. You can create an alarm on the Amazon FSx console or the CloudWatch console.

The following procedures describe how to create alarms for Amazon FSx using the console, AWS CLI, and API.

To set alarms using the Amazon FSx console

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the navigation pane, choose **File systems**, and then choose the file system that you want to create the alarm for.
3. Choose **Actions > View details**.

4. On the **Summary** page, choose **Monitoring and performance**.
5. Choose **Create CloudWatch alarm**. You are redirected to the CloudWatch console.
6. Choose **Select metrics**, and choose **Next**.
7. In the **Metrics** section, choose **FSX**.
8. Choose **File System Metrics**, choose the metric that you want to set the alarm for, and then choose **Select metric**.
9. In the **Conditions** section, choose the conditions for the alarm, and choose **Next**.

> **Note**
>
> Metrics might not be published during file system maintenance. To prevent unnecessary and misleading alarm condition changes and to configure your alarms so that they are resilient to missing data points, see Configuring how CloudWatch alarms treat missing data in the *Amazon CloudWatch User Guide*.

10. If you want CloudWatch to send you an email or SNS notification when the alarm state triggers the action, choose **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**.

> **Warning**
>
> If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, they do not receive a notification.

11. Fill in the **Name**, **Description**, and **Whenever** values for the metric, and choose **Next**.
12. On the **Preview and create** page, review the alarm and 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** to locate a metric. To narrow the results, you can search for your file system ID. Select the metric that you want to create an alarm for and choose **Next**.
4. Enter a **Name** and a **Description**, and choose a **Whenever** value for the metric.
5. If you want CloudWatch to send you an email when the alarm state is reached, choose **State is ALARM** for **Whenever this alarm**. For **Send notification to**, choose an existing SNS topic. If you select **Create topic**, you can set up the names and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms.

   > **Warning**
   >
   > 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. View the **Alarm Preview** and then choose **Create Alarm** or go back to make changes.

To set an alarm using the AWS CLI

- Call `put-metric-alarm`. For more information, see *AWS CLI Command Reference*.

To set an alarm using the CloudWatch

- Call `PutMetricAlarm`. For more information, see *Amazon CloudWatch API Reference*.

Logging FSx for OpenZFS API calls with AWS CloudTrail

Amazon FSx is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon FSx. CloudTrail captures all API calls for Amazon FSx as events. Captured calls include calls from the Amazon FSx console and from code calls to Amazon FSx API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon FSx. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by

CloudTrail, you can determine the request that was made to Amazon FSx. You can also determine the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Amazon FSx Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When API activity occurs in Amazon FSx, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for Amazon FSx, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following topics in the *AWS CloudTrail User Guide:*

- Creating a trail for your AWS account
- AWS service integrations with CloudTrail Logs
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

All Amazon FSx API calls are logged by CloudTrail. For example, calls to the `CreateFileSystem` and `TagResource` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element in the *AWS CloudTrail User Guide.*

Understanding Amazon FSx Log File Entries

A *trail* is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `TagResource` operation when a tag for a file system is created from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": “111122223333”,
        "arn": "arn:aws:sts::111122223333:root”,
        "accountId": “111122223333”,
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE”,
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T22:36:07Z"
            }
        }
    },
    "eventTime": "2018-11-14T22:36:07Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": “192.0.2.0”,
    "userAgent": “console.amazonaws.com”,
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789”
    },
    "responseElements": null,
    "requestID": “aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51”,
    "eventID": “bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p”,
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
```

```
    "recipientAccountId": “111122223333”
}
```

The following example shows a CloudTrail log entry that demonstrates the `UntagResource` action when a tag for a file system is deleted from the console.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": “Root”,
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T23:40:54Z"
            }
        }
    },
    "eventTime": "2018-11-14T23:40:54Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789"
    },
    "responseElements": null,
    "requestID": "aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51",
    "eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": "111122223333"
}
```

Migrating your existing file storage to Amazon FSx for OpenZFS

The following sections provide information on how to migrate your existing file storage to Amazon FSx for OpenZFS using AWS DataSync, `rsync` or `Robocopy`.

- **AWS DataSync** – An online data transfer service designed to simplify, automate, and accelerate copying large amounts of data to and from AWS storage services.
- **rsync** – Remote sync is an open source utility for efficiently transferring and synchronizing files commonly available on most Linux or other Unix-based operating systems.
- **Robocopy** – Robust File Copy is a command line directory and file replication command set for Microsoft Windows.

Before you begin using the procedures described in the following sections, be sure that the following prerequisites are met:

- You have created a destination FSx for OpenZFS file system. For more information, see Creating an Amazon FSx for OpenZFS file system.
- The source and destination file systems are connected in the same virtual private cloud (VPC). The source file system can be located on-premises or in another Amazon VPC, AWS account, or AWS Region, but it must be in a network peered with that of the destination file system using Amazon VPC Peering, Transit Gateway, AWS Direct Connect, or AWS VPN. For more information, see Access from a different VPC and What is VPC peering? in the *Amazon VPC Peering Guide*.

Topics

- Migrating files to Amazon FSx for OpenZFS using AWS DataSync
- Migrating files to Amazon FSx for OpenZFS using rsync
- Migrating files to Amazon FSx for OpenZFS using Robocopy
- Cutting over to your Amazon FSx for OpenZFS file system

Migrating files to Amazon FSx for OpenZFS using AWS DataSync

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

You can use DataSync to transfer files between two FSx for OpenZFS file systems, and also move data to a file system in a different AWS Region or AWS account. You can also use DataSync with FSx for OpenZFS 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 DataSync, a *location* is an endpoint for an FSx for OpenZFS file system. For information about specific transfer scenarios, see Working with locations in the *AWS DataSync User Guide*.

