

Lustre User Guide

# **FSx for Lustre**



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

## **FSx for Lustre: Lustre 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.

This documentation is a draft for private preview for regions in the AWS European Sovereign Cloud. Documentation content will continue to evolve. Published: December 31, 2025.

# **Table of Contents**

| What is Amazon FSx for Lustre?                           | 1  |
|--|----|
| Multiple deployment options and storage classes          | 2  |
| FSx for Lustre and data repositories                     | 2  |
| FSx for Lustre S3 data repository integration            | 2  |
| FSx for Lustre and on-premises data repositories         | 3  |
| Accessing file systems                                   | 3  |
| Integrations with AWS services                           | 4  |
| Security and compliance                                  | 4  |
| Assumptions  | 5  |
| Pricing for Amazon FSx for Lustre                        | 5  |
| Amazon FSx for Lustre forums                             | 5  |
| Are you a first-time user of Amazon FSx for Lustre?      | 5  |
| Setting up   | 7  |
| Sign up for Amazon Web Services                          | 7  |
| Sign up for an AWS account                               | 7  |
| Create a user with administrative access                 | 8  |
| Adding permissions to use data repositories in Amazon S3 | 9  |
| How FSx for Lustre checks access to S3 buckets           | 10 |
| Next step  | 12 |
| Getting started  | 13 |
| Prerequisites  | 13 |
| Step 1: Create your FSx for Lustre file system           | 14 |
| Install the Lustre client                                | 18 |
| Step 3: Mount the file system                            | 19 |
| Step 4: Run your workflow                                | 21 |
| Step 5: Clean up resources                               | 21 |
| Deployment and storage class options                     | 23 |
| Persistent file systems                                  | 23 |
| Persistent 2 deployment type                             | 24 |
| Persistent 1 deployment type                             | 24 |
| Scratch file systems                                     | 24 |
| IP addresses   | 25 |
| FSx for Lustre storage classes                           | 26 |
| How the Intelligent-Tiering storage class tiers data     | 27 |

|    | Deployment type availability                         | 28    |
|----|--|-------|
| Us | ing data repositories                                | 31    |
|    | Overview of data repositories                        | 31    |
|    | Region and account support for linked S3 buckets     | 33    |
|    | POSIX metadata support                               | 34    |
|    | Exporting hard links                                 | 35    |
|    | Attaching POSIX permissions to an S3 bucket          | 36    |
|    | Linking your file system to an S3 bucket             | 39    |
|    | Creating a link to an S3 bucket                      | 41    |
|    | Updating data repository association settings        | 44    |
|    | Deleting an association to an S3 bucket              | 46    |
|    | Viewing data repository association details          | 47    |
|    | Data repository association lifecycle state          | 48    |
|    | Working with server-side encrypted Amazon S3 buckets | 48    |
|    | Importing changes from your data repository          | 51    |
|    | Automatically import updates from your S3 bucket     | 53    |
|    | Using data repository tasks to import changes        | 57    |
|    | Preloading files into your file system               | 59    |
|    | Exporting changes to the data repository             | 62    |
|    | Automatically export updates to your S3 bucket       | 63    |
|    | Using data repository tasks to export changes        | 66    |
|    | Exporting files using HSM commands                   | 69    |
|    | Data repository tasks                                | 70    |
|    | Types of data repository tasks                       | 70    |
|    | Task status and details                              | 71    |
|    | Using data repository tasks                          | 72    |
|    | Working with task completion reports                 | 79    |
|    | Troubleshooting task failures                        | 81    |
|    | Releasing files                                      | 85    |
|    | Using data repository tasks to release files         | 87    |
|    | Using Amazon FSx with your on-premises data          | 89    |
|    | Data repository event logs                           | 90    |
|    | Import events  | 90    |
|    | Export events  | 97    |
|    | HSM restore events                                   | . 104 |
|    | Working with older deployment types                  | . 105 |

| Link your file system to an Amazon S3 bucket                  | 106 |
|---|-----|
| Automatically import updates from your S3 bucket              | 114 |
| Performance   | 119 |
| Overview  | 119 |
| How FSx for Lustre file systems work                          | 119 |
| File system metadata performance                              | 120 |
| Throughput to individual client instances                     | 122 |
| File system storage layout                                    | 123 |
| Striping data in your file system                             | 123 |
| Modifying your striping configuration                         | 124 |
| Progressive file layouts                                      | 126 |
| Monitoring performance and usage                              | 127 |
| SSD and HDD storage classes                                   | 127 |
| Example: Aggregate baseline and burst throughput              | 132 |
| Intelligent-Tiering storage class                             | 132 |
| File system performance for Intelligent-Tiering               | 133 |
| Performance tips  | 135 |
| Intelligent-Tiering performance tips                          | 137 |
| Accessing file systems  | 138 |
| Lustre file system and client kernel compatibility            | 138 |
| Installing the Lustre client                                  | 143 |
| Amazon Linux  | 143 |
| CentOS, Rocky Linux, and Red Hat                              | 145 |
| Ubuntu  | 155 |
| SUSE Linux  | 157 |
| Mount from Amazon EC2   | 159 |
| Configure EFA clients   | 161 |
| Step 1: Install required drivers                              | 161 |
| Step 2: Configure EFA for the Lustre client                   | 162 |
| Step 3: EFA interfaces  | 163 |
| Mounting from Amazon ECS                                      | 165 |
| Mounting from an Amazon EC2 instance hosting Amazon ECS tasks | 166 |
| Mounting from a Docker container                              | 167 |
| Mounting from on-premises or another VPC                      | 168 |
| Mounting Amazon FSx automatically                             | 170 |
| Automount using /etc/fstab                                    | 170 |

|    | Mounting specific filesets                                     | 174   |
|----|--|-------|
|    | Unmounting file systems  | 175   |
|    | Using EC2 Spot Instances                                       | 176   |
|    | Handling Amazon EC2 Spot Instance interruptions                | 176   |
| Ac | dministering file systems                                      | 179   |
|    | EFA-enabled file systems                                       | 179   |
|    | Considerations when using EFA-enabled file systems             | 180   |
|    | Prerequisites for using EFA-enabled file systems               | 180   |
|    | Creating an EFA-enabled file system                            | 182   |
|    | Storage quotas   | 182   |
|    | Quota enforcement  | 183   |
|    | Types of quotas  | 183   |
|    | Quota limits and grace periods                                 | 184   |
|    | Setting and viewing quotas                                     | 184   |
|    | Quotas and Amazon S3 linked buckets                            | 188   |
|    | Quotas and restoring backups                                   | 189   |
|    | Storage capacity   | 189   |
|    | Considerations when increasing storage capacity                | 190   |
|    | When to increase storage capacity                              | 191   |
|    | How concurrent storage scaling and backup requests are handled | 191   |
|    | Increasing storage capacity                                    | 192   |
|    | Monitoring storage capacity increases                          | 193   |
|    | SSD read caches  | 196   |
|    | Considerations when updating SSD read cache                    | 199   |
|    | Updating a provisioned SSD read cache                          | . 199 |
|    | Monitoring SSD read cache updates                              | 201   |
|    | Manage metadata performance                                    | 202   |
|    | Lustre metadata performance configuration                      | 203   |
|    | Considerations when increasing metadata performance            | 204   |
|    | When to increase metadata performance                          | 205   |
|    | Increasing metadata performance                                | 205   |
|    | Changing the metadata configuration mode                       | 207   |
|    | Monitoring metadata configuration updates                      | 208   |
|    | Throughput capacity  | 210   |
|    | Considerations when updating throughput capacity               | 211   |
|    | When to modify throughout capacity                             | 212   |

| Modifying throughput capacity               | 212 |
|---|-----|
| Monitoring throughput capacity changes      | 215 |
| Data compression                            | 217 |
| Managing data compression                   | 218 |
| Compressing previously written files        | 221 |
| Viewing file sizes                          | 221 |
| Using CloudWatch metrics                    | 221 |
| Root squash                                 | 222 |
| How root squash works                       | 222 |
| Managing root squash                        | 223 |
| File system status                          | 228 |
| Tag your resources                          | 228 |
| Tag basics                                  | 229 |
| Tagging your resources                      | 229 |
| Tag restrictions                            | 230 |
| Permissions and tag                         | 231 |
| Maintenance                                 | 231 |
| Lustre versions                             | 232 |
| Best practices for Lustre version upgrades  | 233 |
| Performing the upgrade                      | 233 |
| Deleting a file system                      | 235 |
| Backups                                     | 236 |
| Backup support in FSx for Lustre            | 237 |
| Working with automatic daily backups        | 237 |
| Working with user-initiated backups         | 238 |
| Creating user-initiated backups             | 238 |
| Using AWS Backup with Amazon FSx            | 239 |
| Copying backups                             | 240 |
| Backup copy limitations                     | 241 |
| Permissions for cross-Region backup copies  | 241 |
| Full and incremental copies                 | 242 |
| Copying backups within the same AWS account | 242 |
| Restoring backups                           | 243 |
| Deleting backups                            | 244 |
| Monitoring file systems                     | 245 |
| Monitoring with CloudWatch                  | 245 |

| Using CloudWatch metrics                             | 247   |
|--|-------|
| Accessing CloudWatch metrics                         | 251   |
| Metrics and dimensions                               | 252   |
| Performance warnings and recommendations             | . 272 |
| Creating CloudWatch alarms                           | . 275 |
| Logging with CloudWatch Logs                         | . 277 |
| Logging overview                                     | 278   |
| Log destinations                                     | 278   |
| Managing logging                                     | . 279 |
| Viewing logs   | 281   |
| Logging with AWS CloudTrail                          | 281   |
| Amazon FSx for Lustre information in CloudTrail      | . 282 |
| Understanding Amazon FSx for Lustre log file entries | . 283 |
| Migrating to FSx for Lustre                          | 286   |
| Migrating files with AWS DataSync                    | 286   |
| Prerequisites  | 286   |
| DataSync migration basic steps                       | 287   |
| Security   | 288   |
| Data protection                                      | . 289 |
| Data encryption                                      | . 290 |
| Internetwork traffic privacy                         | . 293 |
| Identity and access management                       | . 294 |
| Audience   | . 294 |
| Authenticating with identities                       | 294   |
| Managing access using policies                       | . 296 |
| FSx for Lustre and IAM                               | 297   |
| Identity-based policy examples                       | . 302 |
| AWS managed policies                                 | . 305 |
| Troubleshooting                                      | . 320 |
| Using tags with Amazon FSx                           | . 322 |
| Using service-linked roles                           | . 329 |
| File system access control with Amazon VPC           | . 335 |
| Amazon VPC Security Groups                           | . 335 |
| Lustre client VPC security group rules               | . 340 |
| Amazon VPC network ACLs                              | . 342 |
| Compliance Validation                                | . 342 |

| Interface VPC endpoints  | . 343 |
|--|-------|
| Considerations for Amazon FSx interface VPC endpoints                                    | 343   |
| Creating an interface VPC endpoint for Amazon FSx API                                    | . 344 |
| Creating a VPC endpoint policy for Amazon FSx  | 344   |
| Service quotas   | 346   |
| Quotas that you can increase   | 346   |
| Resource quotas for each file system   | . 348 |
| Additional considerations  | . 349 |
| Troubleshooting  | . 350 |
| Creating a file system fails   | . 350 |
| Cannot create an EFA-enabled file system because of misconfigured security group         | . 350 |
| Cannot create a file system because of misconfigured security group                      | 351   |
| Cannot create a file system because of insufficient capacity errors                      | 351   |
| Cannot create a file system that is linked to an S3 bucket                               | . 351 |
| File system mount fails  | . 352 |
| File system mount fails right away   | 352   |
| File system mount hangs and then fails with timeout error                                | . 352 |
| Automatic mounting fails and the instance is unresponsive                                | . 353 |
| File system mount fails during system boot   | . 353 |
| File system mount using DNS name fails   |       |
| You cannot access your file system   | . 355 |
| The Elastic IP address attached to the file system elastic network interface was deleted | . 355 |
| The file system elastic network interface was modified or deleted                        | . 355 |
| Creating a DRA fails   | . 355 |
| Renaming directories takes a long time   |       |
| Misconfigured linked S3 bucket   | . 357 |
| Storage issues   | . 358 |
| Write error due to no space on storage target  | . 358 |
| Unbalanced storage on OSTs   | . 359 |
| CSI driver issues  | . 362 |
| Additional information   | 363   |
| Setting up a custom backup schedule  | . 363 |
| Architecture overview  | . 363 |
| CloudFormation template  | . 364 |
| Automated deployment   | . 364 |
| Additional options   | . 366 |

| FSx for Lustre | Lustre User Guide |
|----------------|-------------------|
|                |                   |
|                |                   |

Document history ...... 368

## What is Amazon FSx for Lustre?

FSx for Lustre makes it easy and cost-effective to launch and run the popular, high-performance Lustre file system. You use Lustre for workloads where speed matters, such as machine learning, high performance computing (HPC), video processing, and financial modeling.

The Lustre file system is designed for applications that require fast storage—where you want your storage to keep up with your compute. Lustre was built to solve the problem of quickly and cheaply processing the world's ever-growing datasets. It's a widely used file system designed for the fastest computers in the world. It provides sub-millisecond latencies, up to multiple TBps of throughput and up to millions of IOPS. For more information on Lustre, see the Lustre website.

As a fully managed service, Amazon FSx makes it easier for you to use Lustre for workloads where storage speed matters. FSx for Lustre eliminates the traditional complexity of setting up and managing Lustre file systems, enabling you to spin up and run a battle-tested high-performance file system in minutes. It also provides multiple deployment options and storage classes so you can optimize cost for your needs.

FSx for Lustre is POSIX-compliant, so you can use your current Linux-based applications without having to make any changes. FSx for Lustre provides a native file system interface and works as any file system does with your Linux operating system. It also provides read-after-write consistency and supports file locking.

#### **Topics**

- Multiple deployment options and storage classes
- FSx for Lustre and data repositories
- Accessing FSx for Lustre file systems
- Integrations with AWS services
- Security and compliance
- Assumptions
- Pricing for Amazon FSx for Lustre
- Amazon FSx for Lustre forums
- Are you a first-time user of Amazon FSx for Lustre?

# Multiple deployment options and storage classes

Amazon FSx for Lustre offers a choice of *scratch* and *persistent* file systems to accommodate different data processing needs. Scratch file systems are ideal for temporary storage and shorter-term processing of data. Data is not replicated and does not persist if a file server fails. Persistent file systems are ideal for longer-term storage and throughput-focused workloads. In persistent file systems, data is replicated, and file servers are replaced if they fail. For more information, see Deployment and storage class options for FSx for Lustre file systems.

Amazon FSx for Lustre offers solid state drive (SSD), AWS Interconnect, and hard disk drive (HDD) storage classes that are optimized for different data processing requirements:

- The SSD storage class is optimized for workloads that have small, random file operations and need up to TBps of throughput. It provides consistent sub-millisecond latency access to your full dataset.
- The Intelligent-Tiering storage class is suitable and recommended for most workloads that do not need consistent low-latency across your full dataset. It provides fully-elastic and cost-effective storage, up to multiple TBps of throughput and sub-millisecond latency access to frequently-accessed data with an optional SSD read cache.
- The HDD storage class can be used with workloads that need consistent single-digit ms latency and up to tens of GBps of throughput for your full dataset. You can optionally provision a SSD read cache that is sized to 20% of your HDD storage capacity.

For more information, see FSx for Lustre storage classes.

## FSx for Lustre and data repositories

You can link FSx for Lustre file systems to data repositories on Amazon S3 or to on-premises data stores.

## FSx for Lustre S3 data repository integration

FSx for Lustre integrates with Amazon S3, making it easier for you to process cloud datasets using the Lustre high-performance file system. When linked to an Amazon S3 bucket, an FSx for Lustre file system transparently presents S3 objects as files. Amazon FSx imports listings of all existing files in your S3 bucket at file system creation. Amazon FSx can also import listings of files added to the data repository after the file system is created. You can set the import preferences to match

your workflow needs. The file system also makes it possible for you to write file system data back to S3. Data repository tasks simplify the transfer of data and metadata between your FSx for Lustre file system and its durable data repository on Amazon S3. For more information, see <u>Using</u> data repositories with Amazon FSx for Lustre and Data repository tasks.

## **FSx for Lustre and on-premises data repositories**

With Amazon FSx for Lustre, you can burst your data processing workloads from on-premises into the AWS Cloud by importing data using Direct Connect or Site-to-Site VPN. For more information, see Using Amazon FSx with your on-premises data.

# **Accessing FSx for Lustre file systems**

You can mix and match the compute instance types and Linux Amazon Machine Images (AMIs) that are connected to a single FSx for Lustre file system.

Amazon FSx for Lustre file systems are accessible from compute workloads running on Amazon Elastic Compute Cloud (Amazon EC2) instances, on Amazon Elastic Container Service (Amazon ECS) Docker containers, and containers running on Amazon Elastic Kubernetes Service (Amazon EKS).

- Amazon EC2 You access your file system from your Amazon EC2 compute instances using
  the open-source Lustre client. Amazon EC2 instances can access your file system from other
  Availability Zones within the same Amazon Virtual Private Cloud (Amazon VPC), provided your
  networking configuration provides for access across subnets within the VPC. After your Amazon
  FSx for Lustre file system is mounted, you can work with its files and directories just as you do
  using a local file system.
- Amazon EKS You access Amazon FSx for Lustre from containers running on Amazon EKS
  using the open-source FSx for Lustre CSI driver, as described in Amazon EKS User Guide. Your
  containers running on Amazon EKS can use high-performance persistent volumes (PVs) backed
  by Amazon FSx for Lustre.
- Amazon ECS You access Amazon FSx for Lustre from Amazon ECS Docker containers on Amazon EC2 instances. For more information, see <u>Mounting from Amazon Elastic Container</u> Service.

Amazon FSx for Lustre is compatible with the most popular Linux-based AMIs, including Amazon Linux 2023 and Amazon Linux 2, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, and SUSE Linux. The Lustre client is included with Amazon Linux 2023 and Amazon Linux 2. For RHEL,

CentOS, and Ubuntu, an AWS Lustre client repository provides clients that are compatible with these operating systems.

Using FSx for Lustre, you can burst your compute-intensive workloads from on-premises into the AWS Cloud by importing data over Direct Connect or AWS Virtual Private Network. You can access your Amazon FSx file system from on-premises, copy data into your file system as-needed, and run compute-intensive workloads on in-cloud instances.

For more information on the clients, compute instances, and environments from which you can access FSx for Lustre file systems, see Accessing file systems.

# Integrations with AWS services

Amazon FSx for Lustre integrates with Amazon SageMaker AI as an input data source. When using SageMaker AI with FSx for Lustre, your machine learning training jobs are accelerated by eliminating the initial download step from Amazon S3. Additionally, your total cost of ownership (TCO) is reduced by avoiding the repetitive download of common objects for iterative jobs on the same dataset as you save on S3 requests costs. For more information, see <a href="What Is SageMaker AI">What Is SageMaker AI</a>? in the Amazon SageMaker AI Developer Guide. For a walkthrough of how to use Amazon FSx for Lustre as a data source for SageMaker AI, see <a href="Speed up training on Amazon SageMaker AI using Amazon">SageMaker AI using Amazon</a> FSx for Lustre and Amazon EFS file systems on the AWS Machine Learning Blog.

FSx for Lustre integrates with AWS Batch using EC2 Launch Templates. AWS Batch enables you to run batch computing workloads on the AWS Cloud, including high performance computing (HPC), machine learning (ML), and other asynchronous workloads. AWS Batch automatically and dynamically sizes instances based on job resource requirements. For more information, see <a href="What Is AWS Batch">What Is AWS Batch User Guide</a>.

FSx for Lustre integrates with AWS ParallelCluster. AWS ParallelCluster is an AWS-supported opensource cluster management tool used to deploy and manage HPC clusters. It can automatically create FSx for Lustre file systems or use existing file systems during the cluster creation process.

# Security and compliance

FSx for Lustre file systems support encryption at rest and in transit. Amazon FSx automatically encrypts file system data at rest using keys managed in AWS Key Management Service (AWS KMS). Data in transit is also automatically encrypted on file systems in certain AWS Regions when accessed from supported Amazon EC2 instances. For more information about data encryption in

FSx for Lustre, including AWS Regions where encryption of data in transit is supported, see <u>Data encryption in Amazon FSx for Lustre</u>. Amazon FSx has been assessed to comply with ISO, PCI-DSS, and SOC certifications, and is HIPAA eligible. For more information, see <u>Security in Amazon FSx for Lustre</u>.

# **Assumptions**

In this guide, we make the following assumptions:

- If you use Amazon Elastic Compute Cloud (Amazon EC2), we assume that you're familiar
  with that service. For more information on how to use Amazon EC2, see the <u>Amazon EC2</u>
  documentation.
- We assume that you are familiar with using Amazon Virtual Private Cloud (Amazon VPC). For more information on how to use Amazon VPC, see the Amazon VPC User Guide.
- We assume that you haven't changed the rules on the default security group for your VPC based on the Amazon VPC service. If you have, make sure that you add the necessary rules to allow network traffic from your Amazon EC2 instance to your Amazon FSx for Lustre file system. For more details, see File system access control with Amazon VPC.

# **Pricing for Amazon FSx for Lustre**

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

## **Amazon FSx for Lustre forums**

If you encounter issues while using Amazon FSx for Lustre, check the forums.

# Are you a first-time user of Amazon FSx for Lustre?

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

1. If you're ready to create your first Amazon FSx for Lustre file system, try <u>Getting started with</u> Amazon FSx for Lustre.

Assumptions

- 2. For information on performance, see Amazon FSx for Lustre performance.
- 3. For information on linking your file system to an Amazon S3 bucket data repository, see <u>Using</u> data repositories with Amazon FSx for Lustre.
- 4. For Amazon FSx for Lustre security details, see Security in Amazon FSx for Lustre.
- 5. For information on the scalability limits of Amazon FSx for Lustre, including throughput and file system size, see Service quotas for Amazon FSx for Lustre.
- 6. For information on the Amazon FSx for Lustre API, see the Amazon FSx for Lustre API Reference.

# **Setting up Amazon FSx for Lustre**

Before you use Amazon FSx for Lustre for the first time, complete the tasks in the <u>Sign up for Amazon Web Services</u> section. To complete the <u>Getting started tutorial</u>, make sure the Amazon S3 bucket that you'll link to your file system has the permissions listed in <u>Adding permissions to use data repositories in Amazon S3</u>.

#### **Topics**

- Sign up for Amazon Web Services
- Adding permissions to use data repositories in Amazon S3
- How FSx for Lustre checks for access to linked S3 buckets
- Next step

# Sign up for Amazon Web Services

To set up for AWS, complete the following tasks:

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

## Sign up for an AWS account

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

#### To sign up for an AWS account

- 1. Open https://portal.www.aws.eu/billing/signup.
- Follow the online instructions.

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

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

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <a href="https://www.aws.eu/">https://www.aws.eu/</a> 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 <u>AWS Management Console</u> 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 <u>Signing in as the root user</u> in the *AWS Sign-In User Guide*.
- 2. Turn on multi-factor authentication (MFA) for your root user.

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

#### Create a user with administrative access

1. Enable IAM Identity Center.

For instructions, see <u>Enabling AWS IAM Identity Center</u> 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. FSx for Lustre Lustre User Guide

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

- In IAM Identity Center, create a permission set that follows the best practice of applying leastprivilege permissions.
  - For instructions, see Create a permission set in the AWS IAM Identity Center User Guide.
- Assign users to a group, and then assign single sign-on access to the group. 2.
  - For instructions, see Add groups in the AWS IAM Identity Center User Guide.

## Adding permissions to use data repositories in Amazon S3

Amazon FSx for Lustre is deeply integrated with Amazon S3. This integration means that applications that access your FSx for Lustre file system can also seamlessly access the objects stored in your linked Amazon S3 bucket. For more information, see Using data repositories with Amazon FSx for Lustre.

To use data repositories, you must first allow Amazon FSx for Lustre certain IAM permissions in a role associated with the account for your administrator user.

### To embed an inline policy for a role using the console

- Sign in to the AWS Management Console and open the IAM console at https://eusc-deeast-1.console.amazonaws-eusc.eu/iam/.
- 2. In the navigation pane, choose Roles.
- In the list, choose the name of the role to embed a policy in.
- Choose the **Permissions** tab. 4.
- Scroll to the bottom of the page and choose **Add inline policy**.



### Note

You can't embed an inline policy in a service-linked role in IAM. Because the linked service defines whether you can modify the permissions of the role, you might be

able to add additional policies from the service console, API, or AWS CLI. To view the service-linked role documentation for a service, see **AWS Services That Work with IAM** and choose **Yes** in the **Service-Linked Role** column for your service.

- 6. Choose Creating Policies with the Visual Editor
- 7. Add the following permissions policy statement.

**JSON** 

After you create an inline policy, it is automatically embedded in your role. For more information about service-linked roles, see Using service-linked roles for Amazon FSx.

## How FSx for Lustre checks for access to linked S3 buckets

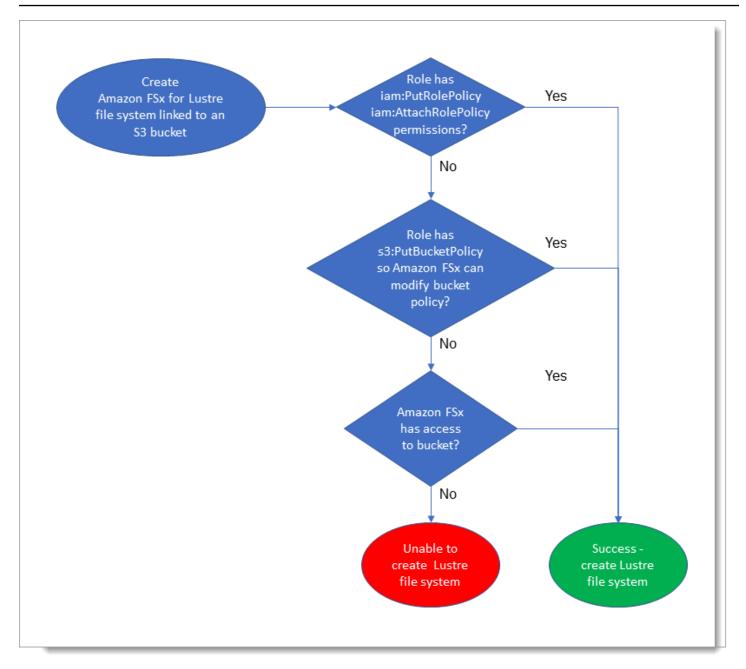
If the IAM role that you use to create the FSx for Lustre file system does not have the iam: AttachRolePolicy and iam: PutRolePolicy permissions, then Amazon FSx checks whether it can update your S3 bucket policy. Amazon FSx can update your bucket policy if the s3: PutBucketPolicy permission is included in your IAM role to allow the Amazon FSx file system to import or export data to your S3 bucket. If allowed to modify the bucket policy, Amazon FSx adds the following permissions to the bucket policy:

- s3:AbortMultipartUpload
- s3:DeleteObject

- s3:PutObject
- s3:Get\*
- s3:List\*
- s3:PutBucketNotification
- s3:PutBucketPolicy
- s3:DeleteBucketPolicy

If Amazon FSx can't modify the bucket policy, it then checks if the existing bucket policy grants Amazon FSx access to the bucket.

If all of these options fail, then the request to create the file system fails. The following diagram illustrates the checks that Amazon FSx follows when determining whether a file system can access the S3 bucket to which it will be linked.



# **Next step**

To get started using FSx for Lustre, see <u>Getting started with Amazon FSx for Lustre</u> for instructions to create your Amazon FSx for Lustre resources.

Next step 12

# **Getting started with Amazon FSx for Lustre**

Following, you can learn how to get started using Amazon FSx for Lustre. These steps walk you through creating an Amazon FSx for Lustre file system and accessing it from your compute instances. Optionally, they show how to use your Amazon FSx for Lustre file system to process the data in your Amazon S3 bucket with your file-based applications.

This getting started exercise includes the following steps.

#### **Topics**

- Prerequisites
- Step 1: Create your FSx for Lustre file system
- Step 2: Install and configure the Lustre client
- Step 3: Mount the file system
- Step 4: Run your workflow
- Step 5: Clean up resources

# **Prerequisites**

To perform this getting started exercise, you need the following:

- An AWS account with the permissions necessary to create an Amazon FSx for Lustre file system and an Amazon EC2 instance. For more information, see Setting up Amazon FSx for Lustre.
- Create a Amazon VPC security group to be associated with your FSx for Lustre file system, and do
  not change it after file system creation. For more information, see <u>To create a security group for
  your Amazon FSx file system</u>.
- An Amazon EC2 instance running a supported Linux release in your virtual private cloud (VPC) based on the Amazon VPC service. For this getting started exercise, we recommend using Amazon Linux 2023. You will install the Lustre client on this EC2 instance, and then mount your FSx for Lustre file system on the EC2 instance. For more information on creating an EC2 instance, see Getting started: Launch an instance or Launch your instance in the Amazon EC2 User Guide.

Besides Amazon Linux 2023, the Lustre client supports the Amazon Linux 2, Red Hat Enterprise Linux (RHEL), CentOS, Rocky Linux, SUSE Linux Enterprise Server, and Ubuntu operating systems. For more information, see Lustre file system and client kernel compatibility.

Prerequisites 13

• When creating your Amazon EC2 instance for this getting started exercise, keep the following in mind:

- We recommend that you create your instance in your default VPC.
- We recommend that you use the default security group when creating your EC2 instance.
- Determine which type of Amazon FSx for Lustre file system you want to create, scratch or persistent. For more information, see <u>Deployment and storage class options for FSx for Lustre file</u> systems.
- Each FSx for Lustre file system requires one IP address for each metadata server (MDS) and one IP address for each storage server (OSS). For more information, see IP addresses for file systems.
- An Amazon S3 bucket storing the data for your workload to process. The S3 bucket will be the linked durable data repository for your FSx for Lustre file system.

# **Step 1: Create your FSx for Lustre file system**

You create your file system in the Amazon FSx console. Note that all FSx for Lustre file systems are built on Lustre version 2.15 when created using the Amazon FSx console.

#### To create your file system

- 1. Open the Amazon FSx console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.
- 2. From the dashboard, choose **Create file system** to start the file system creation wizard.
- 3. Choose **FSx for Lustre** and then choose **Next** to display the **Create File System** page.
  - Begin your configuration with the File system details section.
- 4. For **File system name-optional**, provide a name for your file system. You can use up to 256 Unicode letters, white space, and numbers plus the special characters + = . \_ : /.
- 5. For **Deployment and storage class**, choose one of the options:
  - Choose **Scratch, SSD** deployment for temporary storage and shorter-term processing of data. With SSD storage, you are billed for the amount of storage that you provision.
- 6. Choose the amount of throughput for your file system. You pay for the amount of throughput that you provision.
  - For Persistent SSD storage, choose a **Throughput per unit of storage** value. *Throughput per unit of storage* is the amount of read and write throughput for each 1 tebibyte (TiB) of storage provisioned.

- For Scratch SSD storage, choose a **Throughput per unit of storage** value.
- For Intelligent-Tiering storage, choose a **Throughput capacity** value.
- 7. For **Storage capacity** (SSD storage class only), set the amount of storage capacity for your file system, in TB:
  - For a *Persistent, SSD* deployment type, set this to a value of 1.2 TiB, 2.4 TiB, or increments of 2.4 TiB.
  - For an *EFA-enabled, Persistent, SSD* deployment type, set this value in increments of 4.8 TiB, 9.6 TiB, 19.2 TiB, and 38.4 TiB for 1000, 500, 250, and 125 MBps/TiB throughput tiers, respectively.

You can increase the amount of storage capacity as needed after you create the file system. For more information, see Managing storage capacity.

- 8. For the metadata configuration, choose one of the following options to provision the number of Metadata IOPS for your file system:
  - Choose Automatic (SSD storage class only) if you want Amazon FSx for Lustre to automatically provision and scale the metadata IOPS on your file system based on your file system's storage capacity.
  - Choose **User-provisioned** if you want to specify the number of Metadata IOPS to provision for your file system with SSD or Intelligent-Tiering storage class. Valid values are as follows:
    - For SSD file systems, valid values are 1500, 3000, 6000, 12000, and multiples of 12000
      up to a maximum of 192000.
    - For Intelligent-Tiering file systems, valid values are 6000 and 12000.

For more information about Metadata IOPS, see <u>Lustre metadata performance configuration</u>.

9. For SSD read cache (Intelligent-Tiering only), select either Automatic (proportional to throughput capacity) or Custom (user-provisioned). With the Automatic option, Amazon FSx for Lustre 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. For more information, see <a href="Managing provisioned">Managing provisioned</a>
SSD read cache.

10. For **Data compression type**, choose **NONE** to turn off data compression or choose **LZ4** to turn on data compression with the LZ4 algorithm. For more information, see <u>Lustre data</u> compression.

- 11. In the **Network & security** section, provide the following networking and security group information:
  - For Virtual Private Cloud (VPC), choose the VPC that you want to associate with your file system. For this getting started exercise, choose the same VPC that you chose for your Amazon EC2 instance.
  - For **VPC** security groups, the ID for the default security group for your VPC should be already added.

If you're not using the default security group, make sure that the following inbound rule is added to the security group you're using for this getting started exercise.

| Туре    | Protocol | Port range | Source                                  | Description                    |
|---------|----------|------------|---|--------------------------------|
| All TCP | TCP      | 0-65535    | Custom the_ID_of _this_sec urity_gro up | Inbound Lustre<br>traffic rule |

## Important

- Make sure that the security group you are using follows the configuration
  instructions provided in <u>File system access control with Amazon VPC</u>. You must set
  up the security group to allow inbound traffic on ports 988 and 1018-1023 from
  the security group itself or the full subnet CIDR, which is required to allow the file
  system hosts to communicate with each other.
- If you are creating an EFA-enabled file system, make sure you specify an <a href="EFA-enabled security group"><u>EFA-enabled security group</u></a>.
- For **Subnet**, choose any value from the list of available subnets.
- 12. For the **Encryption** section, the options available vary depending upon which file system type you're creating:

• For a persistent file system, you can choose an AWS Key Management Service (AWS KMS) encryption key to encrypt the data on your file system at rest.

- For a scratch file system, data at rest is encrypted using keys managed by AWS.
- For scratch 2 and persistent file systems, data in transit is encrypted automatically when the file system is accessed from a supported Amazon EC2 instance type. For more information, see Encrypting data in transit.
- 13. For **Data Repository Import/Export** *optional* section, linking your file system to Amazon S3 data repositories is disabled by default. For information about enabling this option and creating a data repository association to an existing S3 bucket, see <u>To link an S3 bucket while creating a file system (console).</u>

## ▲ Important

- Selecting this option also disables backups and you won't be able to enable backups while creating the file system.
- If you link one or more Amazon FSx for Lustre file systems to an Amazon S3 bucket, don't delete the Amazon S3 bucket until all linked file systems have been deleted.
- Intelligent-Tiering file systems don't support linking to Amazon S3 data repositories.
- 14. For **Logging** *optional*, logging is enabled by default. When enabled, failures and warnings for data repository activity on your file system are logged to Amazon CloudWatch Logs. For information about configuring logging, see Managing logging.
- 15. In **Backup and maintenance optional**, you can do the following.
  - Disable the **Daily automatic backup**. This option is enabled by default, unless you enabled **Data Repository Import/Export**.
  - Set the start time for **Daily automatic backup window**.
  - Set the **Automatic backup retention period**, from 1 35 days.
  - Set the Weekly maintenance window start time, or keep it set to the default No preference.

For more information, see <u>Protecting your data with backups</u> and <u>Amazon FSx for Lustre</u> maintenance windows.

16. For **Root Squash** *optional*, root squash is disabled by default. For information about enabling and configuring root squash, see To enable root squash when creating a file system (console).

- 17. Create any tags that you want to apply to your file system.
- 18. Choose **Next** to display the **Create file system summary** page.
- 19. Review the settings for your Amazon FSx for Lustre file system, and choose Create file system.

Now that you've created your file system, note its fully qualified domain name and mount name for a later step. You can find the fully qualified domain name and mount name for a file system by choosing the name of the file system in the **File systems** dashboard, and then choosing **Attach**.

# Step 2: Install and configure the Lustre client

Before you can access your Amazon FSx for Lustre file system from your Amazon EC2 instance, you have to do the following:

- Verify your EC2 instance meets the minimum kernel requirements.
- Update the kernel if needed.
- Download and install the Lustre client.

#### To check the kernel version and download the Lustre client

- 1. Open a terminal window on your EC2 instance.
- 2. Determine which kernel is currently running on your compute instance by running the following command.

```
uname -r
```

- 3. Do one of the following:
  - If the command returns 6.1.79-99.167.amzn2023.x86\_64 for x86-based EC2 instances, or 6.1.79-99.167.amzn2023.aarch64 or higher for Graviton2-based EC2 instances, download and install the Lustre client with the following command.

```
sudo dnf install -y lustre-client
```

• If the command returns a result less than 6.1.79-99.167.amzn2023.x86\_64 for x86-based EC2 instances, or less than 6.1.79-99.167.amzn2023.aarch64 for Graviton2-

Install the Lustre client 18

based EC2 instances, update the kernel and reboot your Amazon EC2 instance by running the following command.

```
sudo dnf -y update kernel && sudo reboot
```

Confirm that the kernel has been updated using the **uname -r** command. Then download and install the Lustre client as described above.

For information about installing the Lustre client on other Linux distributions, see <u>Installing</u> the Lustre client.

# **Step 3: Mount the file system**

To mount your file system, you will create a mounting directory, or mount point, and then mount the file system on to your client, and verify that your client can access the file system.

#### To mount your file system

1. Make a directory for the mount point with the following command.

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

- 2. Mount the Amazon FSx for Lustre file system to the directory that you created. Use the following command and replace the following items:
  - Replace file\_system\_dns\_name with the actual file system's Domain Name System (DNS)
    name.
  - Replace *mount name* with the file system's mount name, which you can get by running the **describe-file-systems** AWS CLI command or the <u>DescribeFileSystems</u> API operation.

```
sudo mount -t lustre -o relatime,flock file_system_dns_name@tcp:/mountname /mnt/fsx
```

This command mounts your file system with two options, -o relatime and flock:

• relatime – While the atime option maintains atime (inode access times) data for each time a file is accessed, the relatime option also maintains atime data, but not for each time that a file is accessed. With the relatime option enabled, atime data is written to

FSx for Lustre Lustre User Guide

disk only if the file has been modified since the atime data was last updated (mtime), or if the file was last accessed more than a certain amount of time ago (6 hours by default). Using either the relatime or a time option will optimize the file release processes.



### Note

If your workload requires precise access time accuracy, you can mount with the atime mount option. However, doing so can impact workload performance by increasing the network traffic required to maintain precise access time values. If your workload does not require metadata access time, using the noatime mount option to disable updates to access time can provide a performance gain. Be aware that atime focused processes like file release or releasing data validity will be inaccurate in their release.

- flock Enables file locking for your file system. If you don't want file locking enabled, use the mount command without flock.
- Verify that the mount command was successful by listing the contents of the directory to which you mounted the file system /mnt/fsx, by using the following command.

```
ls /mnt/fsx
import-path lustre
```

You can also use the df command, following.

| df                           |            |         |            |                   |
|------------------------------|------------|---------|------------|-------------------|
| Filesystem                   | 1K-blocks  | Used    | Available  | Use% Mounted on   |
| devtmpf                      | 1001808    | 0       | 1001808    | 0% /dev           |
| tmpfs                        | 1019760    | 0       | 1019760    | 0% /dev/shm       |
| tmpfs                        | 1019760    | 392     | 1019368    | 1% /run           |
| tmpfs                        | 1019760    | 0       | 1019760    | 0% /sys/fs/cgroup |
| /dev/xvda1                   | 8376300    | 1263180 | 7113120    | 16% /             |
| 123.456.789.0@tcp:/mountname | 3547698816 | 13824   | 3547678848 | 1% /mnt/fsx       |
| tmpfs                        | 203956     | 0       | 203956     | 0% /run/user/1000 |

The results show the Amazon FSx file system mounted on /mnt/fsx.

# Step 4: Run your workflow

Now that your file system has been created and mounted to a compute instance, you can use it to run your high-performance compute workload.

You can create a data repository association to link your file system to an Amazon S3 data repository, For more information, see Linking your file system to an Amazon S3 bucket.

After you've linked your file system to an Amazon S3 data repository, you can export data that you've written to your file system back to your Amazon S3 bucket at any time. From a terminal on one of your compute instances, run the following command to export a file to your Amazon S3 bucket.

```
sudo lfs hsm_archive file_name
```

For more information on how to run this command on a folder or large collection of files quickly, see Exporting files using HSM commands.

## **Step 5: Clean up resources**

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

#### To clean up resources

1. If you want to do a final export, run the following command.

```
nohup find /mnt/fsx -type f -print0 | xargs -0 -n 1 sudo lfs hsm_archive &
```

- 2. On the Amazon EC2 console, terminate your instance. For more information, see <u>Terminate</u> <u>Your Instance</u> in the *Amazon EC2 User Guide*.
- 3. On the Amazon FSx for Lustre console, delete your file system with the following procedure:
  - a. In the navigation pane, choose **File systems**.
  - b. Choose the file system that you want to delete from list of file systems on the dashboard.
  - c. For **Actions**, choose **Delete file system**.
  - d. In the dialog box that appears, choose if you want to take a final backup of the file system. Then provide the file system ID to confirm the deletion. Choose **Delete file** system.

Step 4: Run your workflow 21

4. If you created an Amazon S3 bucket for this exercise, and if you don't want to preserve the data you exported, you can now delete it. For more information, see <u>Deleting a bucket</u> in the *Amazon Simple Storage Service User Guide*.

Step 5: Clean up resources 22

# Deployment and storage class options for FSx for Lustre file systems

Amazon FSx for Lustre provides two file system deployment options: **persistent** and **scratch**. It provides three storage classes: **SSD** (solid state drive), **AWS Interconnect**, and **HDD** (hard disk drive).

You choose the file system deployment type and storage class when you create a new file system, using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx for Lustre API. For more information, see <a href="Step 1: Create your FSx for Lustre file system">Step 1: Create your FSx for Lustre file system</a> and <a href="CreateFileSystem">CreateFileSystem</a> in the Amazon FSx API Reference.

# Persistent file systems

Persistent file systems are designed for longer-term storage and workloads, and the file servers are highly available. For SSD and HDD-based file systems, data is automatically replicated within the same Availability Zone in which the file system is located. For Intelligent-Tiering file systems, data is replicated across multiple Availability Zones. The data volumes attached to the file servers are replicated independently from the file servers to which they are attached.

Amazon FSx continuously monitors persistent file systems for hardware failures, and automatically replaces infrastructure components in the event of a failure. On a persistent file system, if a file server becomes unavailable, it's replaced automatically within minutes of failure. During that time, client requests for data on that server transparently retry and eventually succeed after the file server is replaced. Data on persistent file systems is replicated on disks, and any failed disks are automatically replaced transparently.

Use persistent file systems for longer-term storage and for throughput-focused workloads that run for extended periods or indefinitely, and that might be sensitive to disruptions in availability.

Persistent deployment types automatically encrypt data in transit when they are accessed from Amazon EC2 instances that support encryption in transit.

Amazon FSx for Lustre supports two persistent deployment types: Persistent 1 and Persistent 2.

Persistent file systems 23

# Persistent 2 deployment type

Persistent 2 is the latest generation of Persistent deployment type, and is best-suited for use cases that require longer-term storage, and that require the highest levels of IOPS and throughput. Persistent 2 file systems support SSD and Intelligent-Tiering storage classes.

You can create Persistent 2 file systems with a metadata configuration and EFA enabled using the Amazon FSx console, AWS Command Line Interface, and Amazon FSx API.

## Persistent 1 deployment type

The Persistent 1 deployment type is well-suited for use cases that require longer-term storage. Persistent 1 deployment types support SSD (solid state drive) and HDD (hard disk drive) storage classes.

You can create Persistent 1 deployment types only by using the AWS CLI and the Amazon FSx API.

## Scratch file systems

Scratch file systems are designed for temporary storage and shorter-term processing of data. Data isn't replicated and doesn't persist if a file server fails. Scratch file systems provide high burst throughput of up to six times the baseline throughput of 200 MBps per TiB of storage capacity. For more information, see Performance characteristics of SSD and HDD storage classes.

Use scratch file systems when you need cost-optimized storage for short-term, processing-heavy workloads.

On a scratch file system, file servers aren't replaced if they fail and data isn't replicated. If a file server or a storage disk becomes unavailable on a scratch file system, files stored on other servers are still accessible. If clients try to access data that is on the unavailable server or disk, clients experience an immediate I/O error.

The following table illustrates the availability or durability that scratch file systems of example sizes are designed for, over the course of a day and a week. Because larger file systems have more file servers and more disks, the probabilities of failure are increased.

| File system size (TiB) | Number of file servers | Availability/durab ility over one day | Availability/durab ility over one week |
|------------------------|------------------------|---------------------------------------|--|
| 1.2                    | 2                      | 99.9%                                 | 99.4%                                  |

| File system size (TiB) | Number of file servers | Availability/durab ility over one day | Availability/durab ility over one week |
|------------------------|------------------------|---------------------------------------|--|
| 2.4                    | 2                      | 99.9%                                 | 99.4%                                  |
| 4.8                    | 3                      | 99.8%                                 | 99.2%                                  |
| 9.6                    | 5                      | 99.8%                                 | 98.6%                                  |
| 50.4                   | 22                     | 99.1%                                 | 93.9%                                  |

# IP addresses for file systems

Each FSx for Lustre file system requires one IP address for each metadata server (MDS) and one IP address for each storage server (OSS).

## File systems using SSD or HDD storage class

| File<br>System<br>Type        | Throughput, MBps/TiB | Storage per OSS  |
|-------------------------------|----------------------|------------------|
| Persistent<br>2 EFA*          | 125                  | 38.4 TiB per OSS |
|                               | 250                  | 19.2 TiB per OSS |
|                               | 500                  | 9.6 TiB per OSS  |
|                               | 1000                 | 4.8 TiB per OSS  |
| Persisten<br>t 2 non-<br>EFA* | 125, 250, 500, 1000  | 2.4 TiB per OSS  |
| Persistent<br>1 SSD           | 50, 100, 200         | 2.4 TiB per OSS  |
| Persistent<br>HDD             | 12                   | 6 TiB per OSS    |
|                               | 40                   | 1.8 TiB per OSS  |

IP addresses 25

| File<br>System<br>Type | Throughput, MBps/TiB | Storage per OSS |
|------------------------|----------------------|-----------------|
| Scratch 2              | 200                  | 2.4 TiB per OSS |
| Scratch 1              | 200                  | 3.6 TiB per OSS |

#### File systems using Intelligent-Tiering storage class

| File System Type         | Throughput per OSS |
|--------------------------|--------------------|
| Intelligent-Tierin<br>g* | 4000 MBps per OSS  |

## Note

\* Amazon FSx provisions a metadata server for every 12,000 Metadata IOPS on Persistent 2 SSD and Intelligent-Tiering file systems configured with metadata configuration. Amazon FSx for Lustre Intelligent-Tiering file systems support a maximum of 512 TiB of storage per OSS.

# **FSx for Lustre storage classes**

Amazon FSx for Lustre offers solid state drive (SSD), AWS Interconnect, and hard disk drive (HDD) storage classes that are optimized for different data processing requirements:

 The SSD storage class provides low-latency (sub-millisecond) access to your full dataset. The SSD storage class is provisioned, which means that you specify a file system size and pay storage costs for the amount of storage provisioned. Use the SSD storage class for latency-sensitive workloads that require the performance of all-flash storage across all data.

Persistent 2 file systems with SSD storage support higher levels of throughput per unit of storage (that is, 250, 500, or 1000 MBps per TiB) compared to Persistent 1 file systems. For a Persistent 1 file system with SSD storage, the throughput per unit of storage is either 50, 100, or

FSx for Lustre storage classes 26

200 MBps per TiB. For a Scratch file system with SSD storage, the throughput per unit of storage is 200 MBps per TiB.

- The Intelligent-Tiering storage class provides fully elastic, intelligently tiered storage. Elasticity means that you pay for the amount of data you store and do not have to specify a file system size. Intelligent tiering means that you automatically pay less to store data that you haven't accessed recently. This storage class automatically optimizes costs by tiering cold data to lower-cost storage tiers. You can provision an optional SSD read cache for low-latency (sub-millisecond) access to your frequently-accessed data. The Intelligent-Tiering storage class provides the best balance of price and performance for most workloads. Use the Intelligent-Tiering storage class for workloads that are cache-friendly and do not require the performance of all-flash storage across all data. Intelligent-Tiering file systems support throughput capacities in increments of 4000 MBps.
- The HDD storage class can be used with workloads that need consistent single-digit ms latency
  across all data. You can provision an optional SSD read cache that is sized to 20% of your
  HDD storage capacity to provide low-latency access to frequently-accessed data. With HDD
  storage, you specify a file system size and pay for the amount of storage that you provision. For
  a Persistent 1 file system with HDD storage, the throughput per unit of storage is either 12 or 40
  MBps per TiB.

For more information about performance of these storage classes, see <u>Performance characteristics</u> of SSD and HDD storage classes and <u>Performance characteristics of Intelligent-Tiering storage</u> class.

# How the Intelligent-Tiering storage class tiers data

The Amazon FSx Intelligent-Tiering storage class automatically stores data in three access tiers. It is designed to optimize storage costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. The Intelligent-Tiering storage class automatically tiers data based on last access time, thus automatically optimizing costs for less active data:

- Data accessed in the last 30 days is stored in the Frequent Access tier.
- Data that hasn't been accessed in 30 consecutive days automatically moves to the Infrequent Access tier, and costs less than data in the Frequent Access tier.
- Data that hasn't been accessed in 90 consecutive days automatically moves to the Archive Instant Access tier, and costs less than data in the Infrequent Access tier.

When you access data in the Infrequent Access or Archive Instant Access tiers, the data automatically moves back to the Frequent Access tier. All access to non-cached data has the same performance characteristics, independent of the data's tier, and there are no additional IOPS, retrieval, or transition costs beyond your normal read/write operation costs.

# Deployment type availability

Scratch 2, Persistent 1, and Persistent 2 deployment types are available in the following AWS Regions:

| AWS Region                       | Persistent 2 | Persistent 1 | Scratch 2 |
|----------------------------------|--------------|--------------|-----------|
| US East (Ohio)                   | ✓            | ✓            | ✓         |
| US East (N. Virginia)            | ✓            | ✓            | ✓         |
| US East (Atlanta) Local Zone     | <b>√</b> *   |              |           |
| US East (Dallas) Local Zone      | <b>√</b> *   |              |           |
| US West (N. California)          | ✓            | ✓            | ✓         |
| US West (Los Angeles) Local Zone |              | ✓            | ✓         |
| US West (Oregon)                 | ✓            | ✓            | ✓         |
| US West (Phoenix) Local Zone     | <b>√</b> *   |              |           |
| Africa (Cape Town)               |              | ✓            | ✓         |
| Asia Pacific (Hong Kong)         | ✓            | ✓            | ✓         |
| Asia Pacific (Hyderabad)         |              | ✓            | ✓         |
| Asia Pacific (Jakarta)           |              | ✓            | ✓         |
| Asia Pacific (Malaysia)          | <b>√</b> *   |              |           |
| Asia Pacific (Melbourne)         |              | ✓            | ✓         |
| Asia Pacific (Mumbai)            | ✓            | ✓            | ✓         |

Deployment type availability 28

FSx for Lustre User Guide

| AWS Region               | Persistent 2 | Persistent 1 | Scratch 2    |
|--------------------------|--------------|--------------|--------------|
| Asia Pacific (Osaka)     |              | ✓            | ✓            |
| Asia Pacific (Seoul)     | ✓            | ✓            | ✓            |
| Asia Pacific (Singapore) | ✓            | ✓            | ✓            |
| Asia Pacific (Sydney)    | ✓            | ✓            | ✓            |
| Asia Pacific (Taipei)    | ✓ *          |              |              |
| Asia Pacific (Thailand)  | <b>√</b> *   |              |              |
| Asia Pacific (Tokyo)     | ✓            | ✓            | ✓            |
| Canada (Central)         | ✓            | ✓            | ✓            |
| Canada West (Calgary)    | ✓ *          |              |              |
| Europe (Frankfurt)       | ✓            | ✓            | ✓            |
| Europe (Ireland)         | ✓            | ✓            | ✓            |
| Europe (London)          | ✓            | ✓            | $\checkmark$ |
| Europe (Milan)           |              | ✓            | ✓            |
| Europe (Paris)           |              | $\checkmark$ | $\checkmark$ |
| Europe (Spain)           |              | ✓            | ✓            |
| Europe (Stockholm)       | ✓            | ✓            | ✓            |
| Europe (Zurich)          |              | ✓            | ✓            |
| Israel (Tel Aviv)        | ✓ *          |              | ✓            |
| Mexico (Central)         | <b>√</b> *   |              |              |
| Middle East (Bahrain)    |              | ✓            | ✓            |

| AWS Region                | Persistent 2 | Persistent 1 | Scratch 2 |
|---------------------------|--------------|--------------|-----------|
| Middle East (UAE)         |              | ✓            | ✓         |
| South America (São Paulo) |              | ✓            | ✓         |
| AWS GovCloud (US-East)    |              | ✓            | ✓         |
| AWS GovCloud (US-West)    |              | ✓            | ✓         |

# Note

\* These AWS Regions support Persistent-125 and Persistent-250 file systems with SSD storage class without EFA.

# Using data repositories with Amazon FSx for Lustre

Amazon FSx for Lustre provides high-performance file systems optimized for fast workload processing. It can support workloads such as machine learning, high performance computing (HPC), video processing, financial modeling, and electronic design automation (EDA). These workloads commonly require data to be presented using a scalable, high-speed file system interface for data access. Often, the datasets used for these workloads are stored in long-term data repositories in Amazon S3. FSx for Lustre is natively integrated with Amazon S3, making it easier to process datasets with the Lustre file system.

### Note

- File system backups aren't supported on file systems that are linked an Amazon S3 data repository. For more information, see Protecting your data with backups.
- Intelligent-Tiering file systems don't support linking to Amazon S3 data repositories.

### **Topics**

- · Overview of data repositories
- POSIX metadata support for data repositories
- Linking your file system to an Amazon S3 bucket
- Importing changes from your data repository
- Exporting changes to the data repository
- Data repository tasks
- Releasing files
- Using Amazon FSx with your on-premises data
- Data repository event logs
- Working with older deployment types

# Overview of data repositories

When you use Amazon FSx for Lustre with data repositories, you can ingest and process large volumes of file data in a high-performance file system by using automatic import and import

data repository tasks. At the same time, you can write results to your data repositories by using automatic export or export data repository tasks. With these features, you can restart your workload at any time using the latest data stored in your data repository.



#### Note

Data repository associations, automatic export, and support for multiple data repositories aren't available on FSx for Lustre 2.10 file systems or Scratch 1 file systems.

FSx for Lustre is deeply integrated with Amazon S3. This integration means that you can seamlessly access the objects stored in your Amazon S3 buckets from applications that mount your FSx for Lustre file system. You can also run your compute-intensive workloads on Amazon EC2 instances in the AWS Cloud and export the results to your data repository after your workload is complete.

In order to access objects in the Amazon S3 data repository as files and directories on the file system, file and directory metadata must be loaded into the file system. You can load metadata from a linked data repository when you create a data repository association.

Additionally you can import file and directory metadata from your linked data repositories to the file system using automatic import or using an import data repository task. When you turn on automatic import for a data repository association, your file system automatically imports file metadata as files are created, modified, and/or deleted in the S3 data repository. Alternatively, you can import metadata for new or changed files and directories using an import data repository task.



### Note

Automatic import and import data repository tasks can be used simultaneously on a file system.

You can also export files and their associated metadata in your file system to your data repository using automatic export or using an export data repository task. When you turn on automatic export on a data repository association, your file system automatically exports file data and metadata as files are created, modified, or deleted. Alternatively, you can export files or directories using an export data repository task. When you use an export data repository task, file data and metadata that were created or modified since the last such task are exported.

Overview of data repositories



#### Note

 Automatic export and export data repository tasks can't be used simultaneously on a file system.

 Data repository associations only export regular files, symlinks and directories. This means all the other type of files (FIFO special, block special, character special, and socket) won't be exported as part of the export processes like automatic export and export data repository tasks.

FSx for Lustre also supports cloud bursting workloads with on-premises file systems by enabling you to copy data from on-premises clients using Direct Connect or VPN.



#### Important

If you have linked one or more FSx for Lustre file systems to a data repository on Amazon S3, don't delete the Amazon S3 bucket until you have deleted or unlinked all linked file systems.

# Region and account support for linked S3 buckets

When you create links to S3 buckets, keep in mind the following Region and account support limitations:

- Automatic export supports cross-Region configurations. The Amazon FSx file system and the linked S3 bucket can be located in the same AWS Region or in different AWS Regions.
- Automatic import doesn't support cross-Region configurations. Both the Amazon FSx file system and the linked S3 bucket must be located in the same AWS Region.
- Both automatic export and automatic import support cross-Account configurations. The Amazon FSx file system and the linked S3 bucket can belong to the same AWS account or to different AWS accounts.

# POSIX metadata support for data repositories

Amazon FSx for Lustre automatically transfers Portable Operating System Interface (POSIX) metadata for files, directories, and symbolic links (symlinks) when importing and exporting data to and from a linked data repository on Amazon S3. When you export changes in your file system to its linked data repository, FSx for Lustre also exports POSIX metadata changes as S3 object metadata. This means that if another FSx for Lustre file system imports the same files from S3, the files will have the same POSIX metadata in that file system, including ownership and permissions.

FSx for Lustre imports only S3 objects that have POSIX-compliant object keys, such as the following.

```
mydir/
mydir/myfile1
mydir/mysubdir/
mydir/mysubdir/myfile2.txt
```

FSx for Lustre stores directories and symlinks as separate objects in the linked data repository on S3, For directories, FSx for Lustre creates an S3 object with a key name that ends with a slash ("/"), as follows:

- The S3 object key mydir/ maps to the FSx for Lustre directory mydir/.
- The S3 object key mydir/mysubdir/ maps to the FSx for Lustre directory mydir/mysubdir/.

For symlinks, FSx for Lustre uses the following Amazon S3 schema:

- S3 object key The path to the link, relative to the FSx for Lustre mount directory
- **S3 object data** The target path of this symlink
- S3 object metadata The metadata for the symlink

FSx for Lustre stores POSIX metadata, including ownership, permissions, and timestamps for files, directories, and symbolic links, in S3 objects as follows:

- Content-Type The HTTP entity header used to indicate the media type of the resource for web browsers.
- x-amz-meta-file-permissions The file type and permissions in the format <octal file type><octal permission mask>, consistent with st\_mode in the Linux stat(2) man page.

POSIX metadata support 34



#### Note

FSx for Lustre doesn't import or retain setuid information.

x-amz-meta-file-owner - The owner user ID (UID) expressed as an integer.

- x-amz-meta-file-group The group ID (GID) expressed as an integer.
- x-amz-meta-file-atime The last-accessed time in nanoseconds since the beginning of the Unix epoch. Terminate the time value with ns; otherwise FSx for Lustre interprets the value as milliseconds.
- x-amz-meta-file-mtime The last-modified time in nanoseconds since the beginning of the Unix epoch. Terminate the time value with ns; otherwise, FSx for Lustre interprets the value as milliseconds.
- x-amz-meta-user-agent The user agent, ignored during FSx for Lustre import. During export, FSx for Lustre sets this value to aws-fsx-lustre.

When importing objects from S3 that don't have associated POSIX permissions, the default POSIX permission that FSx for Lustre assigns to a file is 755. This permission allows read and execute access for all users and write access for the owner of the file.



#### Note

FSx for Lustre doesn't retain any user-defined custom metadata on S3 objects.

# Hard links and exporting to Amazon S3

If automatic export (with NEW and CHANGED policies) is enabled on a DRA in your file system, each hard link contained within the DRA is exported to Amazon S3 as a separate S3 object for each hard link. If a file with multiple hard links is modified on the file system, all of the copies in S3 are updated, regardless of which hard link was used when changing the file.

If hard links are exported to S3 using data repository tasks (DRTs), each hard link contained within the paths specified for the DRT is exported to S3 as a separate S3 object for each hard link. If a file with multiple hard links is modified on the file system, each copy in S3 is updated at the time the respective hard link is exported, regardless of which hard link was used when changing the file.

Exporting hard links 35

### Important

When a new FSx for Lustre file system is linked to an S3 bucket to which hard links were previously exported by another FSx for Lustre file system, AWS DataSync, or Amazon FSx File Gateway, the hard links are subsequently imported as separate files on the new file system.

#### Hard links and released files

A released file is a file whose metadata is present in the file system, but whose content is only stored in S3. For more information on released files, see Releasing files.

### Important

The use of hard links in a file system that has data repository associations (DRAs) is subject to the following limitations:

- Deleting and recreating a released file that has multiple hard links may cause the content of all hard links to be overwritten.
- Deleting a released file will delete content from all hard links that reside outside of a data repository association.
- Creating a hard link to a released file whose corresponding S3 object is in either of the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes will not create a new object in S3 for the hard link.

# Walkthrough: Attaching POSIX permissions when uploading objects into an Amazon S3 bucket

The following procedure walks you through the process of uploading objects into Amazon S3 with POSIX permissions. Doing so allows you to import the POSIX permissions when you create an Amazon FSx file system that is linked to that S3 bucket.

### To upload objects with POSIX permissions to Amazon S3

From your local computer or machine, use the following example commands to create a test directory (s3cptestdir) and file (s3cptest.txt) that will be uploaded to the S3 bucket.

```
$ mkdir s3cptestdir
$ echo "S3cp metadata import test" >> s3cptestdir/s3cptest.txt
$ ls -ld s3cptestdir/ s3cptestdir/s3cptest.txt
drwxr-xr-x 3 500 500 96 Jan 8 11:29 s3cptestdir/
-rw-r--r-- 1 500 500 26 Jan 8 11:29 s3cptestdir/s3cptest.txt
```

The newly created file and directory have a file owner user ID (UID) and group ID (GID) of 500 and permissions as shown in the preceding example.