Prerequisites

To migrate data into your FSx for OpenZFS 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*.

Basic steps for migrating files using DataSync

Transferring files from a source to a destination using DataSync involves the following basic steps:

- Download and deploy an agent in your environment and activate it (not required if transferring between AWS services).
- Create a source and destination location.
- Create a task.
- Run the task to transfer files from the source to the destination.

For more information, see the following topics in the AWS DataSync User Guide:

- Data transfer between self-managed storage and AWS

- Creating a location for Amazon FSx for OpenZFS
- Deploy your agent as an Amazon EC2 instance

Migrating files to Amazon FSx for OpenZFS using rsync

With **rsync**, you can replicate data between any source and destination, but at least one must be locally accessible to the client instance.

Amazon EC2 instance

To migrate existing files to Amazon FSx from a Linux-based Amazon EC2 instance

The following procedure configures your FSx for OpenZFS destination volume as a local NFS mount on a Linux-based EC2 instance and uses the **rsync** command to synchronize data from your source file system or existing directory on your EC2 instance.

1. Launch a Linux-based Amazon EC2 instance or connect to an existing EC2 instance that contains your desired source data.
2. Mount your destination FSx for OpenZFS source volume; for more information, see Step 2: Mount your file system from an Amazon EC2 instance. The following step assumes that you have mounted your desired destination on your OpenZFS volume to `/fsx/destination_path` on your EC2 instance.
3. Run **rsync** from this EC2 instance to synchronize data from your source. If your source data is already on the EC2 instance, use the following command:

```
sudo rsync -avR /source_path /fsx/destination_path
```

> **Note**
>
> You can also run **rsync** with GNU parallel to maximize performance. The following instructions apply for EC2 Linux instances running Amazon Linux 2:.
>
> ```
> sudo amazon-linux-extras install epel
> sudo yum install nload sysstat parallel -y
> sudo time find -L /source_path -type f | parallel rsync -avR {} /fsx/
> destination_path
> ```

If your source is a directory on a remote host, use the following command:

```
sudo rsync -avR username@source_dns_or_ip:/source_path /fsx/destination_path
```

Use the following variant if you need to use .pem key-based authentication:

```
sudo rsync -avR -e "ssh -i key.pem" username@source_dns_or_ip:/source_path /fsx/
destination_path
```

On-premises

To migrate existing files to Amazon FSx from your on-premises Linux-based source

The following procedure configures your FSx for OpenZFS destination volume as a local NFS mount on a Linux-based EC2 instance. Then, you use **rsync** from your source to connect to this EC2 instance and synchronize files to the path where the destination Amazon FSx volume is mounted.

1. Launch a Linux-based Amazon EC2 instance and mount your destination FSx for OpenZFS source volume. For more information, see Step 2: Mount your file system from an Amazon EC2 instance. The following step assumes that you have mounted the desired destination on your OpenZFS volume to `/fsx/destination_path` on your EC2 instance.
2. From your on-premises Linux-based source, run **rsync** to connect to this EC2 instance and synchronize data from any locally accessible path. For example, `source_path` can refer to a locally accessible directory or a path on another shared file system.

```
sudo rsync -e "ssh -i key.pem" /source_path ec2-
user@ec2_dns_name.amazonaws.com:/fsx/destination_path
```

Migrating files to Amazon FSx for OpenZFS using Robocopy

Robocopy is designed to replicate data between two locations that are locally accessible on the same host. To use Robocopy to migrate data to your FSx for OpenZFS file system, you need to mount the source file system and the destination OpenZFS volume on the same Windows-based

EC2 client instance. The following procedure outlines the necessary steps to perform this migration using a new EC2 instance.

To migrate existing files to Amazon FSx

1. Launch a Windows Server 2016 Amazon EC2 instance in the same Amazon VPC as that of your Amazon FSx file system.
2. Connect to your Amazon EC2 instance. For more information, see Connect 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 using the Windows client instructions.
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 all the files on the source share to the target share. The example command 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
   - /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.

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

Note

If you are copying large files over a slow or unreliable connection, you can enable restartable mode by using the **/zb** option 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. Using the restartable mode can reduce data transfer speeds.

Cutting over to your Amazon FSx for OpenZFS file system

To cut over to your FSx for OpenZFS file system, do the following:

- Disconnect all clients that write to the source file system.
- Perform an **rsync** or **Robocopy** final file sync to ensure there is no data loss when cutting over.
- Connect all clients to your FSx for OpenZFS file system.

Now your FSx for OpenZFS file system is available with the data from the source file system and is available for clients to read and write to it. To make this data accessible to clients and applications, see Accessing your data.

Security in Amazon FSx for OpenZFS

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon FSx for OpenZFS, 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. The following topics show you how to configure Amazon FSx to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon FSx resources.

Topics

- Data encryption in Amazon FSx for OpenZFS
- Managing file system access with with Amazon VPC
- Identity and access management for Amazon FSx for OpenZFS
- Compliance validation for Amazon FSx for OpenZFS
- Using AWS PrivateLink to configure interface VPC endpoints
- Resilience in Amazon FSx for OpenZFS
- Infrastructure security in Amazon FSx for OpenZFS

Data encryption in Amazon FSx for OpenZFS

The AWS shared responsibility model applies to data protection in Amazon FSx for OpenZFS. 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 Amazon FSx 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.