Call the Amazon S3 API to create the directory s3cptestdir with metadata permissions. You
must specify the directory name with a trailing slash (/). For information about supported
POSIX metadata, see POSIX metadata support for data repositories.

Replace **bucket\_name** with the actual name of your S3 bucket.

3. Verify the POSIX permissions are tagged to S3 object metadata.

```
$ aws s3api head-object --bucket bucket_name --key s3cptestdir/
{
    "AcceptRanges": "bytes",
    "LastModified": "Fri, 08 Jan 2021 17:32:27 GMT",
    "ContentLength": 0,
    "ETag": "\"d41d8cd98f00b204e9800998ecf8427e\"",
    "VersionId": "bAlhCoWq7aIEjc3R6Myc6U0b8sHHtJkR",
    "ContentType": "binary/octet-stream",
    "Metadata": {
        "user-agent": "aws-fsx-lustre",
        "file-atime": "1595002920000000000ns",
        "file-owner": "500",
        "file-permissions": "0100664",
        "file-group": "500",
        "file-mtime": "1595002920000000000ns"
    }
}
```

4. Upload the test file (created in step 1) from your computer to the S3 bucket with metadata permissions.

5. Verify the POSIX permissions are tagged to S3 object metadata.

```
$ aws s3api head-object --bucket bucket_name --key s3cptestdir/s3cptest.txt
{
    "AcceptRanges": "bytes",
    "LastModified": "Fri, 08 Jan 2021 17:33:35 GMT",
    "ContentLength": 26,
    "ETag": "\"eb33f7e1f44a14a8e2f9475ae3fc45d3\"",
    "VersionId": "w9ztRoEhB832m8NC3a_JTlTyIx7Uzq16",
    "ContentType": "text/plain",
    "Metadata": {
        "user-agent": "aws-fsx-lustre",
        "file-atime": "1595002920000000000ns",
        "file-owner": "500",
        "file-permissions": "0100664",
        "file-group": "500",
        "file-mtime": "1595002920000000000ns"
    }
}
```

6. Verify permissions on the Amazon FSx file system linked to the S3 bucket.

```
$ sudo lfs df -h /fsx
UUID
                                               Available Use% Mounted on
                           bytes
                                        Used
3rnxfbmv-MDT0000_UUID
                                         6.1M
                                                    34.4G
                                                            0% /fsx[MDT:0]
                            34.4G
3rnxfbmv-OST0000_UUID
                             1.1T
                                         4.5M
                                                     1.1T
                                                            0% /fsx[0ST:0]
filesystem_summary:
                            1.1T
                                        4.5M
                                                    1.1T
                                                           0% /fsx
$ cd /fsx/s3cptestdir/
$ ls -ld s3cptestdir/
drw-rw-r-- 2 500 500 25600 Jan 8 17:33 s3cptestdir/
$ ls -ld s3cptestdir/s3cptest.txt
```

-rw-rw-r-- 1 500 500 26 Jan 8 17:33 s3cptestdir/s3cptest.txt

Both the s3cptestdir directory and the s3cptest.txt file have POSIX permissions imported.

# Linking your file system to an Amazon S3 bucket

You can link your Amazon FSx for Lustre file system to data repositories in Amazon S3. You can create the link when creating the file system or at any time after the file system has been created.

A link between a directory on the file system and an S3 bucket or prefix is called a data repository association (DRA). You can configure a maximum of 8 data repository associations on an FSx for Lustre file system. A maximum of 8 DRA requests can be queued, but only one request can be worked on at a time for the file system. Each DRA must have a unique FSx for Lustre file system directory and a unique S3 bucket or prefix associated with it.



### Note

Data repository associations, automatic export, and support for multiple data repositories aren't available on FSx for Lustre 2.10 file systems or Scratch 1 file systems.

In order to access objects on the S3 data repository as files and directories on the file system, file and directory metadata must be loaded into the file system. You can load metadata from a linked data repository when you create the DRA or load metadata for batches of files and directories that you want to access using the FSx for Lustre file system at a later time using an import data repository task, or use automatic export to load metadata automatically when objects are added to, changed in, or deleted from the data repository.

You can configure a DRA for automatic import only, for automatic export only, or for both. A data repository association configured with both automatic import and automatic export propagates data in both directions between the file system and the linked S3 bucket. As you make changes to data in your S3 data repository, FSx for Lustre detects the changes and then automatically imports the changes to your file system. As you create, modify, or delete files, FSx for Lustre automatically exports the changes to Amazon S3 asynchronously once your application finishes modifying the file.

### Important

• If you modify the same file in both the file system and the S3 bucket, you should ensure application-level coordination to prevent conflicts. FSx for Lustre doesn't prevent conflicting writes in multiple locations.

 For files marked with an immutable attribute, FSx for Lustre is unable to synchronize changes between your FSx for Lustre file system and an S3 bucket linked to the file system. Setting an immutable flag for an extended period of time can cause the performance of data movement between Amazon FSx and S3 to degrade.

When you create a data repository association, you can configure the following properties:

• File system path – Enter a local path on the file system that points to a directory (such as /ns1/) or subdirectory (such as /ns1/subdir/) that will be mapped one-to-one with the specified data repository path below. The leading forward slash in the name is required. Two data repository associations cannot have overlapping file system paths. For example, if a data repository is associated with file system path /ns1, then you cannot link another data repository with file system path /ns1/ns2.

### Note

If you specify only a forward slash (/) as the file system path, you can link only one data repository to the file system. You can only specify "/" as the file system path for the first data repository associated with a file system.

• Data repository path – Enter a path in the S3 data repository. The path can be an S3 bucket or prefix in the format s3://bucket-name/prefix/. This property specifies where in the S3 data repository files will be imported from or exported to. FSx for Lustre will append a trailing "/" to your data repository path if you don't provide one. For example, if you provide a data repository path of s3://amzn-s3-demo-bucket/my-prefix, FSx for Lustre will interpret it as s3:// amzn-s3-demo-bucket/my-prefix/.

Two data repository associations cannot have overlapping data repository paths. For example, if a data repository with path s3://amzn-s3-demo-bucket/my-prefix/is linked to the file system, then you cannot create another data repository association with data repository path s3://amzn-s3-demo-bucket/my-prefix/my-sub-prefix.

Import metadata from repository – You can select this option to import metadata from the
entire data repository immediately after creating the data repository association. Alternatively,
you can run an import data repository task to load all or a subset of the metadata from the
linked data repository into the file system at any time after the data repository association is
created.

- Import settings Choose an import policy that specifies the type of updated objects (any combination of new, changed, and deleted) that will be automatically imported from the linked S3 bucket to your file system. Automatic import (new, changed, deleted) is turned on by default when you add a data repository from the console, but is disabled by default when using the AWS CLI or Amazon FSx API.
- Export settings Choose an export policy that specifies the type of updated objects (any combination of new, changed, and deleted) that will be automatically exported to the S3 bucket. Automatic export (new, changed, deleted) is turned on by default when you add a data repository from the console, but is disabled by default when using the AWS CLI or Amazon FSx API.

The **File system path** and **Data repository path** settings provide a 1:1 mapping between paths in Amazon FSx and object keys in S3.

### **Topics**

- Creating a link to an S3 bucket
- Updating data repository association settings
- Deleting an association to an S3 bucket
- Viewing data repository association details
- Data repository association lifecycle state
- Working with server-side encrypted Amazon S3 buckets

### Creating a link to an S3 bucket

The following procedures walk you through the process of creating a data repository association for an FSx for Lustre file system to an existing S3 bucket, using the AWS Management Console and AWS Command Line Interface (AWS CLI). For information on adding permissions to an S3 bucket in order to link it to your file system, see <u>Adding permissions to use data repositories in Amazon S3</u>.



### Note

Data repositories cannot be linked to file systems that have file system backups enabled. Disable backups before linking to a data repository.

### To link an S3 bucket while creating a file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Follow the procedure for creating a new file system described in Step 1: Create your FSx for Lustre file system in the Getting Started section.
- Open the **Data Repository Import/Export** *optional* section. The feature is disabled by 3. default.
- 4. Choose Import data from and export data to S3.
- In the Data repository association information dialog, provide information for the following fields.
  - File system path: Enter the name of a high-level directory (such as /ns1) or subdirectory (such as /ns1/subdir) within the Amazon FSx file system that will be associated with the S3 data repository. The leading forward slash in the path is required. Two data repository associations cannot have overlapping file system paths. For example, if a data repository is associated with file system path /ns1, then you cannot link another data repository with file system path /ns1/ns2. The **File system path** setting must be unique across all the data repository associations for the file system.
  - Data repository path: Enter the path of an existing S3 bucket or prefix to associate with your file system (for example, s3://amzn-s3-demo-bucket/my-prefix). Two data repository associations cannot have overlapping data repository paths. The **Data repository** path setting must be unique across all the data repository associations for the file system.
  - Import metadata from repository: Select this property to optionally run an import data repository task to import metadata immediately after the link is created.
- For Import settings optional, set an Import Policy that determines how your file and directory listings are kept up to date as you add, change, or delete objects in your S3 bucket. For example, choose New to import metadata to your file system for new objects created in the S3 bucket. For more information on import policies, see Automatically import updates from your S3 bucket.

7. For **Export policy**, set an export policy that determines how your files are exported to your linked S3 bucket as you add, change, or delete objects in your file system. For example, choose **Changed** to export objects whose content or metadata has been changed on your file system. For more information about export policies, see <u>Automatically export updates to your S3 bucket</u>.

8. Continue with the next section of the file system creation wizard.

### To link an S3 bucket to an existing file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **File systems** and then select the file system that you want to create a data repository association for.
- 3. Choose the **Data repository** tab.
- 4. In the **Data repository associations** pane, choose **Create data repository association**.
- 5. In the **Data repository association information** dialog, provide information for the following fields.
  - File system path: Enter the name of a high-level directory (such as /ns1) or subdirectory (such as /ns1/subdir) within the Amazon FSx file system that will be associated with the S3 data repository. The leading forward slash in the path is required. Two data repository associations cannot have overlapping file system paths. For example, if a data repository is associated with file system path /ns1, then you cannot link another data repository with file system path /ns1/ns2. The File system path setting must be unique across all the data repository associations for the file system.
  - Data repository path: Enter the path of an existing S3 bucket or prefix to associate with your file system (for example, s3://amzn-s3-demo-bucket/my-prefix). Two data repository associations cannot have overlapping data repository paths. The Data repository path setting must be unique across all the data repository associations for the file system.
  - Import metadata from repository: Select this property to optionally run an import data repository task to import metadata immediately after the link is created.
- 6. For Import settings optional, set an Import Policy that determines how your file and directory listings are kept up to date as you add, change, or delete objects in your S3 bucket. For example, choose New to import metadata to your file system for new objects created in the S3 bucket. For more information about import policies, see <u>Automatically import updates</u> from your S3 bucket.

7. For **Export policy**, set an export policy that determines how your files are exported to your linked S3 bucket as you add, change, or delete objects in your file system. For example, choose **Changed** to export objects whose content or metadata has been changed on your file system. For more information about export policies, see <u>Automatically export updates to your S3</u> bucket.

Choose Create.

### To link a file system to an S3 bucket (AWS CLI)

The following example creates a data repository association that links an Amazon FSx file system to an S3 bucket, with an import policy that imports any new or changed files to the file system and an export policy that exports new, changed, or deleted files to the linked S3 bucket.

• To create a data repository association, use the Amazon FSx CLI command create-data-repository-association, as shown following.

```
$ aws fsx create-data-repository-association \
     --file-system-id fs-0123456789abcdef0 \
     --file-system-path /ns1/path1/ \
     --data-repository-path s3://amzn-s3-demo-bucket/myprefix/ \
     --s3
"AutoImportPolicy={Events=[NEW,CHANGED,DELETED]},AutoExportPolicy={Events=[NEW,CHANGED,DELETED]}
```

Amazon FSx returns the JSON description of the DRA immediately. The DRA is created asynchronously.

You can use this command to create a data repository association even before the file system has finished creating. The request will be queued and the data repository association will be created after the file system is available.

# **Updating data repository association settings**

You can update an existing data repository association's settings using the AWS Management Console, the AWS CLI, and the Amazon FSx API, as shown in the following procedures.



#### Note

You cannot update the File system path or Data repository path of a DRA after it's created. If you want to change the File system path or Data repository path, you must delete the DRA and create it again.

### To update settings for an existing data repository association (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **File systems**, and then select the file system that you want to manage.
- Choose the **Data repository** tab.
- In the **Data repository associations** pane, choose the data repository association you want to change.
- Choose **Update**. An edit dialog displays for the data repository association. 5.
- For **Import settings optional**, you can update your **Import Policy**. For more information on import policies, see Automatically import updates from your S3 bucket.
- For **Export settings optional**, you can update your export policy. For more information on export policies, see Automatically export updates to your S3 bucket.
- Choose **Update**.

#### To update settings for an existing data repository association (CLI)

To update a data repository association, use the Amazon FSx CLI command update-datarepository-association, as shown following.

```
$ aws fsx update-data-repository-association \
      --association-id 'dra-872abab4b4503bfc2' \
      --s3
 "AutoImportPolicy={Events=[NEW,CHANGED,DELETED]},AutoExportPolicy={Events=[NEW,CHANGED,DEL
```

After successfully updating the data repository association's import and export policies, Amazon FSx returns the description of the updated data repository association as JSON.

# Deleting an association to an S3 bucket

The following procedures walk you through the process of deleting a data repository association from an existing Amazon FSx file system to an existing S3 bucket, using the AWS Management Console and AWS Command Line Interface (AWS CLI). Deleting the data repository association unlinks the file system from the S3 bucket.

### To delete a link from a file system to an S3 bucket (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **File systems** and then select the file system that you want to delete a data repository association from.
- Choose the **Data repository** tab.
- 4. In the **Data repository associations** pane, choose the data repository association that you want to delete.
- 5. For Actions, choose Delete association.
- 6. In the **Delete** dialog, you can choose **Delete data in file system** to physically delete the data in the file system that corresponds to the data repository association.
  - Choose this option if you plan to create a new data repository association using the same file system path but pointing to a different S3 bucket prefix, or if you no longer need the data in your file system.
- 7. Choose **Delete** to remove the data repository association from the file system.

### To delete a link from a file system to an S3 bucket (AWS CLI)

The following example deletes a data repository association that links an Amazon FSx file system to an S3 bucket. The --association-id parameter specifies the ID of the data repository association to be deleted.

 To delete a data repository association, use the Amazon FSx CLI command delete-datarepository-association, as shown following.

```
$ aws fsx delete-data-repository-association \
    --association-id dra-872abab4b4503bfc \
    --delete-data-in-file-system false
```

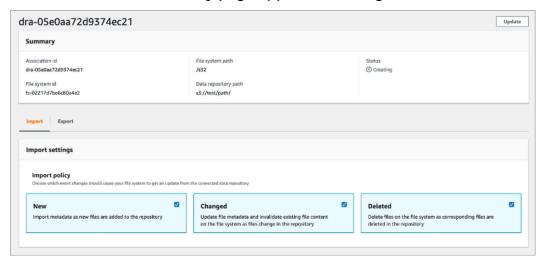
After successfully deleting the data repository association, Amazon FSx returns its description as JSON.

## Viewing data repository association details

You can view the details of a data repository association using the FSx for Lustre console, the AWS CLI, and the API. The details include the DRA's association ID, file system path, data repository path, import settings, export settings, status, and the ID of its associated file system.

### To view DRA details (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **File systems** and then select the file system that you want to view a data repository association's details for.
- 3. Choose the **Data repository** tab.
- 4. In the **Data repository associations** pane, choose the data repository association that you want to view. The **Summary** page appears, showing the DRA details.



### To view DRA details (CLI)

 To view the details of a specific data repository association, use the Amazon FSx CLI command describe-data-repository-associations, as shown following.

```
$ aws fsx describe-data-repository-associations \
    --association-ids dra-872abab4b4503bfc2
```

Amazon FSx returns the description of the data repository association as JSON.

# Data repository association lifecycle state

The data repository association lifecycle state provides status information about a specific DRA. A data repository association can have the following **Lifecycle states**:

- **Creating** Amazon FSx is creating the data repository association between the file system and the linked data repository. The data repository is unavailable.
- Available The data repository association is available for use.
- Updating The data repository association is undergoing a customer initiated update that might affect its availability.
- **Deleting** The data repository association is undergoing a customer initiated deletion.
- Misconfigured Amazon FSx cannot automatically import updates from the S3 bucket or automatically export updates to the S3 bucket until the data repository association configuration is corrected.

A DRA can become **Misconfigured** due to the following:

- Amazon FSx lacks necessary IAM permissions to access the S3 bucket.
- The FSx event notification configuration on the S3 bucket is deleted or modified.
- The S3 bucket has existing event notifications that overlap with FSx event types.

After resolving the underlying issue, the DRA automatically returns to the **Available** state within 15 minutes, or you can immediately trigger the state change using the AWS CLI command update-data-repository-association.

• **Failed** – The data repository association is in a terminal state that cannot be recovered (for example, because its file system path is deleted or the S3 bucket is deleted).

You can view a data repository association's lifecycle state using the Amazon FSx console, the AWS Command Line Interface, and the Amazon FSx API. For more information, see <u>Viewing data</u> repository association details.

## Working with server-side encrypted Amazon S3 buckets

FSx for Lustre supports Amazon S3 buckets that use server-side encryption with S3-managed keys (SSE-S3), and with AWS KMS keys stored in AWS Key Management Service (SSE-KMS).

If you want Amazon FSx to encrypt data when writing to your S3 bucket, you need to set the default encryption on your S3 bucket to either SSE-S3 or SSE-KMS. For more information, see

<u>Configuring default encryption</u> in the *Amazon S3 User Guide*. When writing files to your S3 bucket, Amazon FSx follows the default encryption policy of your S3 bucket.

By default, Amazon FSx supports S3 buckets encrypted using SSE-S3. If you want to link your Amazon FSx file system to an S3 bucket encrypted using SSE-KMS encryption, you need to add a statement to your customer managed key policy that allows Amazon FSx to encrypt and decrypt objects in your S3 bucket using your KMS key.

The following statement allows a specific Amazon FSx file system to encrypt and decrypt objects for a specific S3 bucket, <a href="mailto:bucket\_name">bucket\_name</a>.

```
{
    "Sid": "Allow access through S3 for the FSx SLR to use the KMS key on the objects
 in the given S3 bucket",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::aws_account_id:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/AWSServiceRoleForFSxS3Access_fsx_file_system_id"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:CallerAccount": "aws_account_id",
            "kms:ViaService": "s3.bucket-region.amazonaws.com"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket_name/*"
        }
    }
}
```

### Note

If you're using a KMS with a CMK to encrypt your S3 bucket with S3 bucket keys enabled, set the EncryptionContext to the bucket ARN, not the object ARN, as in this example:

```
"StringLike": {
    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket_name"
}
```

The following policy statement allows all Amazon FSx file systems in your account to link to a specific S3 bucket.

```
{
      "Sid": "Allow access through S3 for the FSx SLR to use the KMS key on the objects
 in the given S3 bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.bucket-region.amazonaws.com",
          "kms:CallerAccount": "aws_account_id"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3::::bucket_name/*"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws_partition:iam::aws_account_id:role/aws-service-
role/s3.data-source.lustre.fsx.amazonaws.com/AWSServiceRoleForFSxS3Access_fs-*"
        }
      }
}
```

# Accessing server-side encrypted Amazon S3 buckets in a different AWS account or from a Shared VPC

After you create an FSx for Lustre file system linked to an encrypted Amazon S3 bucket, you must then grant the AWSServiceRoleForFSxS3Access\_fs-01234567890 service-linked role (SLR) access to the KMS key used to encrypt the S3 bucket before reading or writing data from the linked S3 bucket. You can use an IAM role which already has permissions to the KMS key.



### Note

This IAM role must be in the account that the FSx for Lustre file system was created in (which is the same account as the S3 SLR), not the account that the KMS key/S3 bucket belong to.

You use the IAM role to call the following AWS KMS API to create a grant for the S3 SLR so that the SLR gains permission to the S3 objects. In order to find the ARN associated with your SLR, search your IAM roles using your file system ID as the search string.

```
$ aws kms create-grant --region fs_account_region \
      --key-id arn:aws:kms:s3_bucket_account_region:s3_bucket_account:key/key_id \
      --grantee-principal arn:aws:iam::fs_account_id:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/AWSServiceRoleForFSxS3Access_file-system-id \
      --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

For more information about service-linked roles, see Using service-linked roles for Amazon FSx.

# Importing changes from your data repository

You can import changes to data and POSIX metadata from a linked data repository to your Amazon FSx file system. Associated POSIX metadata includes ownership, permissions, and timestamps.

To import changes to the file system, use one of the following methods:

• Configure your file system to automatically import new, changed, or deleted files from your linked data repository. For more information, see Automatically import updates from your S3 bucket.

• Select the option to import metadata when you create a data repository association. This will initiate an import data repository task immediately after creating the data repository association.

 Use an on-demand import data repository task. For more information, see Using data repository tasks to import changes.

Automatic import and import data repository tasks can run at the same time.

When you turn on automatic import for a data repository association, your file system automatically updates file metadata as objects are created, modified, or deleted in S3. When you select the option to import metadata while creating a data repository association, your file system imports metadata for all objects in the data repository. When you import using an import data repository task, your file system imports only metadata for objects that were created or modified since the last import.

FSx for Lustre automatically copies the content of a file from your data repository and loads it into the file system when your application first accesses the file in the file system. This data movement is managed by FSx for Lustre and is transparent to your applications. Subsequent reads of these files are served directly from the file system with sub-millisecond latencies.

You can also preload your whole file system or a directory within your file system. For more information, see Preloading files into your file system. If you request the preloading of multiple files simultaneously, FSx for Lustre loads files from your Amazon S3 data repository in parallel.

FSx for Lustre only imports S3 objects that have POSIX-compliant object keys. Both automatic import and import data repository tasks import POSIX metadata. For more information, see POSIX metadata support for data repositories.



#### (i) Note

FSx for Lustre doesn't support importing metadata for symbolic links (symlinks) from S3 Glacier Flexible Retrieval and S3 Glacier Deep Archive storage classes. Metadata for S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive objects that aren't symlinks can be imported (that is, an inode is created on the FSx for Lustre file system with the correct metadata). However, to read this data from the file system, you must first restore the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive object. Importing file data directly from Amazon S3 objects in the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage class into FSx for Lustre isn't supported.

# Automatically import updates from your S3 bucket

You can configure FSx for Lustre to automatically update metadata in the file system as objects are added to, changed in, or deleted from your S3 bucket. FSx for Lustre creates, updates, or deletes the file and directory listing, corresponding to the change in S3. If the changed object in the S3 bucket no longer contains its metadata, FSx for Lustre maintains the current metadata values of the file, including the current permissions.



### Note

The FSx for Lustre file system and the linked S3 bucket must be located in the same AWS Region to automatically import updates.

You can configure automatic import when you create the data repository association, and you can update the automatic import settings at any time using the FSx management console, the AWS CLI, or the AWS API.



### Note

You can configure both automatic import and automatic export on the same data repository association. This topic describes only the automatic import feature.

### 

- If an object is modified in S3 with all automatic import policies enabled and automatic export disabled, the content of that object is always imported to a corresponding file in the file system. If a file already exists in the target location, the file is overwritten.
- If a file is modified in both the file system and S3, with all automatic import and automatic export policies enabled, either the file in the file system or the object in S3 could be overwritten by the other. It isn't guaranteed that a later edit in one location will overwrite an earlier edit in another location. If you modify the same file in both the file system and the S3 bucket, you should ensure application-level coordination to prevent such conflicts. FSx for Lustre doesn't prevent conflicting writes in multiple locations.

The import policy specifies how you want FSx for Lustre to update your file system as the contents change in the linked S3 bucket. A data repository association can have one of the following import policies:

- New FSx for Lustre automatically updates file and directory metadata only when new objects are added to the linked S3 data repository.
- **Changed** FSx for Lustre automatically updates file and directory metadata only when an existing object in the data repository is changed.
- **Deleted** FSx for Lustre automatically updates file and directory metadata only when an object in the data repository is deleted.
- Any combination of New, Changed, and Deleted FSx for Lustre automatically updates file
  and directory metadata when any of the specified actions occur in the S3 data repository. For
  example, you can specify that the file system is updated when an object is added to (New) or
  removed from (Deleted) the S3 repository, but not updated when an object is changed.
- No policy configured FSx for Lustre doesn't update file and directory metadata on the
  file system when objects are added to, changed in, or deleted from the S3 data repository.
  If you don't configure an import policy, automatic import is disabled for the data repository
  association. You can still manually import metadata changes by using an import data repository
  task, as described in Using data repository tasks to import changes.

### Important

Automatic import will not synchronize the following S3 actions with your linked FSx for Lustre file system:

- Deleting an object using S3 object lifecycle expirations
- Permanently deleting the current object version in a versioning-enabled bucket
- Undeleting an object in a versioning-enabled bucket

For most use cases, we recommend that you configure an import policy of **New**, **Changed**, and **Deleted**. This policy ensures that all updates made in your linked S3 data repository are automatically imported to your file system.

When you set an import policy to update your file system file and directory metadata based on changes in the linked S3 data repository, FSx for Lustre creates an event notification configuration

on the linked S3 bucket. The event notification configuration is named FSx. Don't modify or delete the FSx event notification configuration on the S3 bucket – doing so will prevent the automatic import of updated file and directory metadata to your file system.

When FSx for Lustre updates a file listing that has changed on the linked S3 data repository, it overwrites the local file with the updated version, even if the file is write-locked.

FSx for Lustre makes a best effort to update your file system. FSx for Lustre cannot update the file system in the following situations:

- If FSx for Lustre doesn't have permission to open the changed or new S3 object. In this case, FSx for Lustre skips the object and continues. The DRA lifecycle state isn't affected.
- If FSx for Lustre doesn't have bucket-level permissions, such as for GetBucketAc1. This will
  cause the data repository lifecycle state to become Misconfigured. For more information, see
  Data repository association lifecycle state.
- If the FSx event notification configuration on the linked S3 bucket is deleted or changed. This
  will cause the data repository lifecycle state to become Misconfigured. For more information,
  see Data repository association lifecycle state.

We recommend that you <u>turn on logging</u> to CloudWatch Logs to log information about any files or directories that couldn't be imported automatically. Warnings and errors in the log contain information about the failure reason. For more information, see <u>Data repository event logs</u>.

### **Prerequisites**

The following conditions are required for FSx for Lustre to automatically import new, changed, or deleted files from the linked S3 bucket:

- The file system and its linked S3 bucket are located in the same AWS Region.
- The S3 bucket doesn't have a misconfigured **Lifecycle state**. For more information, see <u>Data</u> repository association lifecycle state.
- Your account has the permissions required to configure and receive event notifications on the linked S3 bucket.

## Types of file changes supported

FSx for Lustre supports importing the following changes to files and directories that occur in the linked S3 bucket:

- · Changes to file contents.
- Changes to file or directory metadata.
- Changes to symlink target or metadata.

Deletions of files and directories. If you delete an object in the linked S3 bucket which
corresponds to a directory in the file system (that is, an object with a key name that ends with a
slash), FSx for Lustre deletes the corresponding directory on the file system only if it is empty.

## **Updating import settings**

You can set a file system's import settings for a linked S3 bucket when you create the data repository association. For more information, see Creating a link to an S3 bucket.

You can also update the import settings at any time, including the import policy. For more information, see Updating data repository association settings.

## Monitoring automatic import

If the rate of change in your S3 bucket exceeds the rate at which automatic import can process these changes, the corresponding metadata changes being imported to your FSx for Lustre file system are delayed. If this occurs, you can use the AgeOfOldestQueuedMessage metric to monitor the age of the oldest change waiting to be processed by automatic import. For more information on this metric, see FSx for Lustre S3 repository metrics.

If the delay in importing metadata changes exceeds 14 days (as measured using the AgeOfOldestQueuedMessage metric), changes in your S3 bucket that haven't been processed by automatic import aren't imported into your file system. Additionally, your data repository association lifecycle is marked as **MISCONFIGURED** and automatic import is stopped. If you have automatic export enabled, automatic export continues monitoring your FSx for Lustre file system for changes. However, additional changes aren't synchronized from your FSx for Lustre file system to S3.

To return your data repository association from the **MISCONFIGURED** lifecycle state to the **AVAILABLE** lifecycle state, you must update your data repository association. You can update your data repository association using the <u>update-data-repository-association</u> CLI command (or the corresponding <u>UpdateDataRepositoryAssociation</u> API operation). The only request parameter that you need is the AssociationID of the data repository association that you want to update.

After the data repository association lifecycle state changes to **AVAILABLE**, automatic import (and automatic export if enabled) restarts. Upon restarting, automatic export resumes synchronizing file system changes to S3. To synchronize the metadata of new and changed objects in S3 with your FSx for Lustre file system that weren't imported or are from when the data repository association was in a misconfigured state, run an <u>import data repository task</u>. Import data repository tasks don't synchronize deletes in your S3 bucket with your FSx for Lustre file system. If you want to fully synchronize S3 with your file system (including deletes), you must re-create your file system.

To ensure that delays to importing metadata changes don't exceed 14 days, we recommend that you set an alarm on the AgeOfOldestQueuedMessage metric and reduce activity in your S3 bucket if the AgeOfOldestQueuedMessage metric grows beyond your alarm threshold. For an FSx for Lustre file system connected to an S3 bucket with a single shard continuously sending the maximum number of possible changes from S3, with only automatic import running on the FSx for Lustre file system, automatic import can process a 7-hour backlog of S3 changes within 14 days.

Additionally, with a single S3 action, you can generate more changes than automatic import will ever process in 14 days. Examples of these types of actions include, but are not limited to, AWS Snowball uploads to S3 and large-scale deletions. If you make a large-scale change to your S3 bucket that you want synchronized with your FSx for Lustre file system, to prevent automatic import changes from exceeding 14 days, you should delete your file system and re-create it once the S3 change has completed.

If your AgeOfOldestQueuedMessage metric is growing, review your S3 bucket GetRequests, PutRequests, PostRequests, and DeleteRequests metrics for activity changes that would cause an increase in the rate and/or number of changes being sent to automatic import. For information about available S3 metrics, see Monitoring Amazon S3 in the Amazon S3 User Guide.

For a list of all available FSx for Lustre metrics, see Monitoring with Amazon CloudWatch.

# Using data repository tasks to import changes

The import data repository task imports metadata of objects that are new or changed in your S3 data repository, creating a new file or directory listing for any new object in the S3 data repository. For any object that has been changed in the data repository, the corresponding file or directory listing is updated with the new metadata. No action is taken for objects that have been deleted from the data repository.

Use the following procedures to import metadata changes by using the Amazon FSx console and CLI. Note that you can use one data repository task for multiple DRAs.

### To import metadata changes (console)

- Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/. 1.
- On the navigation pane, choose **File systems**, then choose your Lustre file system. 2.
- Choose the **Data repository** tab. 3.
- In the **Data repository associations** pane, choose the data repository associations you want to create the import task for.
- From the **Actions** menu, choose **Import task**. This choice isn't available if the file system isn't linked to a data repository. The **Create import data repository task** page appears.
- (Optional) Specify up to 32 directories or files to import from your linked S3 buckets by 6. providing the paths to those directories or files in **Data repository paths to import**.



### Note

If a path that you provide isn't valid, the task fails.

- 7. (Optional) Choose **Enable** under **Completion report** to generate a task completion report after the task completes. A task completion report provides details about the files processed by the task that meet the scope provided in **Report scope**. To specify the location for Amazon FSx to deliver the report, enter a relative path on a linked S3 data repository for **Report path**.
- Choose Create. 8.

A notification at the top of the **File systems** page shows the task that you just created in progress.

To view the task status and details, scroll down to the Data Repository Tasks pane in the Data **Repository** tab for the file system. The default sort order shows the most recent task at the top of the list.

To view a task summary from this page, choose **Task ID** for the task you just created. The **Summary** page for the task appears.

### To import metadata changes (CLI)

Use the create-data-repository-task CLI command to import metadata changes on your FSx for Lustre file system. The corresponding API operation is CreateDataRepositoryTask.

```
$ aws fsx create-data-repository-task \
    --file-system-id fs-0123456789abcdef0 \
    --type IMPORT_METADATA_FROM_REPOSITORY \
    --paths s3://bucketname1/dir1/path1 \
    --report Enabled=true,Path=s3://bucketname1/dir1/
path1,Format=REPORT_CSV_20191124,Scope=FAILED_FILES_ONLY
```

After successfully creating the data repository task, Amazon FSx returns the task description as JSON.

After creating the task to import metadata from the linked data repository, you can check the status of the import data repository task. For more information about viewing data repository tasks, see Accessing data repository tasks.

## Preloading files into your file system

You can optionally preload contents individual files or directories into your file system.

### Importing files using HSM commands

Amazon FSx copies data from your Amazon S3 data repository when a file is first accessed. Because of this approach, the initial read or write to a file incurs a small amount of latency. If your application is sensitive to this latency, and you know which files or directories your application needs to access, you can optionally preload contents of individual files or directories. You do so using the hsm\_restore command, as follows.

You can use the hsm\_action command (issued with the 1fs user utility) to verify that the file's contents have finished loading into the file system. A return value of NOOP indicates that the file has successfully been loaded. Run the following commands from a compute instance with the file system mounted. Replace <code>path/to/file</code> with the path of the file you're preloading into your file system.

```
sudo lfs hsm_restore path/to/file
sudo lfs hsm_action path/to/file
```

You can preload your whole file system or an entire directory within your file system by using the following commands. (The trailing ampersand makes a command run as a background process.) If you request the preloading of multiple files simultaneously, Amazon FSx loads your files from

your Amazon S3 data repository in parallel. If a file has already been loaded to the file system, the hsm restore command doesn't reload it.

```
nohup find local/directory -type f -print0 | xargs -0 -n 1 -P 8 sudo lfs hsm_restore &
```



### Note

If your linked S3 bucket is larger than your file system, you should be able to import all the file metadata into your file system. However, you can load only as much actual file data as will fit into the file system's remaining storage space. You'll receive an error if you attempt to access file data when there is no more storage left on the file system. If this occurs, you can increase the amount of storage capacity as needed. For more information, see Managing storage capacity.

## Validation step

You can run the bash script listed below to help you discover how many files or objects are in an archived (released) state.

To improve the script's performance, especially across file systems with a large number of files, CPU threads are automatically determined based in the /proc/cpuproc file. That is, you will see faster performance with a higher vCPU count Amazon EC2 instance.

Set up the bash script.

```
#!/bin/bash
# Check if a directory argument is provided
if [ $# -ne 1 ]; then
    echo "Usage: $0 /path/to/lustre/mount"
    exit 1
fi
# Set the root directory from the argument
ROOT_DIR="$1"
# Check if the provided directory exists
if [ ! -d "$ROOT_DIR" ]; then
    echo "Error: Directory $ROOT_DIR does not exist."
```

```
exit 1
fi
# Automatically detect number of CPUs and set threads
if command -v nproc &> /dev/null; then
    THREADS=$(nproc)
elif [ -f /proc/cpuinfo ]; then
    THREADS=$(grep -c ^processor /proc/cpuinfo)
else
    echo "Unable to determine number of CPUs. Defaulting to 1 thread."
    THREADS=1
fi
# Output file
OUTPUT_FILE="released_objects_$(date +%Y%m%d_%H%M%S).txt"
echo "Searching in $ROOT_DIR for all released objects using $THREADS threads"
echo "This may take a while depending on the size of the filesystem..."
# Find all released files in the specified lustre directory using parallel
# If you get false positives for file names/paths that include the word
 'released',
# you can grep 'released exists archived' instead of just 'released'
time sudo lfs find "$ROOT_DIR" -type f | \
parallel --will-cite -j "$THREADS" -n 1000 "sudo lfs hsm_state {} | grep released"
> "$OUTPUT_FILE"
echo "Search complete. Released objects are listed in $OUTPUT_FILE"
echo "Total number of released objects: $(wc -1 <"$OUTPUT_FILE")"
```

2. Make the script executable:

```
$ chmod +x find_lustre_released_files.sh
```

3. Run the script, as in the following example:

```
$ ./find_lustre_released_files.sh /fsxl/sample
Searching in /fsxl/sample for all released objects using 16 threads
This may take a while depending on the size of the filesystem...
real 0m9.906s
user 0m1.502s
sys 0m5.653s
```

```
Search complete. Released objects are listed in
released_objects_20241121_184537.txt
Total number of released objects: 30000
```

If there are released objects present, then perform a bulk restore on the desired directories to bring the files into FSx for Lustre from S3, as in the following example:

```
$ DIR=/path/to/lustre/mount
$ nohup find $DIR -type f -print0 | xargs -0 -n 1 -P 8 sudo lfs hsm_restore &
```

Note that hsm restore will take a while where there are millions of files.

# **Exporting changes to the data repository**

You can export changes to data and POSIX metadata changes from your FSx for Lustre file system to a linked data repository. Associated POSIX metadata includes ownership, permissions, and timestamps.

To export changes from the file system, use one of the following methods.

- Configure your file system to automatically export new, changed, or deleted files to your linked data repository. For more information, see Automatically export updates to your S3 bucket.
- Use an on-demand export data repository task. For more information, see Using data repository tasks to export changes

Automatic export and export data repository tasks cannot run at the same time.

#### Important

Automatic export will not synchronize the following metadata operations on your file system with S3 if the corresponding objects are stored in S3 Glacier Flexible Retrieval:

- chmod
- chown
- rename

When you turn on automatic export for a data repository association, your file system automatically exports file data and metadata changes as files are created, modified, or deleted. When you export files or directories using an export data repository task, your file system exports only data files and metadata that were created or modified since the last export.

Both automatic export and export data repository tasks export POSIX metadata. For more information, see POSIX metadata support for data repositories.

### ▲ Important

- To ensure that FSx for Lustre can export your data to your S3 bucket, it must be stored in a UTF-8 compatible format.
- S3 object keys have a maximum length of 1,024 bytes. FSx for Lustre will not export files whose corresponding S3 object key would be longer than 1,024 bytes.

### Note

All objects created by automatic export and export data repository tasks are written using the S3 Standard storage class.

### **Topics**

- Automatically export updates to your S3 bucket
- Using data repository tasks to export changes
- Exporting files using HSM commands

### Automatically export updates to your S3 bucket

You can configure your FSx for Lustre file system to automatically update the contents of a linked S3 bucket as files are added, changed, or deleted on the file system. FSx for Lustre creates, updates, or deletes the object in S3, corresponding to the change in the file system.



#### Note

Automatic export isn't available on FSx for Lustre 2.10 file systems or Scratch 1 file systems.

You can export to a data repository that is in the same AWS Region as the file system or in a different AWS Region.

You can configure automatic export when you create the data repository association and update the automatic export settings at any time using the FSx management console, the AWS CLI, and the AWS API.

#### 

- If a file is modified in the file system with all automatic export policies enabled and automatic import disabled, the content of that file is always exported to a corresponding object in S3. If an object already exists in the target location, the object is overwritten.
- If a file is modified in both the file system and S3, with all automatic import and automatic export policies enabled, either the file in the file system or the object in S3 could be overwritten by the other. It isn't guaranteed that a later edit in one location will overwrite an earlier edit in another location. If you modify the same file in both the file system and the S3 bucket, you should ensure application-level coordination to prevent such conflicts. FSx for Lustre doesn't prevent conflicting writes in multiple locations.

The export policy specifies how you want FSx for Lustre to update your linked S3 bucket as the contents change in the file system. A data repository association can have one of the following automatic export policies:

- New FSx for Lustre automatically updates the S3 data repository only when a new file, directory, or symlink is created on the file system.
- Changed FSx for Lustre automatically updates the S3 data repository only when an existing file in the file system is changed. For file content changes, the file must be closed before it's propagated to the S3 repository. Metadata changes (rename, ownership, permissions, and timestamps) are propagated when the operation is done. For renaming changes (including

moves), the existing (pre-renamed) S3 object is deleted and a new S3 object is created with the new name.

 Deleted – FSx for Lustre automatically updates the S3 data repository only when a file, directory, or symlink is deleted in the file system.

- Any combination of New, Changed, and Deleted FSx for Lustre automatically updates the S3 data repository when any of the specified actions occur in file system. For example, you can specify that the S3 repository is updated when a file is added to (New) or removed from (**Deleted**) the file system, but not when a file is changed.
- No policy configured FSx for Lustre doesn't automatically update the S3 data repository when files are added to, changed in, or deleted from the file system. If you don't configure an export policy, automatic export is disabled. You can still manually export changes by using an export data repository task, as described in Using data repository tasks to export changes.

For most use cases, we recommend that you configure an export policy of New, Changed, and **Deleted.** This policy ensures that all updates made on your file system are automatically exported to your linked S3 data repository.

We recommend that you turn on logging to CloudWatch Logs to log information about any files or directories that couldn't be exported automatically. Warnings and errors in the log contain information about the failure reason. For more information, see Data repository event logs.



#### Note

While access time (atime) and modification time (mtime) are synchronized with S3 during export operations, changes to these timestamps alone do not trigger automatic export. Only changes to file content or other metadata (such as ownership or permissions) will trigger an automatic export to S3.

### **Updating export settings**

You can set a file system's export settings to a linked S3 bucket when you create the data repository association. For more information, see Creating a link to an S3 bucket.

You can also update the export settings at any time, including the export policy. For more information, see Updating data repository association settings.

### Monitoring automatic export

You can monitor automatic export enabled data repository associations using a set of metrics published to Amazon CloudWatch. The AgeOfOldestQueuedMessage metric represents the age of the oldest update made to the file system which has not yet been exported to S3. If the AgeOfOldestQueuedMessage is greater than zero for an extended period of time, we recommend temporarily reducing the number of changes (directory renames in particular) that are actively being made to the file system until the message queue has been reduced. For more information, see FSx for Lustre S3 repository metrics.

#### 

When deleting a data repository association or file system with automatic export enabled, you should first make sure that AgeOfOldestQueuedMessage is zero, meaning that there are no changes that have not yet been exported. If AgeOfOldestQueuedMessage is greater than zero when you delete your data repository association or file system, the changes that had not yet been exported will not reach your linked S3 bucket. To avoid this, wait for AgeOfOldestQueuedMessage to reach zero before deleting your data repository association or file system.

## Using data repository tasks to export changes

The export data repository task exports files that are new or changed in your file system. It creates a new object in S3 for any new file on the file system. For any file that has been modified on the file system, or whose metadata has been modified, the corresponding object in S3 is replaced with a new object with the new data and metadata. No action is taken for files that have been deleted from the file system.



#### Note

Keep the following in mind when using export data repository tasks:

- The use of wildcards to include or exclude files for export isn't supported.
- When performing mv operations, the target file after being moved will be exported to S3 even if there is no UID, GID, permission, or content change.

Use the following procedures to export data and metadata changes on the file system to linked S3 buckets by using the Amazon FSx console and CLI. Note that you can use one data repository task for multiple DRAs.

#### To export changes (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. On the navigation pane, choose **File systems**, then choose your Lustre file system.
- 3. Choose the **Data repository** tab.
- In the **Data repository associations** pane, choose the data repository association you want to 4. create the export task for.
- For **Actions**, choose **Export task**. This choice isn't available if the file system isn't linked to a data repository on S3. The Create export data repository task dialog appears.
- (Optional) Specify up to 32 directories or files to export from your Amazon FSx file system by 6. providing the paths to those directories or files in File system paths to export. The paths you provide need to be relative to the mount point of the file system. If the mount point is /mnt/ fsx and /mnt/fsx/path1 is a directory or file on the file system you want to export, then the path to provide is path1.



#### Note

If a path that you provide isn't valid, the task fails.

- (Optional) Choose Enable under Completion report to generate a task completion report after 7. the task completes. A task completion report provides details about the files processed by the task that meet the scope provided in **Report scope**. To specify the location for Amazon FSx to deliver the report, enter a relative path on the file system's linked S3 data repository for Report path.
- Choose Create.

A notification at the top of the **File systems** page shows the task that you just created in progress.

To view the task status and details, scroll down to the Data Repository Tasks pane in the Data **Repository** tab for the file system. The default sort order shows the most recent task at the top of the list.

To view a task summary from this page, choose **Task ID** for the task you just created. The **Summary** page for the task appears.

#### To export changes (CLI)

• Use the <u>create-data-repository-task</u> CLI command to export data and metadata changes on your FSx for Lustre file system. The corresponding API operation is CreateDataRepositoryTask.

```
$ aws fsx create-data-repository-task \
    --file-system-id fs-0123456789abcdef0 \
    --type EXPORT_TO_REPOSITORY \
    --paths path1,path2/file1 \
    --report Enabled=true
```

After successfully creating the data repository task, Amazon FSx returns the task description as JSON, as shown in the following example.

```
{
    "Task": {
        "TaskId": "task-123f8cd8e330c1321",
        "Type": "EXPORT_TO_REPOSITORY",
        "Lifecycle": "PENDING",
        "FileSystemId": "fs-0123456789abcdef0",
        "Paths": ["path1", "path2/file1"],
        "Report": {
            "Path": "s3://dataset-01/reports",
            "Format": "REPORT_CSV_20191124",
            "Enabled":true,
            "Scope": "FAILED_FILES_ONLY"
        },
        "CreationTime": "1545070680.120",
        "ClientRequestToken": "10192019-drt-12",
        "ResourceARN": "arn:aws:fsx:us-
east-1:123456789012:task:task-123f8cd8e330c1321"
    }
}
```

After creating the task to export data to the linked data repository, you can check the status of the export data repository task. For more information about viewing data repository tasks, see Accessing data repository tasks.

### **Exporting files using HSM commands**



#### Note

To export changes in your FSx for Lustre file system's data and metadata to a durable data repository on Amazon S3, use the automatic export feature described in Automatically export updates to your S3 bucket. You can also use export data repository tasks, described in Using data repository tasks to export changes.

To export an individual file to your data repository and verify that the file has successfully been exported to your data repository, you can run the commands shown following. A return value of states: (0x00000009) exists archived indicates that the file has successfully been exported.

```
sudo lfs hsm_archive path/to/export/file
sudo lfs hsm_state path/to/export/file
```



#### Note

You must run the HSM commands (such as hsm\_archive) as the root user or using sudo.

To export your entire file system or an entire directory in your file system, run the following commands. If you export multiple files simultaneously, Amazon FSx for Lustre exports your files to your Amazon S3 data repository in parallel.

```
nohup find local/directory -type f -print0 | xargs -0 -n 1 sudo lfs hsm_archive &
```

To determine whether the export has completed, run the following command.

```
find path/to/export/file -type f -print0 | xargs -0 -n 1 -P 8 sudo lfs hsm_state | awk
 '!/\<archived\>/ || /\<dirty\>/' | wc -l
```

If the command returns with zero files remaining, the export is complete.

# **Data repository tasks**

By using import and export data repository tasks, you can manage the transfer of data and metadata between your FSx for Lustre file system and any of its durable data repositories on Amazon S3.

Data repository tasks optimize data and metadata transfers between your FSx for Lustre file system and a data repository on S3. One way that they do this is by tracking changes between your Amazon FSx file system and its linked data repository. They also do this by using parallel transfer techniques to transfer data at speeds up to hundreds of GBps. You create and view data repository tasks using the Amazon FSx console, the AWS CLI, and the Amazon FSx API.

Data repository tasks maintain the file system's Portable Operating System Interface (POSIX) metadata, including ownership, permissions, and timestamps. Because the tasks maintain this metadata, you can implement and maintain access controls between your FSx for Lustre file system and its linked data repositories.

You can use a release data repository task to free up file system space for new files by releasing files exported to Amazon S3. The released file's content is removed, but the metadata of the released file remains on the file system. Users and applications can still access a released file by reading the file again. When the user or application reads the released file, FSx for Lustre transparently retrieves the file content from Amazon S3.

## Types of data repository tasks

There are three types of data repository tasks:

- Export data repository tasks export from your Lustre file system to a linked S3 bucket.
- Import data repository tasks import from a linked S3 bucket to your Lustre file system.
- Release data repository tasks release files exported to a linked S3 bucket from your Lustre file system.

For more information, see Creating a data repository task.

#### **Topics**

Data repository tasks 70

- Understanding a task's status and details
- Using data repository tasks
- Working with task completion reports
- Troubleshooting data repository task failures

### Understanding a task's status and details

A data repository task has descriptive information and a lifecycle status.

After a task is created, you can view the following detailed information for a data repository task using the Amazon FSx console, CLI, or API:

- The task type:
  - EXPORT\_TO\_REPOSITORY indicates an export task.
  - IMPORT\_METADATA\_FROM\_REPOSITORY indicates an import task.
  - RELEASE\_DATA\_FROM\_FILESYSTEM indicates a release task.
- The file system that the task ran on.
- The task creation time.
- The task status.
- The total number of files that the task processed.
- The total number of files that the task successfully processed.
- The total number of files that the task failed to process. This value is greater than zero when the task status is FAILED. Detailed information about files that failed is available in a task completion report. For more information, see Working with task completion reports.
- The time that the task started.
- The time that the task status was last updated. Task status is updated every 30 seconds.

A data repository task can have one of the following statuses:

- **PENDING** indicates that Amazon FSx has not started the task.
- **EXECUTING** indicates that Amazon FSx is processing the task.
- **FAILED** indicates that Amazon FSx didn't successfully process the task. For example, there might be files that the task failed to process. The task details provide more information about

Task status and details 71

the failure. For more information about failed tasks, see <u>Troubleshooting data repository task</u> failures.

- SUCCEEDED indicates that Amazon FSx completed the task successfully.
- CANCELED indicates that the task was canceled and not completed.
- **CANCELING** indicates that Amazon FSx is in the process of canceling the task.

For more information about accessing existing data repository tasks, see <u>Accessing data repository</u> tasks.

### Using data repository tasks

In the following sections, you can find detailed information about managing data repository tasks. You can create, duplicate, view details, and cancel data repository tasks using the Amazon FSx console, CLI, or API.

#### **Topics**

- Creating a data repository task
- Duplicating a task
- · Accessing data repository tasks
- Canceling a data repository task

### Creating a data repository task

You can create a data repository task by using the Amazon FSx console, CLI, or API. After you create a task, you can view the task's progress and status by using the console, CLI, or API.

You can create three types of data repository tasks:

- The **Export** data repository task exports from your Lustre file system to a linked S3 bucket. For more information, see Using data repository tasks to export changes.
- The **Import** data repository task imports from a linked S3 bucket to your Lustre file system. For more information, see Using data repository tasks to import changes.
- The Release data repository task releases files from your Lustre file system that have been
  exported to a linked S3 bucket. For more information, see <u>Using data repository tasks to release</u>
  files.

### **Duplicating a task**

You can duplicate an existing data repository task in the Amazon FSx console. When you duplicate a task, an exact copy of the existing task is displayed in the Create import data repository task or Create export data repository task page. You can make changes to the paths to export or import, as needed, before creating and running the new task.

#### Note

A request to run a duplicate task will fail if an exact copy of that task is already running. An exact copy of a task that is already running contains the same file system path or paths in the case of an export task or the same data repository paths in the case of an import task.

You can duplicate a task from the task details view, the **Data Repository Tasks** pane in the **Data Repository** tab for the file system, or from the **Data repository tasks** page.

#### To duplicate an existing task

- Choose a task on the **Data Repository Tasks** pane in the **Data Repository** tab for the file system.
- Choose **Duplicate task**. Depending on which type of task you chose, the **Create import data** repository task or Create export data repository task page appears. All settings for the new task are identical to those for the task that you're duplicating.
- Change or add the paths that you want to import from or export to.
- Choose Create.

### Accessing data repository tasks

After you create a data repository task, you can access the task, and all existing tasks in your account, using the Amazon FSx console, CLI, and API. Amazon FSx provides the following detailed task information:

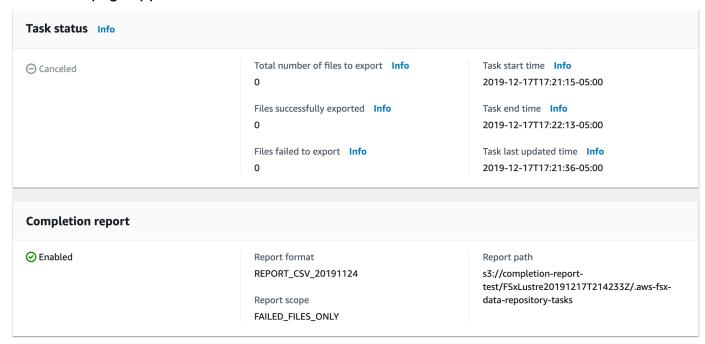
- All existing tasks.
- All tasks for a specific file system.
- All tasks for a specific data repository association.

 All tasks with a specific lifecycle status. For more information about task lifecycle status values, see Understanding a task's status and details.

You can access all existing data repository tasks in your account by using the Amazon FSx console, CLI, or API, as described following.

#### To view data repository tasks and task details (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. On the navigation pane, choose the file system that you want to view data repository tasks for. The file system details page appears.
- 3. On the file system details page, choose the **Data repository** tab. Any tasks for this file system appear on the **Data repository tasks** panel.
- 4. To see a task's details, choose **Task ID** or **Task name** in the **Data repository tasks** panel. The task detail page appears.



#### To retrieve data repository tasks and task details (CLI)

Using the Amazon FSx <u>describe-data-repository-tasks</u> CLI command, you can view all the data repository tasks, and their details, in your account. <u>DescribeDataRepositoryTasks</u> is the equivalent API command.

• Use the following command to view all data repository task objects in your account.

```
aws fsx describe-data-repository-tasks
```

If the command is successful, Amazon FSx returns the response in JSON format.

```
{
    "DataRepositoryTasks": [
        {
            "Lifecycle": "EXECUTING",
            "Paths": [],
            "Report": {
                "Path": "s3://dataset-01/reports",
                "Format": "REPORT_CSV_20191124",
                "Enabled":true,
                "Scope": "FAILED_FILES_ONLY"
            },
            "StartTime": 1591863862.288,
            "EndTime": ,
            "Type": "EXPORT_TO_REPOSITORY",
            "Tags": [],
            "TaskId": "task-0123456789abcdef3",
            "Status": {
                "SucceededCount": 4255,
                "TotalCount": 4200,
                "FailedCount": 55,
                "LastUpdatedTime": 1571863875.289
            },
            "FileSystemId": "fs-0123456789a7",
            "CreationTime": 1571863850.075,
            "ResourceARN": "arn:aws:fsx:us-east-1:1234567890:task/
task-0123456789abcdef3"
        },
        {
            "Lifecycle": "FAILED",
            "Paths": [],
            "Report": {
                "Enabled": false,
            },
            "StartTime": 1571863862.288,
            "EndTime": 1571863905.292,
            "Type": "EXPORT_TO_REPOSITORY",
            "Tags": [],
            "TaskId": "task-0123456789abcdef1",
```

```
"Status": {
                "SucceededCount": 1153,
                "TotalCount": 1156,
                "FailedCount": 3,
                "LastUpdatedTime": 1571863875.289
            },
            "FileSystemId": "fs-0123456789abcdef0",
            "CreationTime": 1571863850.075,
            "ResourceARN": "arn:aws:fsx:us-east-1:1234567890:task/
task-0123456789abcdef1"
        },
        {
            "Lifecycle": "SUCCEEDED",
            "Paths": [],
            "Report": {
                "Path": "s3://dataset-04/reports",
                "Format": "REPORT_CSV_20191124",
                "Enabled":true,
                "Scope": "FAILED_FILES_ONLY"
            },
            "StartTime": 1571863862.288,
            "EndTime": 1571863905.292,
            "Type": "EXPORT_TO_REPOSITORY",
            "Tags": [],
            "TaskId": "task-04299453935122318",
            "Status": {
                "SucceededCount": 258,
                "TotalCount": 258,
                "FailedCount": 0,
                "LastUpdatedTime": 1771848950.012,
            },
            "FileSystemId": "fs-0123456789abcdef0",
            "CreationTime": 1771848950.012,
            "ResourceARN": "arn:aws:fsx:us-east-1:1234567890:task/
task-0123456789abcdef0"
        }
    ]
}
```

#### Viewing tasks by file system

You can view all tasks for a specific file system using the Amazon FSx console, CLI, or API, as described following.

#### To view tasks by file system (console)

- 1. Choose **File systems** on the navigation pane. The **File systems** page appears.
- 2. Choose the file system that you want to view data repository tasks for. The file system details page appears.
- 3. On the file system details page, choose the **Data repository** tab. Any tasks for this file system appear on the **Data repository tasks** panel.

#### To retrieve tasks by file system (CLI)

• Use the following command to view all data repository tasks for file system fs-0123456789abcdef0.

```
aws fsx describe-data-repository-tasks \
    --filters Name=file-system-id, Values=fs-0123456789abcdef0
```

If the command is successful, Amazon FSx returns the response in JSON format.

```
{
    "DataRepositoryTasks": [
        {
            "Lifecycle": "FAILED",
            "Paths": [],
            "Report": {
                "Path": "s3://dataset-04/reports",
                "Format": "REPORT_CSV_20191124",
                "Enabled":true,
                "Scope": "FAILED_FILES_ONLY"
            },
            "StartTime": 1571863862.288,
            "EndTime": 1571863905.292,
            "Type": "EXPORT_TO_REPOSITORY",
            "Tags": [],
            "TaskId": "task-0123456789abcdef1",
            "Status": {
                "SucceededCount": 1153,
```

```
"TotalCount": 1156,
                "FailedCount": 3,
                "LastUpdatedTime": 1571863875.289
            },
            "FileSystemId": "fs-0123456789abcdef0",
            "CreationTime": 1571863850.075,
            "ResourceARN": "arn:aws:fsx:us-east-1:1234567890:task/
task-0123456789abcdef1"
        },
        {
            "Lifecycle": "SUCCEEDED",
            "Paths": [],
            "Report": {
                "Enabled": false,
            },
            "StartTime": 1571863862.288,
            "EndTime": 1571863905.292,
            "Type": "EXPORT_TO_REPOSITORY",
            "Tags": [],
            "TaskId": "task-0123456789abcdef0",
            "Status": {
                "SucceededCount": 258,
                "TotalCount": 258,
                "FailedCount": 0,
                "LastUpdatedTime": 1771848950.012,
            },
            "FileSystemId": "fs-0123456789abcdef0",
            "CreationTime": 1771848950.012,
            "ResourceARN": "arn:aws:fsx:us-east-1:1234567890:task/
task-0123456789abcdef0"
        }
    ]
}
```

### Canceling a data repository task

You can cancel a data repository task while it's in either the PENDING or EXECUTING state. When you cancel a task, the following occurs:

- Amazon FSx doesn't process any files that are in the queue to be processed.
- Amazon FSx continues processing any files that are currently in process.

• Amazon FSx doesn't revert any files that the task already processed.

#### To cancel a data repository task (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Click on the file system for which you want to cancel a data repository task.
- 3. Open the **Data Repository** tab and scroll down to view the **Data Repository Tasks** panel.
- 4. Choose **Task ID** or **Task name** for the task that you want to cancel.
- 5. Choose Cancel task to cancel the task.
- 6. Enter the task ID to confirm the cancellation request.

#### To cancel a data repository task (CLI)

Use the Amazon FSx <u>cancel-data-repository-task</u> CLI command, to cancel a task. CancelDataRepositoryTask is the equivalent API command.

• Use the following command to cancel a data repository task.

```
aws fsx cancel-data-repository-task \
--task-id task-0123456789abcdef0
```

If the command is successful, Amazon FSx returns the response in JSON format.

```
{
    "Status": "CANCELING",
    "TaskId": "task-0123456789abcdef0"
}
```

### Working with task completion reports

A task completion report provides details about the results of an export, import, or release data repository task. The report includes results for the files processed by the task that match the scope of the report. You can specify whether to generate a report for a task by using the Enabled parameter.

Amazon FSx delivers the report to the file system's linked data repository in Amazon S3, using the path that you specify when you enable the report for a task. The report's file name is report.csv for import tasks and failures.csv for export or release tasks.

The report format is a comma-separated value (CSV) file that has three fields: FilePath, FileStatus, and ErrorCode.

Reports are encoded using RFC-4180-format encoding as follows:

- Paths starting with any of the following characters are contained in single quotation marks: @ +
   =
- Strings that contain at least one of the following characters are contained in double quotation marks: " ,
- All double quotation marks are escaped with an additional double quotation mark.

Following are a few examples of the report encoding:

- @filename.txt becomes """@filename.txt"""
- +filename.txt becomes """+filename.txt"""
- file, name.txt becomes "file, name.txt"
- file"name.txt becomes "file""name.txt"

For more information about RFC-4180 encoding, see <a href="RFC-4180"><u>RFC-4180 - Common Format and MIME Type</u></a> for Comma-Separated Values (CSV) Files on the IETF website.

The following is an example of the information provided in a task completion report that includes only failed files.