Topics

- Encryption at rest
- Encryption in transit

Encryption at rest

Encryption of data at rest is automatically enabled when you create an Amazon FSx for OpenZFS file system through the AWS Management Console, the AWS CLI, or programmatically through the Amazon FSx API or one of the AWS SDKs. Your organization might require the encryption of all data that meets a specific classification or is associated with a particular application, workload, or environment. When you create a file system using the custom create flow, you can specify the KMS key with which to encrypt the data. If you create a file system using the Quick create flow, the data is encrypted using the AWS managed key. For more information about creating a file system encrypted at rest using the console, see Create Your Amazon FSx for OpenZFS File System and Creating an Amazon FSx for OpenZFS file system.

> **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.

For more information on how FSx for OpenZFS uses AWS KMS, see How Amazon FSx for OpenZFS uses AWS KMS.

How encryption at rest works

In an encrypted file system, data and metadata are automatically encrypted before being written to the file system. Similarly, as data and metadata are read, they are automatically decrypted before being presented to the application. These processes are handled transparently by Amazon FSx for OpenZFS, so you don't have to modify your applications.

Amazon FSx for OpenZFS uses industry-standard AES-256 encryption algorithm to encrypt file system data at rest. For more information, see Cryptography Basics in the *AWS Key Management Service Developer Guide*.

How Amazon FSx for OpenZFS uses AWS KMS

Amazon FSx for OpenZFS integrates with AWS Key Management Service (AWS KMS) for key management for encrypting data at rest. Amazon FSx uses AWS KMS keys to encrypt your file system in the following way:

- **Encrypting data at rest** – Amazon FSx for OpenZFS uses a KMS key, either the AWS managed key for Amazon FSx or a custom KMS key, to encrypt and decrypt file system data. All Amazon FSx for OpenZFS file systems are encrypted at rest with keys managed by the service. Data is encrypted using an XTS-AES-256 block cipher. The keys used to encrypt data at-rest are unique per file system and destroyed after the file system is deleted. 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 for Amazon FSx** – This is the default KMS key. You're not charged to create and store a KMS key, but there are usage charges. For more information, see AWS Key Management Service pricing.
  - **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 customer managed key at any time. For more information, see Rotating AWS KMS keys and Enabling and disabling keys in the *AWS Key Management Service Developer Guide*.

> **Important**
>
> Amazon FSx accepts only symmetric KMS keys. You can't use asymmetric KMS keys with Amazon FSx.

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 plain-text into cipher-text. 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 to select the KMS key. We recommend using this permission to provide the best user experience. This permission is included in the default key policy.

Encryption in transit

Amazon FSx for OpenZFS file systems automatically encrypt data in transit when they are accessed from Amazon EC2 instances that support encryption in transit. For more information about which EC2 instances support encryption in transit, see Encryption in transit in the *Amazon EC2 User Guide*. In-transit encryption of data is available in all AWS Regions where Amazon FSx for OpenZFS is available. For more information, see Availability by AWS Region.

Managing file system access with with Amazon VPC

You access your Amazon FSx for OpenZFS file systems and volumes using the file system's 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, or resources connected with the associated VPC by AWS Direct Connect or VPN, can access the data in your file system over the NFS protocol. For more information, see What is Amazon VPC? in the *Amazon VPC User Guide.*

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

Amazon VPC security groups

A security group acts as a virtual firewall for your FSx for OpenZFS file systems to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your file system, and outbound rules control the outgoing traffic from your file system. When you create a file system, you specify the VPC that it gets created in, and the default security group for that VPC is applied. You can add rules to each security group that allow traffic to or from its associated file systems and volumes. You can modify the rules for a security group at any time. New and modified rules are automatically applied to all resources that are associated with the security group. When Amazon FSx decides whether to allow traffic to reach a resource, it evaluates all of the rules from all of the security groups that are associated with the resource.

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.*

Creating a VPC security group

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. Remove any outbound rules on the security group. FSx for OpenZFS file systems do not initiate outbound connections in your VPC.
7. Add the following rules to the inbound ports of your security group.

Protocol	Ports	Role
TCP	111	Remote procedure call for NFS
UDP	111	Remote procedure call for NFS
TCP	2049	NFS server daemon
UDP	2049	NFS server daemon
TCP	20001 - 20003	NFS mount, status monitor, and lock daemon
UDP	20001 - 20003	NFS mount, status monitor, and lock daemon

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.

Identity and access management for Amazon FSx for OpenZFS

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 Amazon FSx 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 OpenZFS works with IAM
- Identity-based policy examples for Amazon FSx for OpenZFS
- AWS managed policies for Amazon FSx for OpenZFS
- Troubleshooting Amazon FSx for OpenZFS IAM issues
- Using tags to control access to Amazon FSx for OpenZFS resources
- Using service-linked roles for Amazon FSx for OpenZFS

Audience

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

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

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

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

Authenticating with identities

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

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

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

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

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

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

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

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using

credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

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

IAM users and groups

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

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

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

IAM roles

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

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity

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

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

requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

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

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

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

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

Identity-based policies

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

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

between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

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

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

Access control lists (ACLs)

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

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

Other policy types

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

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

service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.

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

Multiple policy types

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

How Amazon FSx for OpenZFS works with IAM

Before you use IAM to manage access to Amazon FSx, learn what IAM features are available to use with Amazon FSx.

IAM features you can use with Amazon FSx for OpenZFS

IAM feature	Amazon FSx support
Identity-based policies	Yes
Resource-based policies	No

IAM feature	Amazon FSx support
Policy actions	Yes
Policy resources	Yes
Policy condition keys	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 Amazon 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 Amazon 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. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Amazon FSx

To view examples of Amazon FSx identity-based policies, see Identity-based policy examples for Amazon FSx for OpenZFS.

Resource-based policies within Amazon FSx

Supports resource-based policies: No

Policy actions for Amazon 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. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

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

To see a list of Amazon FSx actions, see Actions defined by Amazon FSx in the *Service Authorization Reference*.