```
myRestrictedFile,failed,S3AccessDenied
dir1/myLargeFile,failed,FileSizeTooLarge
dir2/anotherLargeFile,failed,FileSizeTooLarge
```

For more information about task failures and how to resolve them, see <u>Troubleshooting data</u> repository task failures.

# Troubleshooting data repository task failures

You can <u>turn on logging</u> to CloudWatch Logs to log information about any failures experienced while importing or exporting files using data repository tasks. For information about CloudWatch Logs event logs, see Data repository event logs.

When a data repository task fails, you can find the number of files that Amazon FSx failed to process in **Files failed to export** on the console's **Task status** page. Or you can use the CLI or API and view the task's Status: FailedCount property. For information about accessing this information, see Accessing data repository tasks.

For data repository tasks, Amazon FSx also optionally provides information about the specific files and directories that failed in a completion report. The task completion report contains the file or directory path on the Lustre file system that failed, its status, and the failure reason. For more information, see Working with task completion reports.

A data repository task can fail for several reasons, including those listed following.

| Error Code            | Explanation   |
|-----------------------|---|
| FileSizeTooLarge      | The maximum object size supported by Amazon S3 is 5 TiB.  |
| InternalError         | An error occurred within the Amazon FSx file system for an import, export, or release task. Generally, this error code means that the Amazon FSx file system that the failed task ran on is in a FAILED lifecycle state. When this occurs, the affected files might not be recoverable due to data loss. Otherwise, you can use hierarchical storage managemen t (HSM) commands to export the files and directories to the data repository on S3. For more information, see <a href="Exporting files using HSM commands">Exporting files using HSM commands</a> . |
| OperationNotPermitted | Amazon FSx was unable to release the file<br>because it ihas not been exported to a linked<br>S3 bucket. You must use automatic export or   |

Troubleshooting task failures

| Error Code      | Explanation  |
|-----------------|--|
|                 | export data repository tasks to ensure that your files are first exported to your linked Amazon S3 bucket.   |
| PathSizeTooLong | The export path is too long. The maximum object key length supported by S3 is 1,024 characters.  |
| ResourceBusy    | Amazon FSx was unable to export or release the file because it was being accessed by another client on the file system. You can retry the DataRepositoryTask after your workflow has finished writing to the file. |

| Error Code     | Explanation  |
|----------------|--|
| S3AccessDenied | Access was denied to Amazon S3 for a data repository export or import task.  |
|                | For export tasks, the Amazon FSx file system must have permission to perform the S3:PutObject operation to export to a linked data repository on S3. This permissio n is granted in the AWSServiceRoleForF SxS3Access_ fs-0123456789abcdef0 service-linked role. For more information, see Using service-linked roles for Amazon FSx.  For export tasks, because the export task requires data to flow outside a file system's VPC, this error can occur if the target repositor y has a bucket policy that contains one of the aws:SourceVpc or aws:SourceVpce IAM global condition keys. |
|                | For import tasks, the Amazon FSx file system must have permission to perform the S3:HeadObject and S3:GetObject operations to import from a linked data repository on S3.  |
|                | For import tasks, if your S3 bucket uses server-side encryption with customer managed keys stored in AWS Key Management Service (SSE-KMS), you must follow the policy configurations in Working with server-side encrypted Amazon S3 buckets.  |
|                | If your S3 bucket contains objects uploaded from a different AWS account than your file system linked S3 bucket account, you can ensure that your data repository tasks can  |

| Error Code                | Explanation   |
|---------------------------|---|
|                           | modify S3 metadata or overwrite S3 objects regardless of which account uploaded them. We recommend that you enable the S3 Object Ownership feature for your S3 bucket. This feature enables you to take ownership of new objects that other AWS accounts upload to your bucket, by forcing uploads to provide the -/-acl bucket-owner-full-control canned ACL. You enable S3 Object Ownership by choosing the <b>Bucket owner preferred</b> option in your S3 bucket. For more informati on, see Controlling ownership of uploaded objects using S3 Object Ownership in the Amazon S3 User Guide. |
| S3Error                   | Amazon FSx encountered an S3-related error that wasn't S3AccessDenied .   |
| S3FileDeleted             | Amazon FSx was unable to export a hard link file because the source file doesn't exist in the data repository.  |
| S30bjectInUnsupportedTier | Amazon FSx successfully imported a non- symlink object from an S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage class. The FileStatus will be succeeded with warning in the task completion report. The warning indicates that to retrieve the data, you must restore the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive object first and then use an hsm_restore command to import the object.   |
| S30bjectNotFound          | Amazon FSx was unable to import or export the file because it doesn't exist in the data repository.   |

| Error Code                                 | Explanation  |
|--|--|
| S30bjectPathNotPosixCompliant              | The Amazon S3 object exists but can't be imported because it isn't a POSIX-compliant object. For information about supported POSIX metadata, see POSIX metadata support for data repositories.   |
| S30bjectUpdateInProgressFro<br>mFileRename | Amazon FSx was unable to release the file because automatic export is processing a rename of the file. The automatic export rename process must finish before the file can be released.  |
| S3SymlinkInUnsupportedTier                 | Amazon FSx was unable to import a symlink object because it's in an Amazon S3 storage class that is not supported, such as an S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage class. The FileStatus will be failed in the task completion report. |
| SourceObjectDeletedBeforeRe leasing        | Amazon FSx was unable to release the file from the file system because the file was deleted from the data repository before it could be released.  |

# **Releasing files**

Release data repository tasks release file data from your FSx for Lustre file system to free up space for new files. Releasing a file retains the file listing and metadata, but removes the local copy of that file's contents. If a user or application accesses a released file, the data is automatically and transparently loaded back onto your file system from your linked Amazon S3 bucket.



### Note

Release data repository tasks are not available on FSx for Lustre 2.10 file systems.

Releasing files 85

The parameters File system paths to release and Minimum duration since last access determine which files will be released.

- **File system paths to release**: Specifies the path from which files will be released.
- Minimum duration since last access: Specifies the duration, in days, such that any file not accessed in that duration should be released. The duration since a file was last accessed is calculated by taking the difference between the release task create time and the last time a file was accessed (maximum value of atime, mtime, and ctime).

Files will only be released along the file path if they have been exported to S3 and have a duration since last access that is greater than the minimum duration since last access value. Providing a minimum duration since last access of 0 days will release files independent of their duration since last access.



#### Note

The use of wildcards to include or exclude files for release is not supported.

Release data repository tasks will only release data from files that have already been exported to a linked S3 data repository. You can export data to S3 using either the automatic export feature, an export data repository task, or HSM commands. To verify that a file has been exported to your data repository, you can run the following command. A return value of states: (0x00000009)exists archived indicates that the file has successfully been exported.

sudo lfs hsm\_state path/to/export/file



#### Note

You must run the HSM command as the root user or using sudo.

To release file data on a regular interval, you can schedule a recurring release data repository task using Amazon EventBridge Scheduler. For more information, see Getting started with EventBridge Scheduler in the Amazon EventBridge Scheduler User Guide.

#### **Topics**

Releasing files

Using data repository tasks to release files

### Using data repository tasks to release files

Use the following procedures to create tasks that release files from the file system by using the Amazon FSx console and CLI. Releasing a file retains the file listing and metadata, but removes the local copy of that file's contents.

#### To release files (console)

- Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/. 1.
- 2. In the left navigation pane, choose **File systems**, then choose your Lustre file system.
- Choose the **Data repository** tab. 3.
- In the **Data repository associations** pane, choose the data repository association that you want to create the release task for.
- For Actions, choose Create release task. This choice is available only if the file system is linked to a data repository on S3. The Create release data repository task dialog appears.
- In File system paths to release, specify up to 32 directories or files to release from your Amazon FSx file system by providing the paths to those directories or files. The paths that you provide must be relative to the mount point of the file system. For example, if the mount point is /mnt/fsx and /mnt/fsx/path1 is a file on the file system that you want to release, then the path to provide is path1. To release all files in the file system, specify a forward slash (/) as the path.



#### Note

If a path that you provide isn't valid, the task fails.

For Minimum duration since last access, specify the the duration, in days, such that any file 7. not accessed in that duration should be released. Last access time is calculated using the maximum value of atime, mtime, and ctime. Files with a last access duration period greater than minimum duration since last access (relative to the task create time) will be released. Files with a last access duration period less than this number of days won't be released, even if they are in the **File system paths to release** field. Provide a duration of 0 days to release files independent of duration since last access.

8. (Optional) Under **Completion report**, choose **Enable** to generate a task completion report that provides details about the files that meet the scope provided in **Report scope**. To specify a location for Amazon FSx to deliver the report, enter a relative path on the file system's linked S3 data repository for **Report path**.

9. Choose **Create data repository task**.

A notification at the top of the **File systems** page shows the task that you just created in progress.

To view the task status and details, in the **Data Repository** tab, scroll down to **Data Repository Tasks**. The default sort order shows the most recent task at the top of the list.

To view a task summary from this page, choose **Task ID** for the task you just created.

#### To release files (CLI)

 Use the <u>create-data-repository-task</u> CLI command to create a task that releases files on your FSx for Lustre file system. The corresponding API operation is CreateDataRepositoryTask.

Set the following parameters:

- Set --file-system-id to the ID of the file system that you are releasing files from.
- Set --paths to the paths on the file system from which the data will be released. If a directory is specified, files within the directory are released. If a file path is specified, only that file is released. To release all files in the file system that been exported to a linked S3 bucket, specify a forward slash (/) for the path.
- Set --type to RELEASE\_DATA\_FROM\_FILESYSTEM.
- Set the --release-configuration DurationSinceLastAccess options as follows:
  - Unit Set to DAYS.
  - Value Specify an integer that represents the the duration, in days, such that any file not accessed in that duration should be released. Files that were accessed during a period less than this number of days won't be released, even if they are in the --paths parameter. Provide a duration of 0 days to release files independent of duration since last access.

This sample command specifies that files that been exported to a linked S3 bucket and meet the --release-configuration criteria will be released from the directories in the specified paths.

```
$ aws fsx create-data-repository-task \
    --file-system-id fs-0123456789abcdef0 \
    --type RELEASE_DATA_FROM_FILESYSTEM \
    --paths path1,path2/file1 \
    --release-configuration '{"DurationSinceLastAccess":
    {"Unit":"DAYS","Value":10}}' \
    --report Enabled=false
```

After successfully creating the data repository task, Amazon FSx returns the task description as JSON.

After creating the task to release files, you can check the status of the task. For more information about viewing data repository tasks, see Accessing data repository tasks.

# Using Amazon FSx with your on-premises data

You can use FSx for Lustre to process your on-premises data with in-cloud compute instances. FSx for Lustre supports access over Direct Connect and VPN, enabling you to mount your file systems from on-premises clients.

#### To use FSx for Lustre with your on-premises data

- 1. Create a file system. For more information, see <a href="Step 1: Create your FSx for Lustre file system">Step 1: Create your FSx for Lustre file system</a> in the getting started exercise.
- 2. Mount the file system from on-premises clients. For more information, see <u>Mounting Amazon</u> FSx file systems from on-premises or a peered Amazon VPC.
- 3. Copy the data that you want to process into your FSx for Lustre file system.
- 4. Run your compute-intensive workload on in-cloud Amazon EC2 instances mounting your file system.
- 5. When you're finished, copy the final results from your file system back to your on-premises data location, and delete your FSx for Lustre file system.

# Data repository event logs

You can turn on logging to CloudWatch Logs to log information about any failures experienced while importing or exporting files using import, export, data repository tasks and restore events. For more information, see Logging with Amazon CloudWatch Logs.



### Note

When a data repository task fails, Amazon FSx also writes failure information to the task completion report. For more information about failure information in completion reports, see Troubleshooting data repository task failures.

### **Topics**

- Import events
- Export events
- HSM restore events

### Import events

| Error type           | Log<br>level | Log message   | Root cause  | Error code in completion report |
|----------------------|--------------|---|---|---------------------------------|
| List objects failure | ERROR        | Failed to list S3 objects in S3 bucket bucket_na me with prefix prefix. | Amazon FSx failed to list S3 objects in the S3 bucket. This can happen if the S3 bucket policy does not provide sufficient permissions to Amazon FSx. | N/A                             |

Data repository event logs 90

| Error type                        | Log<br>level | Log message   | Root cause  | Error code in completion report   |
|-----------------------------------|--------------|---|---|-----------------------------------|
| Unsupported S3 storage class      | WARN         | Failed to import S3 object with key key_value in S3 bucket bucket_na me due to an S3 object in an unsupport ed tier S3_tier_n ame .         | Amazon FSx was unable to import an S3 object because it's in an Amazon S3 storage class that is not supported , such as S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage class.     | S30bjectI<br>nUnsuppor<br>tedTier |
| Unsupported symlink storage class | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me due to an S3 symlink object in an unsupport ed tier S3_tier_n ame . | Amazon FSx was unable to import a symlink object because it's in an Amazon S3 storage class that is not supported , such as S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage class. | S3Symlink InUnsuppo rtedTier      |

| Error type       | Log<br>level | Log message   | Root cause   | Error code in completion report |
|------------------|--------------|---|--|---------------------------------|
| S3 access denied | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me because access to the S3 object was denied. | Access was denied to Amazon S3 for a data repository export import task.  For import tasks, the Amazon FSx file system must have permission to perform the s3: HeadOb ject and s3: GetObj ect operations to import from a linked data repository on S3.  For import tasks, if your S3 bucket uses server-side encryption with customer managed keys stored | S3AccessDenied                  |

| Error type           | Log<br>level | Log message  | Root cause  | Error code in completion report |
|----------------------|--------------|--|---|---------------------------------|
|                      |              |  | in AWS Key Management Service (SSE- KMS), you must follow the policy configurations in Working with server-si de encrypted Amazon S3 buckets. |                                 |
| Delete access denied | ERROR        | Failed to delete local file for S3 object with key key_value in S3 bucket bucket_na me because access to the S3 object was denied. | Automatic import was denied access to an S3 object.   | N/A                             |

| Error type                 | Log<br>level | Log message   | Root cause  | Error code in completion report           |
|----------------------------|--------------|---|---|---|
| Non-POSIX compliant object | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me because S3 object is not POSIX compliant.   | The Amazon S3 object exists but can't be imported because it isn't a POSIX-com pliant object. For informati on about supported POSIX metadata, see POSIX metadata support for data repositor ies. | S30bjectP<br>athNotPos<br>ixComplia<br>nt |
| Object type mismatch       | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me because an S3 object with the same name has already been imported into the file system. | The S3 object being imported is of a different type (file or directory) than an existing object with the same name in the file system.  | S30bjectT<br>ypeMismat<br>ch              |

| Error type                        | Log<br>level | Log message   | Root cause  | Error code in completion report |
|-----------------------------------|--------------|---|---|---------------------------------|
| Directory metadata update failure | ERROR        | Failed to update local directory metadata due to an internal error.                                   | Directory metadata could not be imported due to an internal error.  | N/A                             |
| S3 object not found               | ERROR        | Failed to import S3 object with key key_value because it was not found in S3 bucket bucket_na me .    | Amazon FSx was unable to import file metadata because the corresponding object doesn't exist in the data repositor y. | S3FileDel<br>eted               |
| S3 bucket not found               | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me due to bucket does not exist. | Amazon FSx cannot automatically import an S3 object to the file system because the S3 bucket no longer exists.        | N/A                             |

| Error type                 | Log<br>level | Log message  | Root cause  | Error code in completion report |
|----------------------------|--------------|--|---|---------------------------------|
| S3 bucket not found        | ERROR        | Failed to delete local file for S3 object with key key_value in S3 bucket bucket_na me due to bucket does not exist. | Amazon FSx cannot delete a file linked to an S3 object on the file system because the S3 bucket no longer exists. | N/A                             |
| Directory creation failure | ERROR        | Failed to create local directory due to an internal error.   | Amazon FSx failed to automatic ally import a directory creation on the file system due to an internal error.      | N/A                             |
| Disk space full            | ERROR        | Failed to import S3 object with key key_value in S3 bucket bucket_na me because the file system is full.             | File system ran out of disk space on the metadata server(s) while creating file or directory.                     | N/A                             |

FSx for Lustre User Guide

# **Export events**

| Error type    | Log level | Log message  | Root cause  | Error code in completion report |
|---------------|-----------|--|---|---------------------------------|
| Access denied | ERROR     | Failed to export file because access was denied to S3 object with key key_value in S3 bucket bucket_name . | Access was denied to Amazon S3 for a data repository export task.  For export tasks, the Amazon FSx file system must have permissio n to perform the s3: PutObj ect operation to export to a linked data repository on S3. This permissio n is granted in the AWSServic eRoleForF SxS3Acces s_ fs-012345 6789abcde f0 service-linked role. For more information, see Using service-linked roles for Amazon FSx. | S3AccessD enied                 |

Export events 97

| Error type | Log level | Log message | Root cause  | Error code in completion report |
|------------|-----------|-------------|---|---------------------------------|
|            |           |             | Because the export task requires data to flow outside a file system's VPC, this error can occur if the target repositor y has a bucket policy that contains one of the aws:Sourc eVpc or aws:Sourc eVpce IAM global condition keys.  If your S3 bucket contains objects uploaded from a different AWS account than your file system linked S3 bucket account, you can ensure that your data repository tasks can modify S3 metadata or overwrite S3 objects regardless of |                                 |

Export events 98

| which account uploaded them. We recommend that you enable the S3 Object Ownership feature for your S3 bucket. This feature enables you to take ownership of new objects that other AWS accounts upload to your bucket, by forcing uploads to provide the acl bucket- owner-full- control canned ACL. You enable S3 Object Ownership by choosing the Bucket owner preferred option in your S3 bucket. For more informati on, see Controlli | Error type | Log level | Log message | Root cause  | Error code in completion report |
|---|------------|-----------|-------------|---|---------------------------------|
| ng ownership  |            |           |             | uploaded them. We recommend that you enable the S3 Object Ownership feature for your S3 bucket. This feature enables you to take ownership of new objects that other AWS accounts upload to your bucket, by forcing uploads to provide the acl bucket- owner-full- control canned ACL. You enable S3 Object Ownership by choosing the Bucket owner preferred option in your S3 bucket. For more informati on, see Controlli |                                 |

| Error type           | Log level | Log message   | Root cause  | Error code in completion report |
|----------------------|-----------|---|---|---------------------------------|
|                      |           |   | of uploaded objects using S3 Object Ownership in the Amazon S3 User Guide.                      |                                 |
| Export path too long | ERROR     | Failed to export file because the local file path size exceeds the maximum object key length supported by S3. | The export path is too long. The maximum object key length supported by S3 is 1,024 characters. | PathSizeT<br>ooLong             |
| File too large       | ERROR     | Failed to export file because the file size exceeds the maximum supported S3 objects size.                    | The maximum object size supported by Amazon S3 is 5 TiB.  | FileSizeT<br>ooLarge            |

| Error type        | Log level | Log message   | Root cause  | Error code in completion report |
|-------------------|-----------|---|---|---------------------------------|
| KMS key not found | ERROR     | Failed to export file for S3 object with key key_value in S3 bucket bucket_name because the bucket's KMS key was not found. | Amazon FSx was unable to export the file because the AWS KMS key couldn't be found. Be sure to use a key that's in the same AWS Region as the S3 bucket. For more informati on on creating KMS keys, see Creating keys in the AWS Key Management Service Developer Guide. | N/A                             |

| Error type                    | Log level | Log message  | Root cause  | Error code in completion report |
|-------------------------------|-----------|--|---|---------------------------------|
| Resource busy                 | ERROR     | Failed to export file because it is being used by another process.   | Amazon FSx was unable to export the file because it was being modified by another client on the file system. You can retry the task after your workflow has finished writing to the file. | ResourceB                       |
| File released                 | WARN      | Export skipped: Local file is in released state and a linked S3 object with key key_value was not found in bucket bucket_name. | Amazon FSx was unable to export the file because it was in a released state on the file system.   | N/A                             |
| Data repository path mismatch | WARN      | Export skipped: local file does not belong to a data repositor y linked file system path.                                      | Amazon FSx was unable to export because the object doesn't belong to a file system path that is linked to a data repository.  | N/A                             |

| Error type                           | Log level | Log message  | Root cause  | Error code in completion report |
|--------------------------------------|-----------|--|---|---------------------------------|
| Internal failure                     | ERROR     | Automatic<br>export<br>encountered an<br>internal error<br>while exporting<br>a file system<br>object  | The export failed because of an internal (auto-export-or lustre-level) error.   | N/A                             |
| Completion report upload failure     | ERROR     | Failed to upload data repository task completion report into bucket_name   | Amazon FSx was unable to upload the completion report.  | N/A                             |
| Completion report validation failure | ERROR     | Failed to upload data repository task completion report into bucket bucket_name because the completion report path report_path does not belong to a data repository associated with this file system | Amazon FSx was unable to upload the completion report because the customer- provided S3 path does not belong to a linked data repository. | N/A                             |

# **HSM** restore events

| Error type                   | Log<br>level | Log message  | Root cause   |
|------------------------------|--------------|--|--|
| Access denied                | ERROR        | Failed to restore file because access was denied to S3 object object_na me in S3 bucket bucket_name .                        | Access was denied to Amazon S3 when attemptin g to restore a file using HSM commands. The file system must have permissio n to perform the s3:HeadObject and s3:GetObject operation s to restore from the linked data repository on S3.            |
| Unsupported S3 storage class | WARN         | Failed to restore file because S3 object object_na me in bucket bucket_na me was in an unsupported S3_storag e_class _name . | Amazon FSx was unable to restore the file because the corresponding S3 object is in an S3 unsupport ed storage class, such as S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive. You must first restore the object from the Glacier storage |

HSM restore events 104

| Error type          | Log<br>level | Log message  | Root cause  |
|---------------------|--------------|--|---|
|                     |              |  | <pre>class before using hsm_restore .</pre>   |
| S3 object not found | ERROR        | Failed to restore file because S3 object with key key_value was not found in S3 bucket bucket_name . | Amazon FSx was unable to restore the file because the corresponding S3 object doesn't exist in the data repository.   |
| S3 bucket not found | ERROR        | Failed to restore file because S3 bucket bucket_na me does not exist.                                | Amazon FSx cannot restore the file because the linked S3 bucket no longer exists.   |
| Disk space full     | ERROR        | Failed to restore file because there was no available storage space on the file system.              | The file system ran out of available storage space while attemptin g to restore the file data from S3. Consider increasin g the file system's storage capacity or releasing files to free up space. |

# Working with older deployment types

This section applies to file systems with Scratch 1 deployment type, and also to file systems with Scratch 2 or Persistent 1 deployment types that do not use data repository associations.

Note that automatic export and support for multiple data repositories aren't available on FSx for Lustre file systems that don't use data repository associations.

### **Topics**

- Link your file system to an Amazon S3 bucket
- Automatically import updates from your S3 bucket

## Link your file system to an Amazon S3 bucket

When you create an Amazon FSx for Lustre file system, you can link it to a durable data repository in Amazon S3. Before you create your file system, make sure that you have already created the Amazon S3 bucket to which you are linking. In the **Create file system** wizard, you set the following data repository configuration properties in the optional **Data Repository Import/Export** pane.

- Choose how Amazon FSx keeps your file and directory listing up to date as you add or modify objects in your S3 bucket after the file system is created. For more information, see Automatically import updates from your S3 bucket.
- Import bucket: Enter the name of the S3 bucket that you are using for the linked repository.
- Import prefix: Enter an optional import prefix if you want to import only some file and directory
  listings of data in your S3 bucket into your file system. The import prefix defines where in your
  S3 bucket to import data from.
- Export prefix: Defines where Amazon FSx exports the contents of your file system to your linked S3 bucket.

You can have a 1:1 mapping where Amazon FSx exports data from your FSx for Lustre file system back to the same directories on the S3 bucket that it was imported from. To have a 1:1 mapping, specify an export path to the S3 bucket without any prefixes when you create your file system.

- When you create file system using the console, choose Export prefix > A prefix you specify
  option, and keep the prefix field blank.
- When you create a file system using the AWS CLI or API, specify the export path as the name
  of the S3 bucket without any additional prefixes, for example, ExportPath=s3://amzn-s3demo-bucket/.

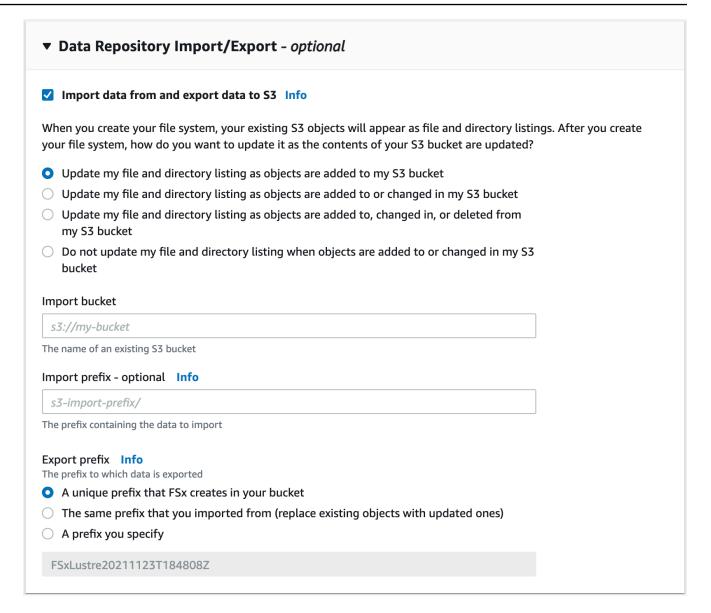
Using this method, you can include an import prefix when you specify the import path, and it doesn't impact a 1:1 mapping for exports.

### Creating file systems linked to an S3 bucket

The following procedures walk you through the process of creating an Amazon FSx file system linked to an S3 bucket using the AWS Management Console and AWS Command Line Interface (AWS CLI).

### Console

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **Create file system**.
- 3. For the file system type, choose **FSx for Lustre**, and then choose **Next**.
- 4. Provide the information required for the **File system details** and **Network and security** sections. For more information, see Step 1: Create your FSx for Lustre file system.
- You use the Data repository import/export panel to configure a linked data repository in Amazon S3. Select Import data from and export data to S3 to expand the Data Repository Import/Export section and configure the data repository settings.



- Choose how Amazon FSx keeps your file and directory listing up to date as you add or modify objects in your S3 bucket. When you create your file system, your existing S3 objects appear as file and directory listings.
  - Update my file and directory listing as objects are added to my S3 bucket: (Default)
    Amazon FSx automatically updates file and directory listings of any new objects added to
    the linked S3 bucket that do not currently exist in the FSx file system. Amazon FSx does
    not update listings for objects that have changed in the S3 bucket. Amazon FSx does not
    delete listings of objects that are deleted in the S3 bucket.



### Note

The default import preferences setting for importing data from a linked S3 bucket using the CLI and API is NONE. The default import preferences setting when using the console is to update Lustre as new objects are added to the S3 bucket.

- Update my file and directory listing as objects are added to or changed in my S3 **bucket**: Amazon FSx automatically updates file and directory listings of any new objects added to the S3 bucket and any existing objects that are changed in the S3 bucket after you choose this option. Amazon FSx does not delete listings of objects that are deleted in the S3 bucket.
- Update my file and directory listing as objects are added to, changed in, or deleted from my S3 bucket: Amazon FSx automatically updates file and directory listings of any new objects added to the S3 bucket, any existing objects that are changed in the S3 bucket, and any existing objects that are deleted in the S3 bucket after you choose this option.
- Do not update my file and directly listing when objects are added to, changed in, or deleted from my S3 bucket - Amazon FSx only updates file and directory listings from the linked S3 bucket when the file system is created. FSx does not update file and directory listings for any new, changed, or deleted objects after choosing this option.
- Enter an optional **Import prefix** if you want to import only some of the file and directory listings of data in your S3 bucket into your file system. The import prefix defines where in your S3 bucket to import data from. For more information, see Automatically import updates from your S3 bucket.
- 8. Choose one of the available **Export prefix** options:
  - A unique prefix that Amazon FSx creates in your bucket: Choose this option to export new and changed objects using a prefix generated by FSx for Lustre. The prefix looks like the following: /FSxLustrefile-system-creation- timestamp. The timestamp is in UTC format, for example FSxLustre20181105T222312Z.
  - The same prefix that you imported from (replace existing objects with updated ones): Choose this option to replace existing objects with updated ones.
  - A prefix you specify: Choose this option to preserve your imported data and to export new and changed objects using a prefix that you specify. To achieve a 1:1 mapping when

exporting data to your S3 bucket, choose this option and leave the prefix field blank. FSx will export data to same directories it was imported from.

- (Optional) Set Maintenance preferences, or use the system defaults.
- 10. Choose **Next**, and review the file system settings. Make any changes if needed.
- 11. Choose Create file system.

### **AWS CLI**

The following example creates an Amazon FSx file system linked to the amzn-s3-demobucket, with an import preference that imports any new, changed, and deleted files in the linked data repository after the file system is created.



### Note

The default import preferences setting for importing data from a linked S3 bucket using the CLI and API is NONE, which is different from the default behavior when using the console.

To create an FSx for Lustre file system, use the Amazon FSx CLI command create-filesystem, as shown below. The corresponding API operation is CreateFileSystem.

```
$ aws fsx create-file-system \
--client-request-token CRT1234 \
--file-system-type LUSTRE \
--file-system-type-version 2.10 \
--lustre-configuration
AutoImportPolicy=NEW_CHANGED_DELETED, DeploymentType=SCRATCH_1, ImportPath=s
3://amzn-s3-demo-bucket/,ExportPath=s3://amzn-s3-demo-bucket/export,
PerUnitStorageThroughput=50 \
--storage-capacity 2400 \
--subnet-ids subnet-123456 \
--tags Key=Name, Value=Lustre-TEST-1 \
--region us-east-2
```

After you successfully create the file system, Amazon FSx returns the file system description as JSON, as shown in the following example.

```
"FileSystems": [
        {
            "OwnerId": "owner-id-string",
            "CreationTime": 1549310341.483,
            "FileSystemId": "fs-0123456789abcdef0",
            "FileSystemType": "LUSTRE",
            "FileSystemTypeVersion": "2.10",
            "Lifecycle": "CREATING",
            "StorageCapacity": 2400,
            "VpcId": "vpc-123456",
            "SubnetIds": [
                "subnet-123456"
            ],
            "NetworkInterfaceIds": [
                "eni-039fcf55123456789"
            ],
            "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
            "ResourceARN": "arn:aws:fsx:us-east-2:123456:file-system/
fs-0123456789abcdef0",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Lustre-TEST-1"
                }
            ],
            "LustreConfiguration": {
                "DeploymentType": "PERSISTENT_1",
                "DataRepositoryConfiguration": {
                    "AutoImportPolicy": "NEW_CHANGED_DELETED",
                    "Lifecycle": "UPDATING",
                    "ImportPath": "s3://amzn-s3-demo-bucket/",
                    "ExportPath": "s3://amzn-s3-demo-bucket/export",
                    "ImportedFileChunkSize": 1024
                },
                "PerUnitStorageThroughput": 50
            }
        }
    ]
}
```

### Viewing a file system's export path

You can view a file system's export path using the FSx for Lustre console, the AWS CLI, and the API.

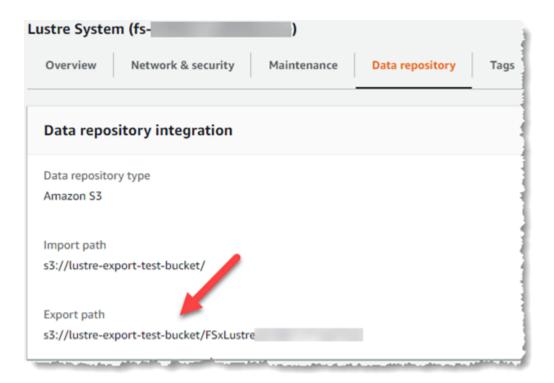
### Console

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/
- 2. Choose **File system name** or **File system ID** for the FSx for Lustre file system for which you want to view the export path.

The file system details page appears for that file system.

3. Choose the **Data repository** tab.

The **Data repository integration** panel appears, showing the import and export paths.



### CLI

To determine the export path for your file system, use the <u>describe-file-systems</u> AWS CLI command.

```
aws fsx describe-file-systems
```

Look for the ExportPath property under LustreConfiguration in the response.

```
{
    "OwnerId": "111122223333",
    "CreationTime": 1563382847.014,
    "FileSystemId": "",
    "FileSystemType": "LUSTRE",
    "Lifecycle": "AVAILABLE",
    "StorageCapacity": 2400,
    "VpcId": "vpc-6296a00a",
    "SubnetIds": [
        "subnet-1111111"
    ],
    "NetworkInterfaceIds": [
        "eni-0c288d5b8cc06c82d",
        "eni-0f38b702442c6918c"
    ],
    "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
    "ResourceARN": "arn:aws:fsx:us-east-2:267731178466:file-system/
fs-0123456789abcdef0",
    "Tags": [
        {
          "Key": "Name",
          "Value": "Lustre System"
    ],
 "LustreConfiguration": {
    "DeploymentType": "SCRATCH_1",
    "DataRepositoryConfiguration": {
    "AutoImportPolicy": " NEW_CHANGED_DELETED",
    "Lifecycle": "AVAILABLE",
    "ImportPath": "s3://amzn-s3-demo-bucket/",
    "ExportPath": "s3://amzn-s3-demo-bucket/FSxLustre20190717T164753Z",
    "ImportedFileChunkSize": 1024
    }
  },
  "PerUnitStorageThroughput": 50,
  "WeeklyMaintenanceStartTime": "6:09:30"
}
```

# Data repository lifecycle state

The data repository lifecycle state provides status information about the file system's linked data repository. A data repository can have the following Lifecycle states.

• Creating: Amazon FSx is creating the data repository configuration between the file system and the linked data repository. The data repository is unavailable.

- Available: The data repository is available for use.
- Updating: The data repository configuration is undergoing a customer-initiated update that might affect its availability.
- Misconfigured: Amazon FSx cannot automatically import updates from the S3 bucket until the data repository configuration is corrected. For more information, see Troubleshooting a misconfigured linked S3 bucket.

You can view a file system's linked data repository lifecycle state using the Amazon FSx console, the AWS Command Line Interface, and the Amazon FSx API. In the Amazon FSx console, you can access the data repository Lifecycle state in the Data Repository Integration pane of the Data Repository tab for the file system. The Lifecycle property is located in the DataRepositoryConfiguration object in the response of a describe-file-systems CLI command (the equivalent API action is DescribeFileSystems).

# Automatically import updates from your S3 bucket

By default, when you create a new file system, Amazon FSx imports the file metadata (the name, ownership, timestamp, and permissions) of objects in the linked S3 bucket at file system creation. You can configure your FSx for Lustre file system to automatically import metadata of objects that are added to, changed in, or deleted from your S3 bucket after file system creation. FSx for Lustre updates the file and directory listing of a changed object after creation in the same manner as it imports file metadata at file system creation. When Amazon FSx updates the file and directory listing of a changed object, if the changed object in the S3 bucket no longer contains its metadata, Amazon FSx maintains the current metadata values of the file, rather than using default permissions.



### Note

Import settings are available on FSx for Lustre file systems created after 3:00 pm EDT, July 23, 2020.

You can set import preferences when you create a new file system, and you can update the setting on existing file systems using the FSx management console, the AWS CLI, and the AWS API. When

you create your file system, your existing S3 objects appear as file and directory listings. After you create your file system, how do you want to update it as the contents of your S3 bucket are updated? A file system can have one of the following Import preferences:



### Note

The FSx for Lustre file system and its linked S3 bucket must be located in the same AWS Region to automatically import updates.

• Update my file and directory listing as objects are added to my S3 bucket: (Default) Amazon FSx automatically updates file and directory listings of any new objects added to the linked S3 bucket that do not currently exist in the FSx file system. Amazon FSx does not update listings for objects that have changed in the S3 bucket. Amazon FSx does not delete listings of objects that are deleted in the S3 bucket.

### Note

The default import preferences setting for importing data from a linked S3 bucket using the CLI and API is NONE. The default import preferences setting when using the console is to update Lustre as new objects are added to the S3 bucket.

- Update my file and directory listing as objects are added to or changed in my S3 bucket: Amazon FSx automatically updates file and directory listings of any new objects added to the S3 bucket and any existing objects that are changed in the S3 bucket after you choose this option. Amazon FSx does not delete listings of objects that are deleted in the S3 bucket.
- Update my file and directory listing as objects are added to, changed in, or deleted from my S3 bucket: Amazon FSx automatically updates file and directory listings of any new objects added to the S3 bucket, any existing objects that are changed in the S3 bucket, and any existing objects that are deleted in the S3 bucket after you choose this option.
- Do not update my file and directly listing when objects are added to, changed in, or deleted from my S3 bucket - Amazon FSx only updates file and directory listings from the linked S3 bucket when the file system is created. FSx does not update file and directory listings for any new, changed, or deleted objects after choosing this option.

When you set the import preferences to update your file system file and directory listings based on changes in the linked S3 bucket, Amazon FSx creates an event notification configuration on the

linked S3 bucket named FSx. Do not modify or delete the FSx event notification configuration on the S3 bucket—doing so prevents the automatic import of new or changed file and directory listings to your file system.

When Amazon FSx updates a file listing that has changed on the linked S3 bucket, it overwrites the local file with the updated version, even if the file is write-locked. Similarly, when Amazon FSx updates a file listing when the corresponding object has been deleted on the linked S3 bucket, it deletes the local file, even if the file is write-locked.

Amazon FSx makes a best effort to update your file system. Amazon FSx cannot update the file system with changes in the following situations:

- When Amazon FSx does not have permission to open the changed or new S3 object.
- When the FSx event notification configuration on the linked S3 bucket is deleted or changed.

Either of these conditions cause the data repository lifecycle state to become **Misconfigured**. For more information, see Data repository lifecycle state.

### **Prerequisites**

The following conditions are required for Amazon FSx to automatically import new, changed, or deleted files from the linked S3 bucket:

- The file system and its linked S3 bucket must be located in the same AWS Region.
- The S3 bucket does not have a misconfigured Lifecycle state. For more information, see <a href="Data">Data</a> repository lifecycle state.
- Your account must have the permissions required to configure and receive event notifications on the linked S3 bucket.

## Types of file changes supported

Amazon FSx supports importing the following changes to files and folders that occur in the linked S3 bucket:

- · Changes to file contents
- Changes to file or folder metadata
- Changes to symlink target or metadata

### **Updating import preferences**

You can set a file system's import preferences when you create a new file system. For more information, see Linking your file system to an Amazon S3 bucket.

You can also update a file system's import preferences after it is created using the AWS Management Console, the AWS CLI, and the Amazon FSx API, as shown in the following procedure.

### Console

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the dashboard, choose **File systems**.
- 3. Select the file system that you want to manage to display the file system details.
- 4. Choose **Data repository** to view the data repository settings. You can modify the import preferences if the lifecycle state is **AVAILABLE** or **MISCONFIGURED**. For more information, see Data repository lifecycle state.
- 5. Choose **Actions**, and then choose **Update import preferences** to display the **Update import preferences** dialog box.
- 6. Select the new setting, and then choose **Update** to make the change.

CLI

To update import preferences, use the <u>update-file-system</u> CLI command. The corresponding API operation is <u>UpdateFileSystem</u>.

After you successfully update the file system's AutoImportPolicy, Amazon FSx returns the description of the updated file system as JSON, as shown here:

```
"subnet-123456"
            ],
            "NetworkInterfaceIds": [
                "eni-039fcf55123456789"
            ],
            "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
            "ResourceARN": "arn:aws:fsx:us-east-2:123456:file-system/
fs-0123456789abcdef0",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Lustre-TEST-1"
                }
            ],
            "LustreConfiguration": {
                "DeploymentType": "SCRATCH_1",
                "DataRepositoryConfiguration": {
                    "AutoImportPolicy": "NEW_CHANGED_DELETED",
                    "Lifecycle": "UPDATING",
                    "ImportPath": "s3://amzn-s3-demo-bucket/",
                    "ExportPath": "s3://amzn-s3-demo-bucket/export",
                    "ImportedFileChunkSize": 1024
                }
                "PerUnitStorageThroughput": 50,
                "WeeklyMaintenanceStartTime": "2:04:30"
            }
        }
    ]
}
```

# **Amazon FSx for Lustre performance**

This chapter provides Amazon FSx for Lustre performance topics, including some important tips and recommendations for maximizing the performance of your file system.

### **Topics**

- Overview
- How FSx for Lustre file systems work
- · File system metadata performance
- Throughput to individual client instances
- File system storage layout
- Striping data in your file system
- Monitoring performance and usage
- Performance characteristics of SSD and HDD storage classes
- Performance characteristics of Intelligent-Tiering storage class
- Performance tips

### **Overview**

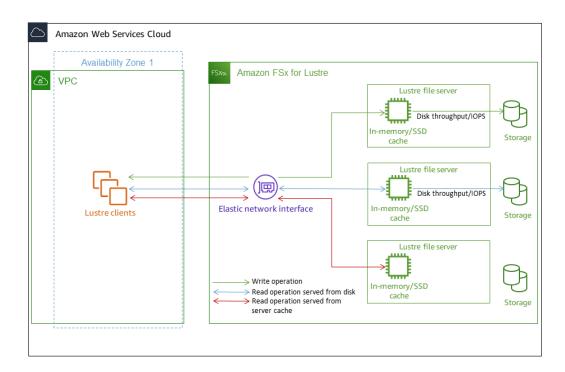
Amazon FSx for Lustre, built on Lustre, the popular high-performance file system, provides scale-out performance that increases linearly with a file system's size. Lustre file systems scale horizontally across multiple file servers and disks. This scaling gives each client direct access to the data stored on each disk to remove many of the bottlenecks present in traditional file systems. Amazon FSx for Lustre builds on the Lustre scalable architecture to support high levels of performance across large numbers of clients.

# How FSx for Lustre file systems work

Each FSx for Lustre file system consists of the file servers that the clients communicate with, and a set of disks attached to each file server that store your data. Each file server employs a fast, inmemory cache to enhance performance for the most frequently accessed data. Depending on the storage class, your file server can be provisioned with an optional SSD read cache. When a client accesses data that's stored in the in-memory or SSD cache, the file server doesn't need to read it

Overview 119

from disk, which reduces latency and increases the total amount of throughput you can drive. The following diagram illustrates the paths of a write operation, a read operation served from disk, and a read operation served from in-memory or SSD cache.



When you read data that is stored on the file server's in-memory or SSD cache, file system performance is determined by the network throughput. When you write data to your file system, or when you read data that isn't stored on the in-memory cache, file system performance is determined by the lower of the network throughput and disk throughput.

To learn more about network throughput, disk throughput, and IOPS characteristics of SSD and HDD storage classes, see <a href="Performance characteristics of Intelligent-Tiering storage class">Performance characteristics of Intelligent-Tiering storage class</a>.

# File system metadata performance

File system metadata IO operations per second (IOPS) determine the number of files and directories that you can create, list, read, and delete per second.

Persistent 2 file systems allow you to provision Metadata IOPS independent from storage capacity and provide increased visibility into the number and type of metadata IOPS client instances are

driving on your file system. With SSD file systems, Metadata IOPS are automatically provisioned based on the storage capacity that you provision. Automatic mode is not supported on Intelligent-Tiering file systems.

With FSx for Lustre Persistent 2 file systems, the number of Metadata IOPS you provision and the type of metadata operation determine the rate of metadata operations that your file system can support. The level of metadata IOPS you provision determines the number of IOPS provisioned for your file system's metadata disks.

| Type of operation           | Operations you can drive per second for each provisioned metadata IOPS |
|-----------------------------|--|
| File Create, Open and Close | 2  |
| File Delete                 | 1  |
| Directory Create, Rename    | 0.1  |
| Directory Delete            | 0.2  |

For SSD file systems, you can choose to provision metadata IOPS using Automatic mode. In Automatic mode, Amazon FSx automatically provisions metadata IOPS based on the storage capacity of your file system according to the table below:

| File system storage capacity | Included metadata IOPS in Automatic mode |
|------------------------------|--|
| 1200 GiB                     | 1500                                     |
| 2400 GiB                     | 3000                                     |
| 4800–9600 GiB                | 6000                                     |
| 12000–45600 GiB              | 12000                                    |
| ≥48000 GiB                   | 12000 IOPS per 24000 GiB                 |

In User-provisioned mode, you can optionally choose to specify the number of metadata IOPS to provision. Valid values are as follows:

 For SSD file systems, valid values are 1500, 3000, 6000, 12000, and multiples of 12000 up to a maximum of 192000.

• For Intelligent-Tiering file systems, valid values are 6000 and 12000.

For information about how to configure Metadata IOPS, see Managing metadata performance. Note that you pay for Metadata IOPS provisioned above the default number of Metadata IOPS for your file system.

# Throughput to individual client instances

If you are creating a file system with over 10 GBps of throughput capacity, we recommend enabling Elastic Fabric Adapter (EFA) to optimize throughput per client instance. To further optimize throughput per client instance, EFA-enabled file systems also support GPUDirect Storage for EFA-enabled NVIDIA GPU-based client instances and ENA Express for ENA Express-enabled client instances.

The throughput that you can drive to a single client instance depends on your choice of file system type and the network interface on your client instance.

| File system type | Client instance network interface | Maximum throughput per client, Gbps |
|------------------|-----------------------------------|-------------------------------------|
| Non EFA-enabled  | Any                               | 100 Gbps*                           |
| EFA-enabled      | ENA                               | 100 Gbps*                           |
| EFA-enabled      | ENA Express                       | 100 Gbps                            |
| EFA-enabled      | EFA                               | 700 Gbps                            |
| EFA-enabled      | EFA with GDS                      | 1200 Gbps                           |



### Note

<sup>\*</sup> The traffic between an individual client instance and an individual FSx for Lustre object storage server is limited to 5 Gbps. Refer to the IP addresses for file systems for the number of object storage servers underpinning your FSx for Lustre file system.

# File system storage layout

All file data in Lustre is stored on storage volumes called *object storage targets* (OSTs). All file metadata (including file names, timestamps, permissions, and more) is stored on storage volumes called *metadata targets* (MDTs). Amazon FSx for Lustre file systems are composed of one or more MDTs and multiple OSTs. Amazon FSx for Lustre spreads your file data across the OSTs that make up your file system to balance storage capacity with throughput and IOPS load.

To view the storage usage of the MDT and OSTs that make up your file system, run the following command from a client that has the file system mounted.

```
lfs df -h mount/path
```

The output of this command looks like the following.

### Example

| UUID  mountname-MDT0000_UUID  mountname-OST0000_UUID  mountname-OST0001_UUID | bytes<br>68.7G<br>1.1T<br>1.1T | Used<br>5.4M<br>4.5M<br>4.5M | Available Use% Mounted on 68.7G 0% /fsx[MDT:0] 1.1T 0% /fsx[OST:0] 1.1T 0% /fsx[OST:1] |
|--|--------------------------------|------------------------------|--|
| filesystem_summary:  | 2.2T                           | 9.0M                         | 2.2T 0% /fsx   |

# Striping data in your file system

You can optimize your file system's throughput performance with file striping. Amazon FSx for Lustre automatically spreads out files across OSTs in order to ensure that data is served from all storage servers. You can apply the same concept at the file level by configuring how files are striped across multiple OSTs.

Striping means that files can be divided into multiple chunks that are then stored across different OSTs. When a file is striped across multiple OSTs, read or write requests to the file are spread across those OSTs, increasing the aggregate throughput or IOPS your applications can drive through it.

Following are the default layouts for Amazon FSx for Lustre file systems.

File system storage layout 123

• For file systems created before December 18, 2020, the default layout specifies a stripe count of 1. This means that unless a different layout is specified, each file created in Amazon FSx for Lustre using standard Linux tools is stored on a single disk.

- For file systems created after December 18, 2020, the default layout is a progressive file layout in which files under 1GiB in size are stored in one stripe, and larger files are assigned a stripe count of 5.
- For file systems created after August 25, 2023, the default layout is a 4-component progressive file layout which is explained in Progressive file layouts.
- For all file systems regardless of their creation date, files imported from Amazon S3 don't use the default layout, but instead use the layout in the file system's ImportedFileChunkSize parameter. S3-imported files larger than the ImportedFileChunkSize will be stored on multiple OSTs with a stripe count of (FileSize / ImportedFileChunksize) + 1. The default value of ImportedFileChunkSize is 1GiB.

You can view the layout configuration of a file or directory using the lfs getstripe command.

```
lfs getstripe path/to/filename
```

This command reports a file's stripe count, stripe size, and stripe offset. The *stripe count* is how many OSTs the file is striped across. The *stripe size* is how much continuous data is stored on an OST. The *stripe offset* is the index of the first OST that the file is striped across.

# Modifying your striping configuration

A file's layout parameters are set when the file is first created. Use the lfs setstripe command to create a new, empty file with a specified layout.

```
lfs setstripe filename --stripe-count number_of_OSTs
```

The lfs setstripe command affects only the layout of a new file. Use it to specify the layout of a file before you create it. You can also define a layout for a directory. Once set on a directory, that layout is applied to every new file added to that directory, but not to existing files. Any new subdirectory you create also inherits the new layout, which is then applied to any new file or directory you create within that subdirectory.

To modify the layout of an existing file, use the lfs migrate command. This command copies the file as needed to distribute its content according to the layout you specify in the command. For

example, files that are appended to or are increased in size don't change the stripe count, so you have to migrate them to change the file layout. Alternatively, you can create a new file using the lfs setstripe command to specify its layout, copy the original content to the new file, and then rename the new file to replace the original file.

There may be cases where the default layout configuration is not optimal for your workload. For example, a file system with tens of OSTs and a large number of multi-gigabyte files may see higher performance by striping the files across more than the default stripe count value of five OSTs. Creating large files with low stripe counts can cause I/O performance bottlenecks and can also cause OSTs to fill up. In this case, you can create a directory with a larger stripe count for these files.

Setting up a striped layout for large files (especially files larger than a gigabyte in size) is important for the following reasons:

- Improves throughput by allowing multiple OSTs and their associated servers to contribute IOPS, network bandwidth, and CPU resources when reading and writing large files.
- Reduces the likelihood that a small subset of OSTs become hot spots that limit overall workload performance.
- Prevents a single large file from filling an OST, possibly causing disk full errors.

There is no single optimal layout configuration for all use cases. For detailed guidance on file layouts, see Managing File Layout (Striping) and Free Space in the Lustre.org documentation. The following are general guidelines:

- Striped layout matters most for large files, especially for use cases where files are routinely
  hundreds of megabytes or more in size. For this reason, the default layout for a new file system
  assigns a striped count of five for files over 1GiB in size.
- Stripe count is the layout parameter that you should adjust for systems supporting large files.
  The stripe count specifies the number of OST volumes that will hold chunks of a striped file. For example, with a stripe count of 2 and a stripe size of 1MiB, Lustre writes alternate 1MiB chunks of a file to each of two OSTs.
- The effective stripe count is the lesser of the actual number of OST volumes and the stripe count value you specify. You can use the special stripe count value of -1 to indicate that stripes should be placed on all OST volumes.

• Setting a large stripe count for small files is sub-optimal because for certain operations Lustre requires a network round trip to every OST in the layout, even if the file is too small to consume space on all the OST volumes.

- You can set up a progressive file layout (PFL) that allows the layout of a file to change with size.
   A PFL configuration can simplify managing a file system that has a combination of large and small files without you having to explicitly set a configuration for each file. For more information, see Progressive file layouts.
- Stripe size by default is 1MiB. Setting a stripe offset may be userful in special circumstances, but in general it is best to leave it unspecified and use the default.

# **Progressive file layouts**

You can specify a progressive file layout (PFL) configuration for a directory to specify different stripe configurations for small and large files before populating it. For example, you can set a PFL on the top-level directory before any data is written to a new file system.

To specify a PFL configuration, use the lfs setstripe command with -E options to specify layout components for different sized files, such as the following command:

```
lfs setstripe -E 100M -c 1 -E 10G -c 8 -E 100G -c 16 -E -1 -c 32 /mountname/directory
```

This command sets four layout components:

- The first component (-E 100M -c 1) indicates a stripe count value of 1 for files up to 100MiB in size.
- The second component (-E 10G -c 8) indicates a stripe count of 8 for files up to 10GiB in size.
- The third component (-E 100G -c 16) indicates a stripe count of 16 for files up to 100GiB in size.
- The fourth component (-E -1 -c 32) indicates a stripe count of 32 for files larger than 100GiB.

### 

Appending data to a file created with a PFL layout will populate all of its layout components. For example, with the 4-component command shown above, if you create a 1MiB file and then add data to the end of it, the layout of the file will expand to have

Progressive file layouts 126

a stripe count of -1, meaning all the OSTs in the system. This does not mean data will be written to every OST, but an operation such as reading the file length will send a request in parallel to every OST, adding significant network load to the file system.

Therefore, be careful to limit the stripe count for any small or medium length file that can subsequently have data appended to it. Because log files usually grow by having new records appended, Amazon FSx for Lustre assigns a default stripe count of 1 to any file created in append mode, regardless of the default stripe configuration specified by its parent directory.

The default PFL configuration on Amazon FSx for Lustre file systems created after August 25, 2023 is set with this command:

```
lfs setstripe -E 100M -c 1 -E 10G -c 8 -E 100G -c 16 -E -1 -c 32 /mountname
```

Customers with workloads that have highly concurrent access on medium and large files are likely to benefit from a layout with more stripes at smaller sizes and striping across all OSTs for the largest files, as shown in the four-component example layout.

# Monitoring performance and usage

Every minute, Amazon FSx for Lustre emits usage metrics for each disk (MDT and OST) to Amazon CloudWatch.

To view aggregate file system usage details, you can look at the Sum statistic of each metric. For example, the Sum of the DataReadBytes statistic reports the total read throughput seen by all the OSTs in a file system. Similarly, the Sum of the FreeDataStorageCapacity statistic reports the total available storage capacity for file data in the file system.

For more information on monitoring your file system's performance, see <u>Monitoring Amazon FSx</u> <u>for Lustre file systems</u>.

# Performance characteristics of SSD and HDD storage classes

The throughput that an FSx for Lustre file system provisioned with SSD or HDD storage class supports is proportional to its storage capacity. Amazon FSx for Lustre file systems scale to

multiple TBps of throughput and millions of IOPS. Amazon FSx for Lustre also supports concurrent access to the same file or directory from thousands of compute instances. This access enables rapid data checkpointing from application memory to storage, which is a common technique in high performance computing (HPC). You can increase the amount of storage and throughput capacity as needed at any time after you create the file system. For more information, see <a href="Managing storage">Managing storage</a> capacity.

FSx for Lustre file systems provide burst read throughput using a network I/O credit mechanism to allocate network bandwidth based on average bandwidth utilization. The file systems accrue credits when their network bandwidth usage is below their baseline limits, and can use these credits when they perform network data transfers.

The following tables show performance that the FSx for Lustre deployment options using SSD and HDD storage classes are designed for.

# File system performance for SSD storage options

| Disk throughput (MBps/latencies TiB of storage or SSD per file cache provisioned) operation (millisec onds, P50) | Baseline Burst | Metadata: 200 (read) -<br>sub-ms 100 (write) | Data: sub-<br>ms 125 500 | 250 500            | - 200              | 1000                |
|--|----------------|--|--------------------------|--------------------|--------------------|---------------------|
| a ab   |                | Me<br>sul                                    | Dat                      |                    |                    |                     |
| storage storage (GiB of ge RAM/TiB n of storage provision ed)  |                | 6.7<br>ds                                    | s 3.4                    | ds 6.8             | 13.7               | 27.3                |
| Network<br>IOPS<br>(IOPS/TiB<br>of storage<br>provision<br>ed)   |                | Tens of<br>thousands<br>baseline             | Hundreds                 | thousands<br>burst |                    |                     |
| of storage<br>d)   | Burst          | 1300   | 1300                     | 1300               | ı                  | ı                   |
| Network throughp<br>(MBps/TiB of stora<br>provisioned)   | Baseline       | 200  | 320                      | 640                | 1300               | 2600                |
| Deploymen<br>t Type  |                | SCRATCH_2                                    | PERSISTEN<br>T-125       | PERSISTEN<br>T-250 | PERSISTEN<br>T-500 | PERSISTEN<br>T 1000 |

# File system performance for HDD storage options

| Deploymen<br>t Type | Network throughput<br>(MBps/TiB of storage or<br>SSD cache provisioned) | oughput<br>storage or<br>ovisioned) | Network<br>IOPS<br>(IOPS/TiB<br>of storage<br>provision<br>ed) | Cache storage (GiB of RAM/TiB of storage provision ed) | Disk<br>latencies<br>per file<br>operation<br>(millisec<br>onds, P50) | Disk throughput (MBps/<br>TiB of storage or SSD<br>cache provisioned) | put (MBps/<br>e or SSD<br>oned) |
|---------------------|---|-------------------------------------|--|--|---|---|---------------------------------|
|                     | Baseline  | Burst                               |  |  |   | Baseline  | Burst                           |
| PERSISTENT-12       | 12  |                                     |  |  |   |   |                                 |
| HDD<br>storage      | 40  | 375*                                | Tens of<br>thousands   | 0.4 memory   | Metadata:<br>sub-ms   | 12  | 80 (read)                       |
|                     |   |                                     | baseline   |  | Data:   |   | (2) (4)                         |
|                     |   |                                     | Hundreds   |  | single-digit  |   |                                 |
|                     |   |                                     | o <del>t</del>   |  | ms  |   |                                 |
| SSD read<br>cache   | 200   | 1,900                               | thousands<br>burst   | 200 SSD<br>cache                                       | Data: sub-<br>ms  | 200   | 1                               |
| PERSISTENT-40       | 40  |                                     |  |  |   |   |                                 |
| HDD                 | 150   | 1,300*                              | Tens of  | 1.5  | Metadata:   | 40  | 250 (read)                      |
| storage             |   |                                     | thousands<br>baseline  |  | sub-ms<br>Data:   |   | 150 (write)                     |
|                     |   |                                     | Hundreds<br>of   |  | single-digit<br>ms  |   |                                 |
| SSD read<br>cache   | 750   | 6500                                | thousands<br>burst   | 200 SSD<br>cache                                       | Data: sub-<br>ms  | 200   |                                 |

# File system performance for previous generation SSD storage options

| Disk throughput (MBps<br>per TiB of storage or SSD<br>cache provisioned) | ne Burst | 240                  | 240        |          | 240                |
|--|----------|----------------------|------------|----------|--------------------|
| Disk tl<br>per Til<br>cache  | Baseline | 20                   | 100        |          | 200                |
| Disk<br>latencies<br>per file<br>operation<br>(millisec<br>onds, P50)    |          | Metadata:<br>sub-ms  | Data: sub- | ms       |                    |
| Cache storage (GiB per TiB of storage provision ed)                      |          | 2.2 RAM              | 4.4 RAM    |          | 8.8 RAM            |
| Network IOPS (IOPS per TiB of storage provision ed)                      |          | Tens of<br>thousands | baseline   | Hundreds | thousands<br>burst |
| Network throughput<br>(MBps per TiB of storage<br>provisioned)           | Burst    | 1,300*               | 1,300*     |          | 1,300*             |
| Network throughpu<br>(MBps per TiB of sto<br>provisioned)                | Baseline | 250                  | 200        |          | 750                |
| Deploymen<br>t Type  |          | PERSISTEN<br>T-50    | PERSISTEN  | T-100    | PERSISTEN<br>T-200 |



### Note

\* Persistent file systems in the following AWS Regions provide network burst up to 530 MBps per TiB of storage: Africa (Cape Town), Asia Pacific (Hong Kong), Asia Pacific (Osaka), Asia Pacific (Singapore), Canada (Central), Europe (Frankfurt), Europe (London), Europe (Milan), Europe (Stockholm), Middle East (Bahrain), South America (São Paulo), China, and US West (Los Angeles).

# Example: Aggregate baseline and burst throughput

The following example illustrates how storage capacity and disk throughput impact file system performance.

A persistent file system with a storage capacity of 4.8 TiB and 50 MBps per TiB of throughput per unit of storage provides an aggregate baseline disk throughput of 240 MBps and a burst disk throughput of 1.152 GBps.

Regardless of file system size, Amazon FSx for Lustre provides consistent, sub-millisecond latencies for file operations.

# Performance characteristics of Intelligent-Tiering storage class

The FSx for Lustre Intelligent-Tiering storage class offers elastic, cost-optimized storage for workloads that traditionally run on HDD-based or mixed HDD-/SDD-based high-performance file storage file systems. File systems using the Intelligent-Tiering storage class utilize fully elastic, intelligently tiered, regional storage which automatically grows and shrinks to fit your workload as it changes. For information on how it tiers data, see How the Intelligent-Tiering storage class tiers data.

The throughput that an FSx for Lustre file system with Intelligent-Tiering storage class supports is independent to its storage. Intelligent-Tiering file systems scale to multiple TBps of throughput and millions of IOPS. File systems using the Intelligent-Tiering storage class also provide an optional provisioned SSD read cache for low-latency access to frequently accessed data. By default, Amazon FSx for Lustre provisions an SSD read cache for frequently accessed metadata. 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 the SSD read caches allows

file systems using the Intelligent-Tiering storage class to provide storage that performs at levels comparable to SSD file systems for most workloads, while delivering storage cost savings relative to the SSD and HDD storage classes.

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.

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. You can switch between SSD read cache sizing modes after your file system has been created and scale the cache up or down. For more information on how to modify your SSD read cache, see Managing provisioned SSD read cache.

A write request occurs when FSx for Lustre writes a block of data to Intelligent-Tiering storage. When you write data to the file system, write requests are aggregated and written to Intelligent-Tiering storage, 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 Lustre 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.

# File system performance for Intelligent-Tiering

The following table shows the performance that FSx for Lustre Intelligent-Tiering file systems are designed for.

FSx for Lustre User Guide

| Provisioned<br>throughpu<br>t capacity<br>(MBps) | Network throughput<br>(MBps) | roughput | Network<br>IOPS | In-memory<br>cache<br>storage<br>(GB) | Maximum<br>SSD cache<br>disk<br>throughput<br>(MBps) | Maximum SSD cache disk<br>IOPS | is X |
|--|------------------------------|----------|-----------------|---------------------------------------|--|--------------------------------|------|
|  | Baseline                     | Burst    |                 |                                       |  | Baseline Burst                 |      |
| Every 4000                                       | 12500                        | ı        | Hundreds        | 76.8                                  | 4000   | 160000                         |      |
|  |                              |          | of              |                                       |  |                                |      |
|  |                              |          | thousands       |                                       |  |                                |      |

# **Performance tips**

When using Amazon FSx for Lustre, keep the following performance tips in mind. For service limits, see Service quotas for Amazon FSx for Lustre.

- Average I/O size Because Amazon FSx for Lustre is a network file system, each file operation goes through a round trip between the client and Amazon FSx for Lustre, incurring a small latency overhead. Due to this per-operation latency, overall throughput generally increases as the average I/O size increases, because the overhead is amortized over a larger amount of data.
- Request model By enabling asynchronous writes to your file system, pending write operations are buffered on the Amazon EC2 instance before they are written to Amazon FSx for Lustre asynchronously. Asynchronous writes typically have lower latencies. When performing asynchronous writes, the kernel uses additional memory for caching. A file system that has enabled synchronous writes issues synchronous requests to Amazon FSx for Lustre. Every operation goes through a round trip between the client and Amazon FSx for Lustre.



#### Note

Your chosen request model has tradeoffs in consistency (if you're using multiple Amazon EC2 instances) and speed.

- Limit directory size To achieve optimal metadata performance on Persistent 2 FSx for Lustre file systems, limit each directory to less than 100K files. Limiting the number of files in a directory reduces the time required for the file system to acquire a lock on the parent directory.
- Amazon EC2 instances Applications that perform a large number of read and write operations likely need more memory or computing capacity than applications that don't. When launching your Amazon EC2 instances for your compute-intensive workload, choose instance types that have the amount of these resources that your application needs. The performance characteristics of Amazon FSx for Lustre file systems don't depend on the use of Amazon EBS-optimized instances.
- Recommended client instance tuning for optimal performance
  - 1. For client instance types with memory of more than 64 GiB, we recommend applying the following tuning:

```
sudo lctl set_param ldlm.namespaces.*.lru_max_age=600000
sudo lctl set_param ldlm.namespaces.*.lru_size=<100 * number_of_CPUs>
```

Performance tips 135

2. For client instance types with more than 64 vCPU cores, we recommend applying the following tuning:

```
echo "options ptlrpc ptlrpcd_per_cpt_max=32" >> /etc/modprobe.d/modprobe.conf
echo "options ksocklnd credits=2560" >> /etc/modprobe.d/modprobe.conf

# reload all kernel modules to apply the above two settings
sudo reboot
```

After the client is mounted, the following tuning needs to be applied:

```
sudo lctl set_param osc.*0ST*.max_rpcs_in_flight=32
sudo lctl set_param mdc.*.max_rpcs_in_flight=64
sudo lctl set_param mdc.*.max_mod_rpcs_in_flight=50
```

3. To optimize performance for directory listing (ls), the following tuning needs to be applied:

```
sudo lctl set_param llite.*.statahead_max=512
sudo lctl set_param llite.*.statahead_agl=1
if sudo lctl get_param llite.*.statahead_xattr > /dev/null 2>&1; then
    sudo lctl set_param llite.*.statahead_xattr=1
else
    echo "Warning: Xattr statahead is not supported on this Lustre client. Please
upgrade to the latest Lustre 2.15 client to apply this tuning"
fi
```

Note that lctl set\_param is known to not persist over reboot. Since these parameters cannot be set permanently from the client side, it is recommended to implement a boot cron job to set the configuration with the recommended tunings.

• Workload balance across OSTs – In some cases, your workload isn't driving the aggregate throughput that your file system can provide (200 MBps per TiB of storage). If so, you can use CloudWatch metrics to troubleshoot if performance is affected by an imbalance in your workload's I/O patterns. To identify if this is the cause, look at the Maximum CloudWatch metric for Amazon FSx for Lustre.

In some cases, this statistic shows a load at or above 240 MBps of throughput (the throughput capacity of a single 1.2-TiB Amazon FSx for Lustre disk). In such cases, your workload is not evenly spread out across your disks. If this is the case, you can use the lfs setstripe command to modify the striping of files your workload is most frequently accessing. For optimal

Performance tips 136

performance, stripe files with high throughput requirements across all the OSTs comprising your file system.

If your files are imported from a data repository, you can take another approach to stripe your high-throughput files evenly across your OSTs. To do this, you can modify the ImportedFileChunkSize parameter when creating your next Amazon FSx for Lustre file system.

For example, suppose that your workload uses a 7.0-TiB file system (which is made up of 6x 1.17-TiB OSTs) and needs to drive high throughput across 2.4-GiB files. In this case, you can set the ImportedFileChunkSize value to (2.4 GiB / 6 OSTs) = 400 MiB so that your files are spread evenly across your file system's OSTs.

• Lustre client for Metadata IOPS – If your file system has a metadata configuration specified, we recommend you install a Lustre 2.15 client or a Lustre 2.12 client with one of these OS versions: Amazon Linux 2023; Amazon Linux 2; Red Hat/Rocky Linux 8.9, 8.10, or 9.x; CentOS 8.9 or 8.10; Ubuntu 22+ with 6.2, 6.5, or 6.8 kernel; or Ubuntu 20.

## **Intelligent-Tiering performance considerations**

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

- Workloads reading data with smaller I/O sizes will require higher concurrency and incur more
  request costs to achieve the same throughput as workloads using large I/O sizes due to higher
  latency from Intelligent-Tiering storage tiers. We recommend configuring your SSD read cache
  large enough to support the higher concurrency and throughput when working with smaller IO
  sizes.
- 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.
- Intelligent-Tiering storage class supports read-ahead to optimize performance for sequential read requests. We recommend configuring your data access pattern sequentially when possible to allow for pre-fetching data and higher performance.

# **Accessing file systems**

Using Amazon FSx, you can burst your compute-intensive workloads from on-premises into the Amazon Web Services Cloud by importing data over Direct Connect or VPN. You can access your Amazon FSx file system from on-premises, copy data into your file system as-needed, and run compute-intensive workloads on in-cloud instances.

In the following section, you can learn how to access your Amazon FSx for Lustre file system on a Linux instance. In addition, you can find how to use the file fstab to automatically remount your file system after any system restarts.

Before you can mount a file system, you must create, configure, and launch your related AWS resources. For detailed instructions, see <u>Getting started with Amazon FSx for Lustre</u>. Next, you can install and configure the Lustre client on your compute instance.

#### **Topics**

- Lustre file system and client kernel compatibility
- · Installing the Lustre client
- Mounting from an Amazon Elastic Compute Cloud instance
- Configuring EFA clients
- Mounting from Amazon Elastic Container Service
- Mounting Amazon FSx file systems from on-premises or a peered Amazon VPC
- Mounting your Amazon FSx file system automatically
- Mounting specific filesets
- Unmounting file systems
- Working with Amazon EC2 Spot Instances

# Lustre file system and client kernel compatibility

We highly recommend using the Lustre version for your FSx for Lustre file system that is compatible with the Linux kernel versions of your client instances.

# **Amazon Linux clients**

| Operating system        | OS<br>version | Minimum<br>kernel<br>version | Maximum<br>kernel<br>version | Lustre<br>client<br>version | Lustre file system version |      |      |
|-------------------------|---------------|------------------------------|------------------------------|-----------------------------|----------------------------|------|------|
|                         |               |                              |                              |                             | 2.10                       | 2.12 | 2.15 |
| Amazon<br>Linux<br>2023 | 6.12          | *                            | *                            | 2.15                        | no                         | yes  | yes  |
|                         | 6.1           | 6.1.79-99<br>.167            | 6.1.79-99<br>.167+           | 2.15                        | no                         | yes  | yes  |
| Amazon<br>Linux 2       | 5.10          | 5.10.144-<br>127.601         | 5.10.144-<br>127.601+        | 2.12                        | yes                        | yes  | yes  |
|                         |               |                              | <5.10.144<br>-127.601        | 2.10                        | yes                        | yes  | no   |
|                         | 5.4           | 5.4.214-1<br>20.368          | 5.4.214-1<br>20.368+         | 2.12                        | yes                        | yes  | yes  |
|                         |               |                              | <5.4.214-<br>120.368         | 2.10                        | yes                        | yes  | no   |
|                         | 4.14          |                              | 4.14.294-<br>220.533+        | 2.12                        | yes                        | yes  | yes  |
|                         |               |                              | <4.14.294<br>-220.533        | 2.10                        | yes                        | yes  | no   |

## **Ubuntu clients**

| Operating system | OS<br>version | Minimum<br>kernel<br>version | Maximum<br>kernel<br>version | Lustre<br>client<br>version | Lustre file system version |      |      |
|------------------|---------------|------------------------------|------------------------------|-----------------------------|----------------------------|------|------|
|                  |               |                              |                              |                             | 2.10                       | 2.12 | 2.15 |
| Ubuntu           | 24            | 6.14.0-10<br>12              | 6.14.0*                      | 2.15                        | no                         | yes  | yes  |
|                  |               | 6.8.0-102<br>4               | 6.8.0*                       | 2.15                        | no                         | yes  | yes  |
|                  | 22            | 6.8.0-101<br>7               | 6.8.0*                       | 2.15                        | no                         | yes  | yes  |
|                  |               | 6.5.0-102<br>3               | 6.5.0*                       | 2.15                        | no                         | yes  | yes  |
|                  |               | 6.2.0-101<br>7               | 6.2.0*                       | 2.15                        | no                         | yes  | yes  |
|                  |               | 5.15.0-10<br>15-aws          | 5.15.0-10<br>51-aws          | 2.12                        | yes                        | yes  | yes  |
|                  | 20            | 5.15.0-10<br>15-aws          | 5.15.0*                      | 2.12                        | yes                        | yes  | yes  |
|                  |               | 5.4.0-101<br>1-aws           | 5.13.0-10<br>31-aws          | 2.10                        | yes                        | yes  | no   |

# RHEL/CentOS/Rocky Linux clients

| Operatir<br>system                | OS<br>version | Architec<br>ure | Minimur<br>kernel<br>version | Maximul<br>kernel<br>version | Lustre<br>client<br>version | Lustre file system version |      | version |
|-----------------------------------|---------------|-----------------|------------------------------|------------------------------|-----------------------------|----------------------------|------|---------|
|                                   |               |                 |                              |                              |                             | 2.10                       | 2.12 | 2.15    |
| RHEL/<br>Rock<br>y Linux          | 9.7           | Arm +<br>x86    | 5.14.0-6<br>1.5.1            | 5.14.0-6<br>1*               | 2.15                        | no                         | yes  | yes     |
|                                   | 9.6           | Arm +<br>x86    | 5.14.0-5<br>0.12.1           | 5.14.0-5<br>0*               | 2.15                        | no                         | yes  | yes     |
|                                   | 9.5           | Arm +<br>x86    | 5.14.0-5<br>3.19.1           | 5.14.0-5<br>3*               | 2.15                        | no                         | yes  | yes     |
|                                   | 9.4           | Arm +<br>x86    | 5.14.0-4<br>7.13.1           | 5.14.0-4<br>7*               | 2.15                        | no                         | yes  | yes     |
|                                   | 9.3           | Arm +<br>x86    | 5.14.0-3<br>2.18.1           | 5.14.0-3<br>2.18.1           | 2.15                        | no                         | yes  | yes     |
|                                   | 9.0           | Arm +<br>x86    | 5.14.0-7<br>.13.1            | 5.14.0-7<br>.30.1            | 2.15                        | no                         | yes  | yes     |
| RHEL/<br>Cent<br>OS/<br>Rocky Lin | 8.10          | Arm +<br>x86    | 4.18.0-5<br>3                | 4.18.0-5<br>3*               | 2.12                        | yes                        | yes  | yes     |
|                                   | 8.9           | Arm +<br>x86    | 4.18.0-5<br>3*               | 4.18.0-5<br>3*               | 2.12                        | yes                        | yes  | yes     |
|                                   | 8.8           | Arm +<br>x86    | 4.18.0-4<br>7*               | 4.18.0-4<br>7*               | 2.12                        | yes                        | yes  | yes     |

| Operatir<br>system  | OS<br>version | Architec<br>ure | Minimur<br>kernel<br>version | Maximul<br>kernel<br>version | Lustre<br>client<br>version | Lustre fi | le system | version |
|---------------------|---------------|-----------------|------------------------------|------------------------------|-----------------------------|-----------|-----------|---------|
|                     | 8.7           | Arm +<br>x86    | 4.18.0-4<br>5*               | 4.18.0-4<br>5*               | 2.12                        | yes       | yes       | yes     |
|                     | 8.6           | Arm +<br>x86    | 4.18.0-3°<br>2*              | 4.18.0-3°<br>2*              | 2.12                        | yes       | yes       | yes     |
|                     | 8.5           | Arm +<br>x86    | 4.18.0-3<br>8*               | 4.18.0-3 <sup>-</sup><br>8*  | 2.12                        | yes       | yes       | yes     |
|                     | 8.4           | Arm +<br>x86    | 4.18.0-3<br>5*               | 4.18.0-3<br>5*               | 2.12                        | yes       | yes       | yes     |
| RHEL/<br>Cent<br>OS | 8.3           | Arm +<br>x86    | 4.18.0-2<br>0*               | 4.18.0-2<br>0*               | 2.10                        | yes       | yes       | no      |
|                     | 8.2           | Arm +<br>x86    | 4.18.0-1<br>3*               | 4.18.0-1<br>3*               | 2.10                        | yes       | yes       | no      |
|                     | 7.9           | x86             | 3.10.0-1<br>60*              | 3.10.0-1<br>60*              | 2.12                        | yes       | yes       | yes     |
|                     | 7.8           | x86             | 3.10.0-1<br>27*              | 3.10.0-1<br>27*              | 2.10                        | yes       | yes       | no      |
|                     | 7.7           | x86             |                              | 3.10.0-1<br>62*              | 2.10                        | yes       | yes       | no      |
| CentOS              | 7.9           | Arm             | 4.18.0-1<br>3*               | 4.18.0-1<br>3*               | 2.12                        | yes       | yes       | yes     |
|                     | 7.8           | Arm             | 4.18.0-1<br>7*               | 4.18.0-1<br>7*               | 2.12                        | yes       | yes       | yes     |

# **Installing the Lustre client**

To mount your Amazon FSx for Lustre file system from a Linux instance, first install the opensource Lustre client. Then, depending on your operating system version, use one of the following procedures. For kernel support information, see Lustre file system and client kernel compatibility.

If your compute instance isn't running the Linux kernel specified in the installation instructions, and you can't change the kernel, you can build your own Lustre client. For more information, see <a href="Compiling Lustre">Compiling Lustre</a> on the Lustre Wiki.

#### **Amazon Linux**

#### To install the Lustre client on Amazon Linux 2023

- 1. Open a terminal on your client.
- 2. Determine which kernel is currently running on your compute instance by running the following command.

```
uname -r
```

- 3. Review the system response and compare it to the following minimum kernel requirement for installing the Lustre client on Amazon Linux 2023:
  - 6.12 kernel minimum requirement 6.12\*
  - 6.1 kernel minimum requirement 6.1.79-99.167.amzn2023

If your EC2 instance meets the minimum kernel requirement, proceed to the step and install the Lustre client.

If the command returns a result less than the kernel minimum requirement, update the kernel and reboot your Amazon EC2 instance by running the following command.

```
sudo dnf -y update kernel && sudo reboot
```

Confirm that the kernel has been updated using the **uname -r** command.

4. Download and install the Lustre client with the following command.

```
sudo dnf install -y lustre-client
```

Installing the Lustre client 143

#### To install the Lustre client on Amazon Linux 2

- 1. Open a terminal on your client.
- 2. Determine which kernel is currently running on your compute instance by running the following command.

```
uname -r
```

- 3. Review the system response and compare it to the following minimum kernel requirements for installing the Lustre client on Amazon Linux 2:
  - 5.10 kernel minimum requirement 5.10.144-127.601.amzn2
  - 5.4 kernel minimum requirement 5.4.214-120.368.amzn2
  - 4.14 kernel minimum requirement 4.14.294-220.533.amzn2

If your EC2 instance meets the minimum kernel requirements, proceed to the step and install the Lustre client.

If the command returns a result less than the kernel minimum requirement, update the kernel and reboot your Amazon EC2 instance by running the following command.

```
sudo yum -y update kernel && sudo reboot
```

Confirm that the kernel has been updated using the uname -r command.

4. Download and install the Lustre client with the following command.

```
sudo amazon-linux-extras install -y lustre
```

If you are unable to upgrade the kernel to the kernel minimum requirement, you may install the legacy 2.10 client with the following command.

```
sudo amazon-linux-extras install -y lustre2.10
```

#### To install the Lustre client on Amazon Linux

1. Open a terminal on your client.

Amazon Linux 144

2. Determine which kernel is currently running on your compute instance by running the following command. The Lustre client requires Amazon Linux kernel 4.14, version 104 or higher.

```
uname -r
```

- 3. Do one of the following:
  - If the command returns 4.14.104-78.84.amzn1.x86\_64 or a higher version of 4.14, download and install the Lustre client using the following command.

```
sudo yum install -y lustre-client
```

• If the command returns a result less than 4.14.104-78.84.amzn1.x86\_64, update the kernel and reboot your Amazon EC2 instance by running the following command.

```
sudo yum -y update kernel && sudo reboot
```

Confirm that the kernel has been updated using the **uname -r** command. Then download and install the Lustre client as described previously.

# CentOS, Rocky Linux, and Red Hat

## To install the Lustre client on Red Hat and Rocky Linux 9.0 or 9.3–9.7

You can install and update Lustre client packages that are compatible with Red Hat Enterprise Linux (RHEL) and Rocky Linux from the Amazon FSx Lustre client yum package repository. These packages are signed to help ensure that they have not been tampered with before or during download. The repository installation fails if you don't install the corresponding public key on your system.

## To add the Amazon FSx Lustre client yum package repository

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key by using the following command.
- 3. Import the key by using the following command.
- 4. Add the repository and update the package manager using the following command.

#### To configure the Amazon FSx Lustre client yum repository

The Amazon FSx Lustre client yum package repository is configured by default to install the Lustre client that is compatible with the kernel version that initially shipped with the latest supported Rocky Linux and RHEL 9 release. To install a Lustre client that is compatible with the kernel version you are using, you can edit the repository configuration file.

This section describes how to determine which kernel you are running, whether you need to edit the repository configuration, and how to edit the configuration file.

 Determine which kernel is currently running on your compute instance by using the following command.

```
uname -r
```

- 2. Do one of the following:
  - If the command returns 5.14.0-611\*, you don't need to modify the repository configuration. Continue to the **To install the Lustre client** procedure.
  - If the command returns 5.14.0-570\*, you must edit the repository configuration so that it points to the Lustre client for the Rocky Linux and RHEL 9.6 release.
  - If the command returns 5.14.0-503\*, you must edit the repository configuration so that it points to the Lustre client for the Rocky Linux and RHEL 9.5 release.
  - If the command returns 5.14.0-427\*, you must edit the repository configuration so that it points to the Lustre client for the Rocky Linux and RHEL 9.4 release.
  - If the command returns 5.14.0-362.18.1, you must edit the repository configuration so that it points to the Lustre client for the Rocky Linux and RHEL 9.3 release.
  - If the command returns 5.14.0-70\*, you must edit the repository configuration so that it points to the Lustre client for the Rocky Linux and RHEL 9.0 release.
- 3. Edit the repository configuration file to point to a specific version of RHEL using the following command. Replace specific\_RHEL\_version with the RHEL version you need to use.

```
sudo sed -i 's#9#specific_RHEL_version#' /etc/yum.repos.d/aws-fsx.repo
```

For example, to point to release 9.6, substitute *specific\_RHEL\_version* with 9.6 in the command, as in the following example.

```
sudo sed -i 's#9#9.6#' /etc/yum.repos.d/aws-fsx.repo
```

4. Use the following command to clear the yum cache.

```
sudo yum clean all
```

#### To install the Lustre client

• Install the packages from the repository using the following command.

```
sudo yum install -y kmod-lustre-client lustre-client
```

#### Additional information (Rocky Linux and Red Hat 9.0 and newer)

The commands preceding install the two packages that are necessary for mounting and interacting with your Amazon FSx file system. The repository includes additional Lustre packages, such as a package containing the source code and packages containing tests, and you can optionally install them. To list all available packages in the repository, use the following command.

```
yum --disablerepo="*" --enablerepo="aws-fsx" list available
```

To download the source rpm, containing a tarball of the upstream source code and the set of patches that we've applied, use the following command.

```
sudo yumdownloader --source kmod-lustre-client
```

When you run yum update, a more recent version of the module is installed if available and the existing version is replaced. To prevent the currently installed version from being removed on update, add a line like the following to your /etc/yum.conf file.

This list includes the default install only packages, specified in the yum.conf man page, and the kmod-lustre-client package.

#### To install the Lustre client on CentOS and Red Hat 8.2-8.10 or on Rocky Linux 8.4-8.10

You can install and update Lustre client packages that are compatible with Red Hat Enterprise Linux (RHEL), Rocky Linux, and CentOS from the Amazon FSx Lustre client yum package repository. These packages are signed to help ensure that they have not been tampered with before or during download. The repository installation fails if you don't install the corresponding public key on your system.

#### To add the Amazon FSx Lustre client yum package repository

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key by using the following command.
- 3. Import the key by using the following command.
- 4. Add the repository and update the package manager using the following command.

#### To configure the Amazon FSx Lustre client yum repository

The Amazon FSx Lustre client yum package repository is configured by default to install the Lustre client that is compatible with the kernel version that initially shipped with the latest supported CentOS, Rocky Linux, and RHEL 8 release. To install a Lustre client that is compatible with the kernel version you are using, you can edit the repository configuration file.

This section describes how to determine which kernel you are running, whether you need to edit the repository configuration, and how to edit the configuration file.

 Determine which kernel is currently running on your compute instance by using the following command.

uname -r

- 2. Do one of the following:
  - If the command returns 4.18.0-553\*, you don't need to modify the repository configuration. Continue to the **To install the Lustre client** procedure.
  - If the command returns 4.18.0-513\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.9 release.
  - If the command returns 4.18.0-477\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.8 release.

• If the command returns 4.18.0-425\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.7 release.

- If the command returns 4.18.0-372\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.6 release.
- If the command returns 4.18.0-348\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.5 release.
- If the command returns 4.18.0-305\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS, Rocky Linux, and RHEL 8.4 release.
- If the command returns 4.18.0-240\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS and RHEL 8.3 release.
- If the command returns 4.18.0-193\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS and RHEL 8.2 release.
- 3. Edit the repository configuration file to point to a specific version of RHEL using the following command.

```
sudo sed -i 's#8#specific_RHEL_version#' /etc/yum.repos.d/aws-fsx.repo
```

For example, to point to release 8.9, substitute *specific\_RHEL\_version* with 8.9 in the command.

```
sudo sed -i 's#8#8.9#' /etc/yum.repos.d/aws-fsx.repo
```

4. Use the following command to clear the yum cache.

```
sudo yum clean all
```

#### To install the Lustre client

• Install the packages from the repository using the following command.

```
sudo yum install -y kmod-lustre-client lustre-client
```

#### Additional information (CentOS, Rocky Linux, and Red Hat 8.2 and newer)

The commands preceding install the two packages that are necessary for mounting and interacting with your Amazon FSx file system. The repository includes additional Lustre packages, such as a package containing the source code and packages containing tests, and you can optionally install them. To list all available packages in the repository, use the following command.

```
yum --disablerepo="*" --enablerepo="aws-fsx" list available
```

To download the source rpm, containing a tarball of the upstream source code and the set of patches that we've applied, use the following command.

```
sudo yumdownloader --source kmod-lustre-client
```

When you run yum update, a more recent version of the module is installed if available and the existing version is replaced. To prevent the currently installed version from being removed on update, add a line like the following to your /etc/yum.conf file.

This list includes the default install only packages, specified in the yum. conf man page, and the kmod-lustre-client package.

#### To install the Lustre client on CentOS and Red Hat 7.7, 7.8, or 7.9 (x86\_64 instances)

You can install and update Lustre client packages that are compatible with Red Hat Enterprise Linux (RHEL) and CentOS from the Amazon FSx Lustre client yum package repository. These packages are signed to help ensure they have not been tampered with before or during download. The repository installation fails if you don't install the corresponding public key on your system.

#### To add the Amazon FSx Lustre client yum package repository

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key using the following command.
- 3. Import the key using the following command.
- 4. Add the repository and update the package manager using the following command.

#### To configure the Amazon FSx Lustre client yum repository

The Amazon FSx Lustre client yum package repository is configured by default to install the Lustre client that is compatible with the kernel version that initially shipped with the latest supported CentOS and RHEL 7 release. To install a Lustre client that is compatible with the kernel version you are using, you can edit the repository configuration file.

This section describes how to determine which kernel you are running, whether you need to edit the repository configuration, and how to edit the configuration file.

 Determine which kernel is currently running on your compute instance by using the following command.

```
uname -r
```

- 2. Do one of the following:
  - If the command returns 3.10.0-1160\*, you don't need to modify the repository configuration. Continue to the **To install the Lustre client** procedure.
  - If the command returns 3.10.0-1127\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS and RHEL 7.8 release.
  - If the command returns 3.10.0-1062\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS and RHEL 7.7 release.
- 3. Edit the repository configuration file to point to a specific version of RHEL using the following command.

```
sudo sed -i 's#7#specific_RHEL_version#' /etc/yum.repos.d/aws-fsx.repo
```

To point to release 7.8, substitute *specific\_RHEL\_version* with 7.8 in the command.

```
sudo sed -i 's#7#7.8#' /etc/yum.repos.d/aws-fsx.repo
```

To point to release 7.7, substitute *specific\_RHEL\_version* with 7.7 in the command.

```
sudo sed -i 's#7#7.7#' /etc/yum.repos.d/aws-fsx.repo
```

4. Use the following command to clear the yum cache.

sudo yum clean all

#### To install the Lustre client

• Install the Lustre client packages from the repository using the following command.

```
sudo yum install -y kmod-lustre-client lustre-client
```

#### Additional information (CentOS and Red Hat 7.7 and newer)

The commands preceding install the two packages that are necessary for mounting and interacting with your Amazon FSx file system. The repository includes additional Lustre packages, such as a package containing the source code and packages containing tests, and you can optionally install them. To list all available packages in the repository, use the following command.

```
yum --disablerepo="*" --enablerepo="aws-fsx" list available
```

To download the source rpm containing a tarball of the upstream source code and the set of patches that we've applied, use the following command.

```
sudo yumdownloader --source kmod-lustre-client
```

When you run yum update, a more recent version of the module is installed if available, and the existing version is replaced. To prevent the currently installed version from being removed on update, add a line like the following to your /etc/yum.conf file.

This list includes the default install only packages, specified in the yum. conf man page, and the kmod-lustre-client package.

#### To install the Lustre client on CentOS 7.8 or 7.9 (Arm-based AWS Graviton-powered instances)

You can install and update Lustre client packages from the Amazon FSx Lustre client yum package repository that are compatible with CentOS 7 for Arm-based AWS Graviton-powered EC2 instances. These packages are signed to help ensure they have not been tampered with before or during download. The repository installation fails if you don't install the corresponding public key on your system.

#### To add the Amazon FSx Lustre client yum package repository

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key using the following command.
- 3. Import the key using the following command.
- 4. Add the repository and update the package manager using the following command.

#### To configure the Amazon FSx Lustre client yum repository

The Amazon FSx Lustre client yum package repository is configured by default to install the Lustre client that is compatible with the kernel version that initially shipped with the latest supported CentOS 7 release. To install a Lustre client that is compatible with the kernel version you are using, you can edit the repository configuration file.

This section describes how to determine which kernel you are running, whether you need to edit the repository configuration, and how to edit the configuration file.

 Determine which kernel is currently running on your compute instance by using the following command.

uname -r

- 2. Do one of the following:
  - If the command returns 4.18.0-193\*, you don't need to modify the repository configuration. Continue to the **To install the Lustre client** procedure.
  - If the command returns 4.18.0-147\*, you must edit the repository configuration so that it points to the Lustre client for the CentOS 7.8 release.
- 3. Edit the repository configuration file to point to the CentOS 7.8 release using the following command.

```
sudo sed -i 's#7#7.8#' /etc/yum.repos.d/aws-fsx.repo
```

4. Use the following command to clear the yum cache.

```
sudo yum clean all
```

#### To install the Lustre client

• Install the packages from the repository using the following command.

```
sudo yum install -y kmod-lustre-client lustre-client
```

#### Additional information (CentOS 7.8 or 7.9 for Arm-based AWS Graviton-powered EC2 instances)

The commands preceding install the two packages that are necessary for mounting and interacting with your Amazon FSx file system. The repository includes additional Lustre packages, such as a package containing the source code and packages containing tests, and you can optionally install them. To list all available packages in the repository, use the following command.

```
yum --disablerepo="*" --enablerepo="aws-fsx" list available
```

To download the source rpm, containing a tarball of the upstream source code and the set of patches that we've applied, use the following command.

```
sudo yumdownloader --source kmod-lustre-client
```

When you run yum update, a more recent version of the module is installed if available, and the existing version is replaced. To prevent the currently installed version from being removed on update, add a line like the following to your /etc/yum.conf file.

This list includes the default install only packages, specified in the yum.conf man page, and the kmod-lustre-client package.

#### Ubuntu

#### To install the Lustre client on Ubuntu 18.04, 20.04, 22.04, or 24.04

You can get Lustre packages from the Amazon FSx Ubuntu repository. To validate that the contents of the repository have not been tampered with before or during download, a GNU Privacy Guard (GPG) signature is applied to the metadata of the repository. Installing the repository fails unless you have the correct public GPG key installed on your system.

- 1. Open a terminal on your client.
- 2. Follow these steps to add the Amazon FSx Ubuntu repository:
  - a. If you have not previously registered an Amazon FSx Ubuntu repository on your client instance, download and install the required public key. Use the following command.
  - Add the Amazon FSx package repository to your local package manager using the following command.
- Determine which kernel is currently running on your client instance, and update as needed. For a list of the required kernels for the Lustre client on Ubuntu for both x86-based EC2 instances and Arm-based EC2 instances powered by AWS Graviton processors, see Ubuntu clients.
  - a. Run the following command to determine which kernel is running.

```
uname -r
```

b. Run the following command to update to the latest Ubuntu kernel and Lustre version and then reboot.

```
sudo apt install -y linux-aws lustre-client-modules-aws && sudo reboot
```

If your kernel version is greater than the minimum kernel version for both x86-based EC2 instances and Graviton-based EC2 instances, and you don't want to update to the latest kernel version, you can install Lustre for the current kernel with the following command.

```
sudo apt install -y lustre-client-modules-$(uname -r)
```

The two Lustre packages that are necessary for mounting and interacting with your FSx for Lustre file system are installed. You can optionally install additional related packages

Ubuntu 155

such as a package containing the source code and packages containing tests that are included in the repository.

c. List all available packages in the repository by using the following command.

```
sudo apt-cache search ^lustre
```

d. (Optional) If you want your system upgrade to also always upgrade Lustre client modules, make sure that the lustre-client-modules-aws package is installed using the following command.

```
sudo apt install -y lustre-client-modules-aws
```



If you get a Module Not Found error, see To troubleshoot missing module errors.

#### To troubleshoot missing module errors

If you get a Module Not Found error while installing on any version of Ubuntu, do the following:

Downgrade your kernel to the latest supported version. List all available versions of the lustreclient-modules package and install the corresponding kernel. To do this, use the following command.

```
sudo apt-cache search lustre-client-modules
```

For example, if the latest version that is included in the repository is lustre-client-modules-5.4.0-1011-aws, do the following:

1. Install the kernel that this package was built for using the following commands.

```
sudo apt-get install -y linux-image-5.4.0-1011-aws
```

```
sudo sed -i 's/GRUB_DEFAULT=.\+/GRUB\_DEFAULT="Advanced options for Ubuntu>Ubuntu,
with Linux 5.4.0-1011-aws"/' /etc/default/grub
```

Ubuntu 156

```
sudo update-grub
```

2. Reboot your instance using the following command.

```
sudo reboot
```

3. Install the Lustre client using the following command.

```
sudo apt-get install -y lustre-client-modules-$(uname -r)
```

### **SUSE Linux**

To install the Lustre client on SUSE Linux 12 SP3, SP4, or SP5

#### To install the Lustre client on SUSE Linux 12 SP3

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key by using the following command.
- 3. Import the key by using the following command.

```
sudo rpm --import fsx-sles-public-key.asc
```

- 4. Add the repository for the Lustre client using the following command.
- 5. Download and install the Lustre client with the following commands.

```
sudo zypper ar --gpgcheck-strict fsx-lustre-client.repo
sudo sed -i 's#SLES-12#SP3#' /etc/zypp/repos.d/aws-fsx.repo
sudo zypper refresh
sudo zypper in lustre-client
```

#### To install the Lustre client on SUSE Linux 12 SP4

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key by using the following command.
- 3. Import the key by using the following command.

SUSE Linux 157

```
sudo rpm --import fsx-sles-public-key.asc
```

- 4. Add the repository for the Lustre client using the following command.
- 5. Do one of the following:
  - If you installed SP4 directly, download and install the Lustre client with the following commands.