Policy actions in Amazon 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"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "fsx:Describe*"
```

To view examples of Amazon FSx identity-based policies, see Identity-based policy examples for Amazon FSx for OpenZFS.

Policy resources for Amazon 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. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

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

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

To see a list of Amazon FSx resource types and their ARNs, see Resources defined by Amazon FSx in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Amazon FSx.

To view examples of Amazon FSx identity-based policies, see Identity-based policy examples for Amazon FSx for OpenZFS.

Policy condition keys for Amazon 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 (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use

condition operators, such as equals or less than, to match the condition in the policy with values in the request.

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

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

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

To see a list of Amazon FSx condition keys, see Condition keys for Amazon FSx in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by Amazon FSx.

To view examples of Amazon FSx identity-based policies, see Identity-based policy examples for Amazon FSx for OpenZFS.

Access control lists (ACLs) in Amazon 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.

Attribute-based access control (ABAC) with Amazon FSx

Supports ABAC (tags in policies): Yes

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

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

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

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

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

For more information about tagging Amazon FSx resources, see Tag your Amazon FSx for OpenZFS resources.

To view an example identity-based policy for limiting access to a resource based on the tags on that resource, see Using tags to control access to your Amazon FSx resources.

Using Temporary credentials with Amazon FSx

Supports temporary credentials: Yes

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

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

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

Forward access sessions for Amazon FSx

Supports forward access sessions (FAS): Yes

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

Service roles for Amazon 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 Amazon FSx functionality. Edit service roles only when Amazon FSx provides guidance to do so.

Service-linked roles for Amazon 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 Amazon FSx service-linked roles, see Using service-linked roles for Amazon FSx for OpenZFS.

Identity-based policy examples for Amazon FSx for OpenZFS

Topics

- Policy best practices
- Using the Amazon FSx console
- Allow users to view their own permissions

Policy best practices

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

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

Using the Amazon FSx console

To access the Amazon FSx for OpenZFS console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Amazon FSx 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 Amazon FSx console, also attach the Amazon FSx `AmazonFSxConsoleFullAccess` or `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 OpenZFS

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 Amazon FSx for OpenZFS 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 Amazon FSx for OpenZFS.

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 AWS 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.

- `logs` – Allows principals to create log groups, log streams, and write events to log streams. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to CloudWatch Logs.
- `firehose` – Allows principals to write records to a Amazon 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.

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 AWS Directory Service directory.
- `ec2`
  - Allows principals to create tags on route tables, list network interfaces, route tables, security groups, subnets and the VPC associated with an Amazon FSx file system.
  - 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).

- `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 AWS Directory Service directory in the Amazon FSx Management Console.
- `ec2`
  - Allows principals to view network interfaces, security groups, subnets and the VPC associated with an Amazon FSx file system in the Amazon FSx Management Console.
  - 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.

- `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 for Amazon FSx for OpenZFS page.

Change	Description	Date
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
AmazonFSxConsoleFullAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:Descr`	February 07, 2025

Change	Description	Date
	`ibeNetworkInterfac es` that allows principals to view the elastic network interfaces associated with their file system.	
AmazonFSxServiceRolePolicy – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSe curityGroupsForVpc` 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:GetSe curityGroupsForVpc` that allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.	January 9, 2024
AmazonFSxConsoleRe adOnlyAccess – Update to an existing policy	Amazon FSx added new permission, `ec2:GetSe curityGroupsForVpc` 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 replicati on 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 replicati on 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 principals 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 principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can choose an existing CloudWatch Logs log group when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021

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 OpenZFS IAM issues

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

Topics

- I am not authorized to perform an action in Amazon FSx
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Amazon FSx resources

I am not authorized to perform an action in Amazon 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 Amazon FSx.

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 Amazon FSx. 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 Amazon 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 Amazon FSx supports these features, see How Amazon FSx for OpenZFS 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 to control access to Amazon FSx for OpenZFS resources

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 Amazon FSx API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based access control (ABAC). For more information, see What is ABAC for AWS in the *IAM User Guide*.

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `fsx:CreateFileSystem` or `fsx:CreateVolume`. 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 volumes and apply tags to them during creation in a specific AWS account.

```
{
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateFileSystem",
         "fsx:CreateVolume",
         "fsx:TagResource"
      ],
      "Resource": [
         "arn:aws:fsx:region:account-id:file-system/*",
         "arn:aws:fsx:region:account-id:file-system/*/volume/*"
      ]
    }
  ]
}
```

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 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 Tag your Amazon FSx for OpenZFS 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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource",
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }

    ]
}
```

Example policy – Create a file system with a specific tag from backups with a specific tag

This policy allows users to create file systems that are tagged with `Department=Finance` only from backups that are tagged with `Department=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:DeleteFileSystem"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Using service-linked roles for Amazon FSx for OpenZFS

Amazon FSx uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon FSx. Service-linked roles are predefined by Amazon FSx and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon FSx easier because you don't have to manually add the necessary permissions. Amazon FSx defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon FSx can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Amazon FSx resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Amazon FSx

Amazon FSx uses the service-linked role named **AWSServiceRoleForAmazonFSx** – which performs certain actions in your account, like creating Elastic Network Interfaces for your file systems in your VPC.

For updates to this policy, see AmazonFSxServiceRolePolicy

Permissions details

The AWSServiceRoleForAmazonFSx role permissions are defined by the AmazonFSxServiceRolePolicy AWS managed policy. The AWSServiceRoleForAmazonFSx has the following permissions:

> **Note**
>
> The AWSServiceRoleForAmazonFSx is used by all Amazon FSx file system types; some of the listed permissions are not applicable to FSx for OpenZFS.

- `ds` – Allows Amazon FSx to view, authorize, and unauthorize applications in your AWS Directory Service directory.
- `ec2` – Allows Amazon 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.
  - 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 Amazon FSx to publish metric data points to CloudWatch under the AWS/FSx namespace.
- `route53` – Allows Amazon FSx to associate an Amazon VPC with a private hosted zone.
- `logs` – Allows Amazon 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 Amazon 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.

```
{
    "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 Amazon FSx

You don't need to manually create a service-linked role. When you create a file system in the AWS Management Console, the IAM CLI, or the IAM API, Amazon FSx creates the service-linked role for you.

> **Important**
>
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. To learn more, see A New Role Appeared in My IAM Account.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a file system, Amazon FSx creates the service-linked role for you again.

Editing a service-linked role for Amazon FSx

Amazon FSx does not allow you to edit the AWSServiceRoleForAmazonFSx service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for Amazon FSx

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must delete all of your file systems and backups before you can manually delete the service-linked role.

> **Note**
>
> If the Amazon FSx service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the AWSServiceRoleForAmazonFSx service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported regions for Amazon FSx service-linked roles

Amazon FSx supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

Compliance validation for Amazon FSx for OpenZFS

Third-party auditors assess the security and compliance of Amazon FSx for OpenZFS as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others. Amazon FSx has been assessed to comply with PCI DSS, ISO 9001, 27001, 27017, and 27018, and SOC 1, 2, and 3, in addition to being HIPAA eligible.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).

- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Using AWS PrivateLink to configure interface VPC endpoints

You can improve the security posture of your VPC by configuring Amazon FSx to use an interface VPC endpoint. Interface VPC endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Amazon FSx APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon FSx APIs. Traffic between your VPC and Amazon FSx does not leave the AWS network.

Each interface VPC endpoint is represented by one or more elastic network interfaces in your subnets. A network interface provides a private IP address that serves as an entry point for traffic to the Amazon FSx API. Amazon FSx supports VPC endpoints configured with IPv4 and Dualstack (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 OpenZFS 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 control access to the Amazon FSx API, you can 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 on which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Resilience in Amazon FSx for OpenZFS

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

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

In addition to the AWS global infrastructure, Amazon FSx offers several features to help support your data resiliency and backup needs.

Backup and restore

Amazon FSx creates and saves automated backups of the volumes in your Amazon FSx for OpenZFS file system. Amazon FSx creates automated backups of your volumes during the backup window of your Amazon FSx for OpenZFS file system. Amazon FSx saves the automated backups of your volumes according to the backup retention period that you specify. You can also back up your volumes manually, by creating a user-initiated backup. You restore a volume backup at any time by creating a new volume with the backup specified as the source.

For more information, see TBD Protecting your data with backups.

Snapshots

Amazon FSx provides the ability to take snapshots of volumes within your file systems. A snapshot is a read-only image of your OpenZFS volume at a point in time, offering protection against accidental deletion or modification of files in your volumes by end users. For more information, see Protecting your data with snapshots.

Infrastructure security in Amazon FSx for OpenZFS

As a managed service, Amazon FSx for OpenZFS is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Amazon FSx through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Troubleshooting Amazon FSx for OpenZFS issues

Use the following sections to help troubleshoot file system, volume mounting, and storage related issues that you have with FSx for OpenZFS.

Topics

- Troubleshooting file system issues
- Troubleshooting volume mounting issues
- Troubleshooting storage issues

Troubleshooting file system issues

This section describes symptoms causes, and resolutions for when you are unable to create or access a file system.

Cannot create a file system because of misconfigured security group

Creating an FSx for OpenZFS file system fails with the following error message:

```
The file system cannot be created because the default security group in the
subnet provided or the provided security groups do not permit inbound NFSv4
traffic on TCP port 2049
```

Make sure that the VPC security group you are using for the creation operation is configured as described in Managing file system access with with Amazon VPC. You must set up the security group to allow inbound traffic on port 2049 from the security group itself or the full subnet CIDR. This is required to allow the file system hosts to communicate with each other.

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 public Elastic IP addresses (an IP address that is reachable from the public Internet), that gets attached to a file system's elastic network interface.

The file system's elastic network interface was modified or deleted

You must not modify or delete any of the file system's elastic network interfaces. Modifying or deleting a network interface can cause a permanent loss of connection between your virtual private cloud (VPC) and your file system. To resolve this issue, you must create a new file system, and do not modify or delete the Amazon FSx network interface. For more information, see Managing file system access with with Amazon VPC.

The compute instance's subnet doesn't use any of the route tables associated with your file system

FSx for OpenZFS creates an endpoint for accessing your file system in a VPC route table. We recommend that you configure your file system to use all of the VPC route tables that are associated with the subnets in which your clients are located. By default, Amazon FSx uses your VPC's main route table. You can optionally specify one or more route tables for Amazon FSx to use when you create your file system.

If your client is in a subnet that's not associated with any of your file system's route tables, you need to update your file system's route tables. For information about updating your file system's Amazon VPC route tables, see Updating an Amazon FSx for OpenZFS file system.

Troubleshooting volume mounting issues

This section describes symptoms, causes, and resolutions for when mounting a file system fails.

Mounting a volume fails right away

Using the `mount` command fails right away, as shown in the following example.

```
mount.nfs: access denied by server while mounting fs-02b568bbca05a9129.fsx.us-
east-1.amazonaws.com:/abc
```

This error can occur if you are using an invalid `volume_path` for the volume you are mounting in the mount command. The `volume_path` must match the fully-qualified path to the volume you want to mount. For example, to mount the root volume, specify the `volume_path` in the mount command using the following format: *`file-system-DNS-name`*`:/fsx`. A file system's DNS name is viewable in the Amazon FSx console on the file system detail page, in the **Network & security tab.**

You can view and copy the exact commands to mount any OpenZFS volume in the Amazon FSx console by choosing **Attach** on that volume's details page. For more information, see Step 2: Mount your file system from an Amazon EC2 instance.

Mounting a volume hangs and then fails with timeout error

The `mount` command hangs for a minute or two, and then fails with a timeout error similar to the following example:

```
mount.nfs: Connection timed out
```

This error can occur because the security groups for the Amazon EC2 instance or the file system aren't configured properly. Make sure that the security groups assigned to the file system have the inbound rules described in Managing file system access with with Amazon VPC.

Mounting a volume using a DNS name fails

A misconfigured Domain Name Service (DNS) name can cause volume mount failures with the following message:

```
Host filesystem_dns_name not found: 3(NXDOMAIN)
```

When this occurs, you will need to check your virtual private cloud (VPC) configuration. If you are using a custom VPC, make sure that DNS settings are enabled. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*.

Here are some considerations when using a DNS name in the mount command:

- Ensure that the Amazon EC2 instance is in the same VPC as your FSx for OpenZFS file system.
- Connect your Amazon EC2 instance inside a VPC configured to use the DNS server provided by AWS. For more information, see DHCP Options Sets in the *Amazon VPC User Guide*.
- Ensure that the VPC of the connecting Amazon EC2 instance has DNS host names enabled. For more information, see Updating DNS Support for Your VPC in the *Amazon VPC User Guide*.
- Ensure that DHCP option set has `AmazonProvidedDNS` configured as a domain name server. Amazon FSx uses Route53 private hosted zones for DNS. For more information, see What is Amazon Route 53 Resolver in the *Amazon Route 53 Resolver Developer Guide*.

Troubleshooting storage issues

This section describes symptoms, causes, and resolutions for storage issues on your file system.

Deleting files does not reduce used storage capacity

If deleting a file does not reduce used storage capacity, it's likely that the file's data is part of an OpenZFS snapshot that you created previously. Snapshots minimize the amount of storage capacity they consume by only storing each data block once, including blocks used in the most recent version of the file. This means that if you delete the file but the data blocks are still part of a non-deleted snapshot, those data blocks will be retained. To reduce your used storage capacity, consider deleting snapshots that you no longer need.

Service quotas on Amazon FSx for OpenZFS resources

FSx for OpenZFS has default AWS account and file system resource quotas. The following sections contain information on these quotas as well as how to request an increase from the default limit.

Topics

- Resource quotas for each AWS account
- Resource quotas for each file system
- Requesting a quota increase

Resource quotas for each AWS account

Following are the default quotas for Amazon FSx for OpenZFS for each AWS account, per AWS Region. For information on how to request an increase on a quota, see Requesting a quota increase.

Resource	Default	Description
OpenZFS file systems	100	The maximum number of Amazon FSx for OpenZFS file systems that you can create in this account.
OpenZFS SSD storage capacity	65,536 (262,144 in US East (N. Virginia), US East (Ohio), and US West (Oregon))	The maximum amount of SSD storage capacity (in GiB) that you can configure for all Amazon FSx for OpenZFS file systems in this account.
OpenZFS throughput capacity	10,240	The total amount of throughput capacity (in MBps) allowed for all Amazon FSx for OpenZFS file systems in this account.
OpenZFS disk IOPS	400,000	The total amount of disk IOPS allowed for all Amazon FSx

Resource	Default	Description
		for OpenZFS file systems in this account.
OpenZFS backups	10,000	The maximum number of user-initiated backups for all Amazon FSx for OpenZFS file systems that you can have in this account.
OpenZFS Intelligent-Tiering SSD read cache storage (GiB)	65,536 (262,144 in US East (N. Virginia), US East (Ohio), and US West (Oregon))	The maximum amount of SSD read cache storage that you configure across all file systems using the Intelligent-Tiering storage class.

Resource quotas for each file system

Following are the default quotas on FSx for OpenZFS resources for each file system in an AWS Region. For information on how to request an increase on a quota, see Requesting a quota increase.

Resource	Limit per file system
Minimum storage capacity	64 GiB
Maximum storage capacity	512 TiB [1]
Minimum throughput capacity	64 MBps
Maximum throughput capacity	10,240 MBps
Maximum number of volumes	100
Maximum number of user and group quotas per volume	100
Maximum number of snapshots	700
Maximum number of tags	50

Resource	Limit per file system
Maximum retention period for automated backups	90 days
Maximum retention period for user-initiated backups	no retention limit
Maximum storage capacity for file systems using the Intelligent-Tiering storage class	512 TiB

Note

[1] The maximum storage capacity for Single-AZ 2 (non-HA and HA) file systems is 512 TiB. The maximum storage capacity for Single-AZ 1 (non-HA and HA) and Multi-AZ (HA) file systems depends on the file system's provisioned throughput capacity, deployment type, and AWS Region. For more information, see the following tables.

Maximum storage capacity of Multi-AZ (HA) file systems

Provisioned throughput capacity (MBps)	Maximum Storage Capacity (TiB)[2]	Maximum Storage Capacity (TiB)[3]	Maximum Storage Capacity (TiB)[4]
160	32	24	96
320	64	48	128
640	128	96	128
1,280	256	128	128
2,560	512	128	128
3,840	512	128	128
5,120	512	128	128
7,680	512	–	–

Provisioned throughput capacity (MBps)	Maximum Storage Capacity (TiB)[2]	Maximum Storage Capacity (TiB)[3]	Maximum Storage Capacity (TiB)[4]
10,240	512	–	–

Note

[2] These storage capacity limits apply to the following AWS Regions: US East (N. Virginia), US East (Ohio), US West (Oregon), Europe (Ireland), Europe (Frankfurt), Asia Pacific (Tokyo), Asia Pacific (Sydney), Asia Pacific (Singapore).
[3] These storage capacity limits apply to the following AWS Regions: Asia Pacific (Osaka), Asia Pacific (Seoul), Asia Pacific (Mumbai), Asia Pacific (Hong Kong), Canada (Central),Europe (Milan), Europe (Paris), Europe (London), Europe (Stockholm) Israel (Tel Aviv), Middle East (Bahrain), South America (São Paulo), Europe (Zurich), Europe (Spain).
[4] These storage capacity limits apply to the following AWS Regions: US West (N. California), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Middle East (UAE), AWS GovCloud (US-East), AWS GovCloud (US-West).

Maximum storage capacity of Single-AZ 1 (non-HA and HA) file systems

Provisioned throughput capacity (MBps)	Maximum Storage Capacity (TiB) [5]	Maximum Storage Capacity (TiB) [6]
64	512	128
128	512	128
256	512	128
512	512	128
1,024	512	128
2,048	512	128
3,072	512	128

Provisioned throughput capacity (MBps)	Maximum Storage Capacity (TiB) [5]	Maximum Storage Capacity (TiB) [6]
4,096	512	128

Note

[5] These storage capacity limits apply to the following AWS Regions: US East (N. Virginia), US East (Ohio), US West (Oregon), Europe (Stockholm), Europe (London), Europe (Ireland), Europe (Frankfurt), Asia Pacific (Tokyo), Asia Pacific (Seoul), Asia Pacific (Mumbai), Asia Pacific (Sydney), Asia Pacific (Singapore), Asia Pacific (Hong Kong), Canada (Central).
[6] These storage capacity limits apply to the following AWS Regions: US West (N. California), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Asia Pacific (Osaka), Europe (Milan), Europe (Paris), Israel (Tel Aviv), Middle East (UAE), Middle East (Bahrain), South America (São Paulo), AWS GovCloud (US-East), AWS GovCloud (US-West), Europe (Zurich), Europe (Spain).

Requesting a quota increase

You can request an increase for all AWS account quotas. You can also request an increase on the following file system quotas:

- Maximum number of volumes
- Maximum number of user and group quotas per volume
- Maximum number of snapshots
- Maximum storage capacity for file systems using the Intelligent-Tiering storage class

To request a quota increase

1. Open the AWS Support page, sign in if necessary, and then choose **Create case**.
2. For **Create case**, choose **Account and billing support**.
3. In the **Case details** panel make the following entries:
   - For **Type** choose **Account**.

- For **Category** choose **Other Account Issues**.
- For **Subject** enter **`Amazon FSx for OpenZFS service limit increase request`**.
- Provide a detailed **Description** of your request, including:
  - The FSx quota that you want increased, and the value you want it increased to, if known.
  - The reason why you are seeking the quota increase.
  - The file system ID and region for each file system you are requesting an increase for.

4. Provide your preferred **Contact options** and choose **Submit**.

Document history for Amazon FSx for OpenZFS

- **API version:** 2018-03-01
- **Latest documentation update:** April 28, 2025

The following table describes important changes to the *Amazon FSx for OpenZFS User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

Change	Description	Date
FSx for OpenZFS now supports lower throughput capacity options for the Intelligent-Tiering storage class.	The Intelligent-Tiering storage class now supports provisioning 160 MBps of throughput capacity or greater. Intelligent-Tiering provides fully elastic storage with an optional SSD cache for low-latency access to frequently accessed data. With Intelligent-Tiering, you are billed for the data you store, depending on the size of your dataset, and do not need to specify a file system size at creation. For more information, see Intelligent-Tiering.	April 28, 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	February 25, 2025

	the AmazonFSxConsoleReadOnlyAccess policy.	
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 OpenZFS 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 the AmazonFSxConsoleFullAccess policy.	February 7, 2025
Additional AWS Region support added.	FSx for OpenZFS is now available in Asia Pacific (Malaysia). For more information, see Availability by AWS Region.	January 2, 2025

FSx for OpenZFS now offers the Intelligent-Tiering storage class.	You can now create FSx for OpenZFS file systems with the Intelligent-Tiering storage class. Intelligent-Tiering provides fully elastic storage with an optional SSD cache for low-latency access to frequently accessed data. With Intelligent-Tiering, you are billed for the data you store, depending on the size of your dataset, and do not need to specify a file system size at creation. For more information, see Intelligent-Tiering.	December 1, 2024
FSx for OpenZFS now offers the Single-AZ (HA) deployment type.	You can now create FSx for OpenZFS file systems using the Single-AZ (HA) deployment type, which offers high availability within a single availability zone. For more information, see Availability and durability.	July 9, 2024
Additional AWS Region support added.	FSx for OpenZFS is now available in Canada West (Calgary). For more informati on, see Deployment type availability.	May 3, 2024

Additional AWS Region support added.	FSx for OpenZFS is now available in China (Beijing) , China (Ningxia), AWS GovCloud (US-West), AWS GovCloud (US-East), Europe (Spain), and Europe (Zurich). For more information, see Deployment type availability.	March 11, 2024
Amazon FSx for OpenZFS now supports up to 400,000 IOPS for Single-AZ 2 file systems	Amazon FSx for OpenZFS now supports up to 400,000 IOPS for Single-AZ 2 file systems. For more information, see Amazon FSx for OpenZFS Performance.	January 17, 2024
Amazon FSx updated the AmazonFSxFullAccess, AmazonFSxConsoleFullAccess, AmazonFSx ReadOnlyAccess, AmazonFSx ConsoleReadOnlyAccess, and AmazonFSxServiceRolePolicy AWS managed policies	Amazon FSx updated the AmazonFSxFullAccess, AmazonFSxConsoleFullAccess, AmazonFSx ReadOnlyAccess, AmazonFSx ConsoleReadOnlyAccess, 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 for OpenZFS now offers cross-region and cross-account data replication	You can now use on-demand data replication with Amazon FSx for OpenZFS to securely transfer data across file systems in any AWS Account and AWS Region. For more information, see On-demand data replication.	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 for OpenZFS now offers on-demand data replication	You can now use on-demand data replication with Amazon FSx for OpenZFS to securely transfer data across file systems in the same AWS Account and AWS Region. For more information, see On-demand data replication.	November 26, 2023

Additional AWS Region support added.	FSx for OpenZFS is now available in Canada (Central) and US West (N. California). For more information, see Deployment type availability.	November 16, 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` , `fsx:CopySnapshotAndUpdateVolume` permission. For more information, see Amazon FSx updates to AWS managed policies.	November 14, 2023
Additional AWS Region support added.	FSx for OpenZFS is now available in Middle East (Bahrain), Asia Pacific (Osaka), Europe (Milan), Europe (Paris), South America (São Paulo) Region, Israel (Tel Aviv), Africa (Cape Town), Asia Pacific (Hyderabad), Asia Pacific (Jakarta), and Middle East (UAE). For more information, see Deployment type availability.	November 9, 2023

Update to AmazonFSx FullAccess policy.	Amazon FSx added new permissions to allow Amazon FSx to manage network configurations for FSx for OpenZFS Multi-AZ file systems. For more informati on, see AWS managed policies for Amazon FSx.	August 9, 2023
FSx for OpenZFS now offers the Multi-AZ deployment type.	You can now use Multi-AZ deployment for FSx for OpenZFS to create file systems with high availabil ity that span across multiple availability zones. For more information, see Availability and durability.	August 9, 2023
Amazon FSx updated the AmazonFSxServiceRolePolicy AWS managed policy	Amazon FSx updated the `cloudwatch:PutMetr icData` permission in the AmazonFSxServiceRolePolicy. For more information, see Amazon FSx updates to AWS managed policies.	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 AmazonFSx FullAccess 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 using the Amazon Elastic Kubernetes Service container storage interface (CSI) driver for FSx for OpenZFS.	The Amazon FSx for OpenZFS CSI Driver provides a CSI interface that allows Amazon EKS clusters to manage the life cycle of Amazon FSx for OpenZFS file systems and volumes. For more information, see Using FSx for OpenZFS with AWS container services and Amazon FSx for OpenZFS CSI Driver on GitHub.	June 30, 2023
Support added for a new generation Single-AZ 2 deployment type.	You can now specify file systems with 10 GBps maximum throughput, 350,000 maximum IOPS, and data access from a new NVMe L2ARC read cache. For more information, see Amazon FSx for OpenZFS performance.	November 28, 2022

Support added for restoring backups to file system's with increased storage capacity and throughput capacity	You can now specify a storage capacity equal to or greater than the original file system's storage capacity when restoring an FSx for OpenZFS backup to a new file system. For more information, see Restoring backups.	September 29, 2022
Additional AWS Region support added	FSx for OpenZFS is now available in the Asia Pacific (Hong Kong), Asia Pacific (Seoul), Asia Pacific (Mumbai), and Europe (Stockholm) AWS Regions.	September 29, 2022
Support added for SSD storage capacity and provisioned IOPS scaling	You can now increase the SSD storage capacity and provisioned IOPS for existing FSx for OpenZFS file systems as your storage and IOPS requirements evolve. For more information, see Managing SSD storage capacity and provisioned IOPS.	May 31, 2022
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 the native Amazon FSx backups. For more informati on, see Using AWS Backup with Amazon FSx.	May 18, 2022

AWS Region support added	FSx for OpenZFS is now available in the Europe (London), Asia Pacific (Singapore), and Asia Pacific (Sydney) AWS Regions.	April 19, 2022
Support added for using AWS DataSync to migrate files to your FSx for OpenZFS file systems.	You can now use AWS DataSync to migrate files from existing file systems to FSx for OpenZFS file systems. For more information, see How to migrate existing files to FSx for OpenZFS using AWS DataSync.	April 5, 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 to unset, or turn off volume storage capacity quotas and reservations.	You can now unset or turn off a volume's storage capacity quota and reservation settings . For more information, see Volume properties.	February 28, 2022
Support added for customizing a volume's ZFS record size.	You can now set the ZFS record size on a per volume basis to improve performance for specific workflows. For more information, see Volume properties.	February 28, 2022

Support added for LZ4 data compression.	You can now use LZ4 data compression in addition to Z-Standard data compressi on on a per volume basis. For more information, see Volume properties.	February 28, 2022
Support added for full-copy volumes	You can use a snapshot to create an FSx for OpenZFS full-copy volume. A full-copy volume is initialized with the same data as its source snapshot, but is a fully independent writable copy. For more information, see Managing FSx for OpenZFS volumes.	February 2, 2022
FSx for OpenZFS is now generally available	FSx for OpenZFS for OpenZFS is a fully managed file storage service that makes it easy to move data residing in on-premises ZFS or other NFS file servers to AWS without changing your application code or how you manage data. It offers highly reliable, scalable, performant, and feature-rich file storage built on the open-source OpenZFS file system.	November 30, 2021