```
sudo zypper ar --gpgcheck-strict fsx-lustre-client.repo
sudo sed -i 's#SLES-12#SP4#' /etc/zypp/repos.d/aws-fsx.repo
sudo zypper refresh
sudo zypper in lustre-client
```

• If you migrated from SP3 to SP4 and previously added the Amazon FSx repository for SP3, download and install the Lustre client with the following commands.

```
sudo zypper ar --gpgcheck-strict fsx-lustre-client.repo
sudo sed -i 's#SP3#SP4#' /etc/zypp/repos.d/aws-fsx.repo
sudo zypper ref
sudo zypper up --force-resolution lustre-client-kmp-default
```

#### To install the Lustre client on SUSE Linux 12 SP5

- 1. Open a terminal on your client.
- 2. Install the Amazon FSx rpm public key by using the following command.
- 3. Import the key by using the following command.

```
sudo rpm --import fsx-sles-public-key.asc
```

- 4. Add the repository for the Lustre client using the following command.
- 5. Do one of the following:
  - If you installed SP5 directly, download and install the Lustre client with the following commands.

```
sudo zypper ar --gpgcheck-strict fsx-lustre-client.repo
sudo zypper refresh
sudo zypper in lustre-client
```

SUSE Linux 158

• If you migrated from SP4 to SP5 and previously added the Amazon FSx repository for SP4, download and install the Lustre client with the following commands.

```
sudo sed -i 's#SP4#SLES-12' /etc/zypp/repos.d/aws-fsx.repo
sudo zypper ref
sudo zypper up --force-resolution lustre-client-kmp-default
```



#### Note

You might need to reboot your compute instance for the client to finish installing.

# Mounting from an Amazon Elastic Compute Cloud instance

You can mount your file system from an Amazon EC2 instance.

#### To mount your file system from Amazon EC2

- 1. Connect to your Amazon EC2 instance.
- Make a directory on your FSx for Lustre file system for the mount point with the following 2. command.

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

- Mount the Amazon FSx for Lustre file system to the directory that you created. Use the following command and replace the following items:
  - Replace file\_system\_dns\_name with the actual file system's DNS name.
  - Replace mount name with the file system's mount name. This mount name is returned in the CreateFileSystem API operation response. It's also returned in the response of the describe-file-systems AWS CLI command, and the DescribeFileSystems API operation.

```
sudo mount -t lustre -o relatime,flock file_system_dns_name@tcp://mountname /fsx
```

This command mounts your file system with two options, -o relatime and flock:

Mount from Amazon EC2 159

• relatime – While the atime option maintains atime (inode access times) data for each time a file is accessed, the relatime option also maintains atime data, but not for each time that a file is accessed. With the relatime option enabled, atime data is written to disk only if the file has been modified since the atime data was last updated (mtime), or if the file was last accessed more than a certain amount of time ago (6 hours by default). Using either the relatime or a time option will optimize the file release processes.



#### Note

If your workload requires precise access time accuracy, you can mount with the atime mount option. However, doing so can impact workload performance by increasing the network traffic required to maintain precise access time values. If your workload does not require metadata access time, using the noatime mount option to disable updates to access time can provide a performance gain. Be aware that atime focused processes like file release or releasing data validity will be inaccurate in their release.

- flock Enables file locking for your file system. If you don't want file locking enabled, use the mount command without flock.
- Verify that the mount command was successful by listing the contents of the directory to which you mounted the file system, /mnt/fsx by using the following command.

```
$ ls /fsx
import-path lustre
```

You can also use the df command, following.

```
$ df
Filesystem
                              1K-blocks
                                           Used Available Use% Mounted on
devtmpfs
                                1001808
                                               0
                                                    1001808
                                                              0% /dev
tmpfs
                                                    1019760
                                                              0% /dev/shm
                                1019760
                                               0
tmpfs
                                                    1019368
                                                              1% /run
                                1019760
                                             392
tmpfs
                                               0
                                                              0% /sys/fs/cgroup
                                1019760
                                                    1019760
/dev/xvda1
                                8376300 1263180
                                                    7113120 16% /
                                                              1% /fsx
123.456.789.0@tcp:/mountname 3547698816
                                           13824 3547678848
tmpfs
                                               0
                                                     203956
                                                              0% /run/user/1000
                                 203956
```

Mount from Amazon EC2 160

The results show the Amazon FSx file system mounted on /fsx.

# **Configuring EFA clients**

Use the following procedures to set up your Lustre client to access an FSx for Lustre file system via Elastic Fabric Adapter (EFA).

EFA is supported on Lustre clients running the following operating systems:

- Amazon Linux 2023 (AL2023)
- Red Hat Enterprise Linux (RHEL) 9.5 or newer
- Ubuntu 22.04 or newer with kernel version 6.8+

EFA is supported on Lustre clients listed below. For more information, see Installing the Lustre client.

EFA is supported on Nitro v4 (or higher) EC2 instances that support EFA, excluding the trn2 instance family. See Supported instance types in the Amazon EC2 User Guide.

#### **Topics**

- Step 1: Install required drivers
- Step 2: Configure EFA for the Lustre client
- Step 3: EFA interfaces

## **Step 1: Install required drivers**



#### Note

If you are using a Deep Learning AMI, you can skip this step as both the EFA driver and NVIDIA GPUDirect Storage (GDS) driver are pre-installed.

#### Install the EFA driver

Follow the instructions in Step 3: Install the EFA software in the Amazon EC2 User Guide.

Configure EFA clients 161

## Install the GDS driver (optional)

This step is only required if you plan to use NVIDIA GPUDirect Storage (GDS) with FSx for Lustre.

#### Requirements:

- Amazon EC2 P5, P5e, P5en, P6-B200, or P6e-GB200 instance
- NVIDIA GDS driver version 2.24.2 or higher

#### To install the NVIDIA GPUDirect Storage driver on your client instance

1. Clone the NVIDIA GDS repository:

```
git clone https://github.com/NVIDIA/gds-nvidia-fs.git
```

Build and install the driver:

```
cd gds-nvidia-fs/src/
export NVFS_MAX_PEER_DEVS=128
export NVFS_MAX_PCI_DEPTH=16
sudo -E make
sudo insmod nvidia-fs.ko
```

# **Step 2: Configure EFA for the Lustre client**

To access an FSx for Lustre file system using an EFA interface, you must install the Lustre EFA modules and configure EFA interfaces.

## **Quick setup**

#### To quickly configure your Lustre client

- 1. Connect to your Amazon EC2 instance.
- 2. Download and unzip the file containing the configuration script:

```
curl -0 https://docs.aws.amazon.com/fsx/latest/LustreGuide/samples/configure-efa-
fsx-lustre-client.zip
unzip configure-efa-fsx-lustre-client.zip
```

3. Change to the configure-efa-fsx-lustre-client folder and run the setup script:

```
cd configure-efa-fsx-lustre-client
sudo ./setup.sh
```

The script automatically does the following:

- Imports Lustre modules
- Configures TCP and EFA interfaces
- Creates a systemd service for automatic configuration on reboot

For a list of options and usage examples you can use with the setup.sh script, see the README.md file in the zip file.

## Managing the systemd service manually

The systemd service file is created at /etc/systemd/system/configure-efa-fsx-lustre-client.service. The following are some helpful systemd-related commands:

```
# Check status
sudo systemctl status configure-efa-fsx-lustre-client.service

# View logs
sudo journalctl -u configure-efa-fsx-lustre-client.service
# View warnings/errors from dmesg
sudo dmesg
```

For more information, see the README.md file in the zip file.

## **Auto-mount configuration (optional)**

For information on automatically mounting your Amazon FSx for Lustre file system on boot, see Mounting your Amazon FSx file system automatically.

# **Step 3: EFA interfaces**

Each FSx for Lustre file system has a maximum limit of 1024 EFA connections across all client instances.

Step 3: EFA interfaces 163

The configure-efa-fsx-lustre-client.sh script automatically configures EFA interfaces based on the instance type.

| Instance Type                               | Default Number of EFA Interfaces |
|---|----------------------------------|
| p6e-gb200.36xlarge                          | 8                                |
| p6-b200.48xlarge                            | 8                                |
| p5en.48xlarge                               | 8                                |
| p5e.48xlarge                                | 8                                |
| p5.48xlarge                                 | 8                                |
| Other instances with multiple network cards | 2                                |
| Other instances with a single network card  | 1                                |

Each configured EFA interface on a client instance counts as one connection against the 1024 EFA connection limit when connected to an FSx for Lustre file system.

## Managing EFA interfaces manually

Instances with more EFA interfaces typically support higher throughput. You can customize the number of interfaces to optimize performance for your specific workloads, as long as you stay within the total EFA connection limit.

You can manually manage EFA interfaces using the following commands:

#### 1. View available EFA devices:

```
for interface in /sys/class/infiniband/*; do
   if [ ! -e "$interface/device/driver" ]; then continue; fi
   driver=$(basename "$(realpath "$interface/device/driver")")
   if [ "$driver" != "efa" ]; then continue; fi
   echo $(basename $interface)
done
```

Step 3: EFA interfaces 164

2. View currently configured interfaces:

```
sudo lnetctl net show
```

3. Add an EFA interface:

```
sudo lnetctl net add --net efa --if device_name -peer-credits 32
```

Replace device\_name with an actual device name from the list in step 1.

4. Remove an EFA interface:

```
sudo lnetctl net del --net efa --if device_name
```

Replace device\_name with an actual device name from the list in step 2.

# **Mounting from Amazon Elastic Container Service**

You can access your FSx for Lustre file system from an Amazon Elastic Container Service (Amazon ECS) Docker container on an Amazon EC2 instance. You can do so by using either of the following options:

- By mounting your FSx for Lustre file system from the Amazon EC2 instance that is hosting your Amazon ECS tasks, and exporting this mount point to your containers.
- 2. By mounting the file system directly inside your task container.

For more information about Amazon ECS, see <u>What is Amazon Elastic Container Service?</u> in the *Amazon Elastic Container Service Developer Guide*.

We recommend using option 1 (<u>Mounting from an Amazon EC2 instance hosting Amazon ECS</u> <u>tasks</u>) because it provides better resource use, especially if you start many containers (more than five) on the same EC2 instance or if your tasks are short-lived (less than 5 minutes).

Use option 2 (<u>Mounting from a Docker container</u>), if you're unable to configure the EC2 instance, or if your application requires the container's flexibility.

Mounting from Amazon ECS 165



#### Note

Mounting FSx for Lustre on an AWS Fargate launch type isn't supported.

The following sections describe the procedures for each of the options for mounting your FSx for Lustre file system from an Amazon ECS container.

#### **Topics**

- Mounting from an Amazon EC2 instance hosting Amazon ECS tasks
- Mounting from a Docker container

## Mounting from an Amazon EC2 instance hosting Amazon ECS tasks

This procedure shows how you can configure an Amazon ECS on EC2 instance to locally mount your FSx for Lustre file system. The procedure uses volumes and mountPoints container properties to share the resource and make this file system accessible to locally running tasks. For more information, see Launching an Amazon ECS Container Instance in the Amazon Elastic Container Service Developer Guide.

This procedure is for an Amazon ECS-Optimized Amazon Linux 2 AMI. If you are using another Linux distribution, see Installing the Lustre client.

#### To mount your file system from Amazon ECS on an EC2 instance

- When launching Amazon ECS instances, either manually or using an Auto Scaling group, add the lines in the following code example to the end of the User data field. Replace the following items in the example:
  - Replace file system dns name with the actual file system's DNS name.
  - Replace mount name with the file system's mount name.
  - Replace *mountpoint* with the file system's mount point, which you need to create.

```
#!/bin/bash
...<existing user data>...
```

```
fsx_dnsname=file_system_dns_name
fsx_mountname=mountname
fsx_mountpoint=mountpoint
amazon-linux-extras install -y lustre
mkdir -p "$fsx_mountpoint"
mount -t lustre ${fsx_dnsname}@tcp:/${fsx_mountname} ${fsx_mountpoint} -o
relatime,flock
```

2. When creating your Amazon ECS tasks, add the following volumes and mountPoints container properties in the JSON definition. Replace *mountpoint* with the file system's mount point (such as /mnt/fsx).

```
{
    "volumes": [
           {
                  "host": {
                       "sourcePath": "mountpoint"
                  },
                  "name": "Lustre"
           }
    ],
    "mountPoints": [
           {
                  "containerPath": "mountpoint",
                  "sourceVolume": "Lustre"
           }
    ],
}
```

## Mounting from a Docker container

The following procedure shows how you can configure an Amazon ECS task container to install the lustre-client package and mount your FSx for Lustre file system in it. The procedure uses an Amazon Linux (amazonlinux) Docker image, but a similar approach can work for other distributions.

#### To mount your file system from a Docker container

1. On your Docker container, install the lustre-client package and mount your FSx for Lustre file system with the command property. Replace the following items in the example:

- Replace file\_system\_dns\_name with the actual file system's DNS name.
- Replace *mount name* with the file system's mount name.
- Replace mount point with the file system's mount point.

```
"command": [
  "/bin/sh -c \"amazon-linux-extras install -y lustre; mount -t
  lustre file_system_dns_name@tcp:/mountname mountpoint -o relatime,flock;\""
],
```

2. Add SYS\_ADMIN capability to your container to authorize it to mount your FSx for Lustre file system, using the linuxParameters property.

# Mounting Amazon FSx file systems from on-premises or a peered Amazon VPC

You can access your Amazon FSx file system in two ways. One is from Amazon EC2 instances located in an Amazon VPC that's peered to the file system's VPC. The other is from on-premises clients that are connected to your file system's VPC using Direct Connect or VPN.

You connect the client's VPC and your Amazon FSx file system's VPC using either a VPC peering connection or a VPC transit gateway. When you use a VPC peering connection or transit gateway to connect VPCs, Amazon EC2 instances that are in one VPC can access Amazon FSx file systems in another VPC, even if the VPCs belong to different accounts.

Before using the following the procedure, you need to set up either a VPC peering connection or a VPC transit gateway.

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

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

You can mount your file system from outside its VPC using the IP address of its primary network interface. The primary network interface is the first network interface returned when you run the aws fsx describe-file-systems AWS CLI command. You can also get this IP address from the Amazon Web Services Management Console.

The following table illustrates IP address requirements for accessing Amazon FSx file systems using a client that's outside of the file system's VPC.

| For clients located in                                      | Access to file systems<br>created before<br>December 17, 2020                 | Access to file systems created on or after December 17, 2020 |
|---|---|--|
| Peered VPCs using VPC Peering or AWS Transit Gateway        | Clients with IP addresses in an <u>RFC 1918</u> private IP address range:     | ✓  |
| Peered networks using Direct<br>Connect or Site-to-Site VPN | <ul> <li>10.0.0.0/8</li> <li>172.16.0.0/12</li> <li>192.168.0.0/16</li> </ul> | ✓  |

If you need to access your Amazon FSx file system that was created before December 17, 2020 using a non-private IP address range, you can create a new file system by restoring a backup of the file system. For more information, see Protecting your data with backups.

#### To retrieve the IP address of the primary network interface for a file system

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. In the navigation pane, choose **File systems**.

- Choose your file system from the dashboard. 3.
- From the file system details page, choose **Network & security**. 4.
- For **Network interface**, choose the ID for your primary elastic network interface. Doing this 5. takes you to the Amazon EC2 console.

On the **Details** tab, find the **Primary private IPv4 IP**. This is the IP address for your primary network interface.



#### Note

You can't use Domain Name System (DNS) name resolution when mounting an Amazon FSx file system from outside the VPC it is associated with.

# Mounting your Amazon FSx file system automatically

You can update the /etc/fstab file in your Amazon EC2 instance after you connect to the instance for the first time so that it mounts your Amazon FSx file system each time it reboots.

# Using /etc/fstab to mount FSx for Lustre automatically

To automatically mount your Amazon FSx file system directory when the Amazon EC2 instance reboots, you can use the fstab file. The fstab file contains information about file systems. The command mount -a, which runs during instance startup, mounts the file systems listed in the fstab file.

## Note

- Before you can update the /etc/fstab file of your EC2 instance, make sure that you've already created your Amazon FSx file system. For more information, see Step 1: Create your FSx for Lustre file system in the Getting Started exercise.
- For EFA-enabled file systems, configuring systemd is a pre-requisite. For more information, see Quick setup.

### To update the /etc/fstab file in your EC2 instance

- 1. Connect to your EC2 instance, and open the /etc/fstab file in an editor.
- 2. Add the following line to the /etc/fstab file.

Mount the Amazon FSx for Lustre file system to the directory that you created. Use the following command and replace the following:

- Replace /fsx with the directory that you want to mount your Amazon FSx file system to.
- Replace file\_system\_dns\_name with the actual file system's DNS name.
- Replace mountname with the file system's mount name. This mount name is returned in the CreateFileSystem API operation response. It's also returned in the response of the describe-file-systems AWS CLI command, and the DescribeFileSystems API operation.

### For non-EFA file systems:

```
file_system_dns_name@tcp:/mountname /fsx lustre defaults,relatime,flock,_netdev,x-
systemd.automount,x-systemd.requires=network.service 0 0
```

### For EFA-enabled file systems:

```
file_system_dns_name@tcp:/mountname /fsx lustre defaults,relatime,flock,_netdev,x-
systemd.automount,x-systemd.requires=configure-efa-fsx-lustre-client.service,x-
systemd.after=configure-efa-fsx-lustre-client.service 0 0
```

### Marning

Use the \_netdev option, used to identify network file systems, when mounting your file system automatically. If \_netdev is missing, your EC2 instance might stop responding. This result is because network file systems need to be initialized after the compute instance starts its networking. For more information, see <u>Automatic</u> mounting fails and the instance is unresponsive.

3. Save the changes to the file.

Your EC2 instance is now configured to mount the Amazon FSx file system whenever it restarts.

Automount using /etc/fstab 171

FSx for Lustre Lustre User Guide



### Note

In some cases, your Amazon EC2 instance might need to start regardless of the status of your mounted Amazon FSx file system. In these cases, add the nofail option to your file system's entry in your /etc/fstab file.

The fields in the line of code that you added to the /etc/fstab file do the following.

| Field                                    | Description   |
|--|---|
| <pre>file_syst em_dns_na me @tcp:/</pre> | The DNS name for your Amazon FSx file system, which identifies the file system. You can get this name from the console or programma tically from the AWS CLI or an AWS SDK.   |
| mountname                                | The mount name for the file system. You can get this name from the console or programmatically from the AWS CLI using the <b>describe-file-systems</b> command or the AWS API or SDK using the <b>DescribeFileSystems</b> operation.  |
| /fsx                                     | The mount point for the Amazon FSx file system on your EC2 instance.  |
| lustre                                   | The type of file system, Amazon FSx.  |
| mount options                            | Mount options for the file system, presented as a comma-separated list of the following options:  |
|  | <ul> <li>defaults – This value tells the operating system to use the<br/>default mount options. You can list the default mount options<br/>after the file system has been mounted by viewing the output of<br/>the mount command.</li> </ul>  |
|  | <ul> <li>relatime – This option maintains atime (inode access times)     data, but not for each time that a file is accessed. With this option     enabled, atime data is written to disk only if the file has been     modified since the atime data was last updated (mtime), or if the     file was last accessed more than a certain amount of time ago (one</li> </ul> |

Automount using /etc/fstab 172

| Field   | Description   |
|---|---|
|   | day by default). If you want to turn off inode access time updates, use the noatime mount option instead.   |
|   | <ul> <li>flock – mounts your file system with file locking enabled. If you<br/>don't want file locking enabled, use the noflock mount option<br/>instead.</li> </ul>  |
|   | <ul> <li>_netdev – The value tells the operating system that the file<br/>system resides on a device that requires network access. This<br/>option prevents the instance from mounting the file system until<br/>the network has been enabled on the client.</li> </ul> |
| <pre>x-systemd .automount,x- systemd.requir</pre>   | These options for non-EFA file systems ensure that the auto mounter does not run until the network connectivity is online.  |
| es=networ<br>k.service  | (3) Note  For Amazon Linux 2023 and Ubuntu 22.04 and higher, use the x-systemd.requires=systemd-networkd-wait-online.service option instead of the x-systemd.requires=network.service option.   |
| x-systemd .automount,x- systemd.requir es=configure- efa-fsx-lustre- client.service,x- systemd.a fter=configure- efa-fsx-lustre- client.service | These options for EFA-enabled file systems ensure that the auto mounter does not run until after EFA client configuration completes.  |
| 0   | A value that indicates whether the file system should be backed up<br>by dump. For Amazon FSx, this value should be 0.  |

Automount using /etc/fstab 173

| Field | Description  |
|-------|--|
| 0     | A value that indicates the order in which fsck checks file systems at boot. For Amazon FSx file systems, this value should be 0 to indicate that fsck should not run at startup. |

# Mounting specific filesets

By using the Lustre fileset feature, you can mount only a subset of the file system namespace, which is called a *fileset*. To mount a fileset of the file system, on the client you specify the subdirectory path after the file system name. A fileset mount (also called a subdirectory mount) limits the file system namespace visibility on a specific client.

### Example - Mount a Lustre fileset

1. Assume you have an FSx for Lustre file system with the following directories:

```
team1/dataset1/
team2/dataset2/
```

- You mount only the team1/dataset1 fileset, making only this part of the file system visible locally on the client. Use the following command and replace the following items:
  - Replace file\_system\_dns\_name with the actual file system's DNS name.
  - Replace *mountname* with the file system's mount name. This mount name is returned in the CreateFileSystem API operation response. It's also returned in the response of the **describe-file-systems** AWS CLI command, and the DescribeFileSystems API operation.

```
mount -t lustre file_system_dns_name@tcp:/mountname/team1/dataset1 /fsx
```

When using the Lustre fileset feature, keep the following in mind:

- There are no constraints preventing a client from remounting the file system using a different fileset, or no fileset at all.
- When using a fileset, some Lustre administrative commands requiring access to the .lustre/ directory may not work, such as the lfs fid2path command.

Mounting specific filesets 174

• If you plan to mount several subdirectories from the same file system on the same host, be aware that this consumes more resources than a single mount point, and it could be more efficient to mount the file system root directory only once instead.

For more information on the Lustre fileset feature, see the *Lustre Operations Manual* on the <u>Lustre</u> documentation website.

## **Unmounting file systems**

Before deleting an FSx for Lustre file system, ensure it is unmounted from all Amazon EC2 instances that have mounted it, and before shutting down or terminating any Amazon EC2 instance, ensure any mounted FSx for Lustre file systems are unmounted from that instance.

FSx for Lustre servers grant temporary file and directory locks to clients during I/O operations, and clients must respond promptly when servers ask clients to release their locks to unblock I/O operations from other clients. If clients become non-responsive, they may be forcefully evicted after several minutes to allow other clients to proceed with their requested I/O operations. To avoid these waiting periods, you should always unmount the file system from client instances before shutting down or terminating them, and before deleting FSx for Lustre file systems.

You can unmount a file system on your Amazon EC2 instance by running the umount command on the instance itself. You can't unmount an Amazon FSx file system through the AWS CLI, the AWS Management Console, or through any of the AWS SDKs. To unmount an Amazon FSx file system connected to an Amazon EC2 instance running Linux, use the umount command as follows:

umount /mnt/fsx

We recommend that you do not specify any other umount options. Avoid setting any other umount options that are different from the defaults.

You can verify that your Amazon FSx file system has been unmounted by running the df command. This command displays the disk usage statistics for the file systems currently mounted on your Linux-based Amazon EC2 instance. If the Amazon FSx file system that you want to unmount isn't listed in the df command output, this means that the file system is unmounted.

Example - Identify the mount status of an Amazon FSx file system and unmount it

\$ df -T

Unmounting file systems 175

```
Filesystem Type 1K-blocks Used Available Use% Mounted on file-system-id.fsx.aws-region.amazonaws.com@tcp:/mountname /fsx 3547708416 61440 3547622400 1% /fsx /dev/sdal ext4 8123812 1138920 6884644 15% /
```

```
$ df -T
```

```
Filesystem Type 1K-blocks Used Available Use% Mounted on /dev/sda1 ext4 8123812 1138920 6884644 15% /
```

\$ umount /fsx

# **Working with Amazon EC2 Spot Instances**

FSx for Lustre can be used with EC2 Spot Instances to significantly lower your Amazon EC2 costs. A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Amazon EC2 can interrupt your Spot Instance when the Spot price exceeds your maximum price, when the demand for Spot Instances rises, or when the supply of Spot Instances decreases.

When Amazon EC2 interrupts a Spot Instance, it provides a Spot Instance interruption notice, which gives the instance a two-minute warning before Amazon EC2 interrupts it. For more information, see <a href="Spot Instances">Spot Instances</a> in the Amazon EC2 User Guide.

To ensure that Amazon FSx file systems are unaffected by EC2 Spot Instances Interruptions, we recommend unmounting Amazon FSx file systems prior to terminating or hibernating EC2 Spot Instances. For more information, see Unmounting file systems.

## **Handling Amazon EC2 Spot Instance interruptions**

FSx for Lustre is a distributed file system where server and client instances cooperate to provide a performant and reliable file system. They maintain a distributed and coherent state across both client and server instances. FSx for Lustre servers delegate temporary access permissions to clients while they are actively doing I/O and caching file system data. Clients are expected to reply in a short period of time when servers request them to revoke their temporary access permissions. To protect the file system against misbehaving clients, servers can evict Lustre clients that do not respond after a few minutes. To avoid having to wait multiple minutes for a non-responding client to reply to the server request, it is important to cleanly unmount Lustre clients, especially before terminating EC2 Spot Instances.

Using EC2 Spot Instances 176

EC2 Spot sends termination notices 2 minutes in advance before shutting down an instance. We recommend that you automate the process of cleanly unmounting Lustre clients before terminating EC2 Spot Instances.

### Example – Script to cleanly unmount terminating EC2 Spot Instances

This example script cleanly unmounts terminating EC2 Spot Instances by doing the following:

- Watches for Spot termination notices.
- When it receives a termination notice:
  - Stop applications that are accessing the file system.
  - Unmounts the file system before the instance is terminated.

You can adapt the script as needed, especially for gracefully shutting down your application. For more information about best practices for handling Spot Instance interruptions, see <a href="Message: Best practices">Best practices</a> for handling EC2 Spot Instance interruptions.

```
#!/bin/bash
# TODO: Specify below the FSx mount point you are using
*FSXPATH=/fsx*
cd /
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")
if [ "$?" -ne 0 ]; then
    echo "Error running 'curl' command" >&2
    exit 1
fi
# Periodically check for termination
while sleep 5
do
    HTTP_CODE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s -w %{http_code} -o /dev/
null http://169.254.169.254/latest/meta-data/spot/instance-action)
    if [[ "$HTTP_CODE" -eq 401 ]] ; then
        # Refreshing Authentication Token
```

```
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 30")
        continue
    elif [[ "$HTTP_CODE" -ne 200 ]] ; then
        # If the return code is not 200, the instance is not going to be interrupted
        continue
    fi
    echo "Instance is getting terminated. Clean and unmount '$FSXPATH' ..."
    curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-
data/spot/instance-action
    echo
    # Gracefully stop applications accessing the filesystem
    # TODO*: Replace with the proper command to stop your application if possible*
    # Kill every process still accessing Lustre filesystem
    echo "Kill every process still accessing Lustre filesystem..."
    fuser -kMm -TERM "${FSXPATH}"; sleep 2
    fuser -kMm -KILL "${FSXPATH}"; sleep 2
    # Unmount FSx For Lustre filesystem
    if ! umount -c "${FSXPATH}"; then
        echo "Error unmounting '$FSXPATH'. Processes accessing it:" >&2
        lsof "${FSXPATH}"
        echo "Retrying..."
        continue
    fi
    # Start a graceful shutdown of the host
    shutdown now
done
```

# Administering file systems

FSx for Lustre provides a set of features that simplify the performance of your administrative tasks. These include the ability to take point-in-time backups, to manage file system storage quotas, to manage your storage and throughput capacity, to manage data compression, and to set maintenance windows for performing routine software patching of the system.

You can administer your FSx for Lustre file systems using the Amazon FSx Management Console, AWS Command Line Interface (AWS CLI), Amazon FSx API, or AWS SDKs.

### **Topics**

- · Working with EFA-enabled file systems
- Using Lustre storage quotas
- · Managing storage capacity
- Managing provisioned SSD read cache
- Managing metadata performance
- Managing provisioned throughput capacity
- Lustre data compression
- Lustre root squash
- FSx for Lustre file system status
- Tag your Amazon FSx for Lustre resources
- Amazon FSx for Lustre maintenance windows
- Managing Lustre versions
- Deleting a file system

# Working with EFA-enabled file systems

If you are creating a file system with over 10 GBps of throughput capacity, we recommend enabling Elastic Fabric Adapter (EFA) to optimize throughput per client instance. EFA is a high-performance network interface that uses a custom-built operating system bypass technique and the AWS Scalable Reliable Datagram (SRD) network protocol to increase performance. For information about EFA, see Elastic Fabric Adapter for AI/ML and HPC workloads on Amazon EC2 in the Amazon EC2 User Guide.

EFA-enabled file systems 179

EFA-enabled file systems support two additional performance features: GPUDirect Storage (GDS) and ENA Express. GDS support builds on EFA to further enhance performance by enabling direct data transfer between the file system and the GPU memory, bypassing the CPU. This direct path eliminates the need for redundant memory copies and CPU involvement in the data transfer operations. With EFA and GDS support, you can achieve higher throughput to individual EFA-enabled client instances. ENA Express provides optimized network communication for Amazon EC2 instances using an advanced path selection algorithm and enhanced congestion control mechanism. With ENA Express support, you can achieve higher throughput to individual ENA Express-enabled client instances. For information about ENA Express, see <a href="Improve network">Improve network</a> performance between EC2 instances with ENA Express in the Amazon EC2 User Guide.

### **Topics**

- · Considerations when using EFA-enabled file systems
- Prerequisites for using EFA-enabled file systems
- Creating an EFA-enabled file system

## Considerations when using EFA-enabled file systems

Here are a few important items to consider when creating EFA-enabled file systems:

- Multiple connectivity options: EFA-enabled file systems can communicate with client instances using ENA, ENA Express, and EFA.
- **Deployment type:** EFA is supported on Persistent 2 file systems with a metadata configuration specified, including file systems using the Intelligent-Tiering storage class.
- **Updating EFA setting:** You can choose to enable EFA when you create a new file system but you cannot enable or disable EFA on an existing file system.
- Scaling throughput with storage capacity: You can scale storage capacity on an EFA-enabled SSD-based file system to increase throughput capacity but you cannot change the throughput tier of an EFA-enabled file system.
- AWS Regions: For a list of AWS Regions that support EFA-enabled Persistent 2 file systems, see Deployment type availability.

## Prerequisites for using EFA-enabled file systems

The following are prerequisites for using EFA-enabled file systems:

### To create your EFA-enabled file system:

- Use an EFA-enabled security group. For more information, see EFA-enabled security groups.
- Use the same Availability Zone and /16 CIDR as your EFA-enabled client instances within your Amazon VPC.
- On Intelligent-Tiering file systems, EFA is only supported with a throughput capacity of 4,000 MBps or increments of 4,000 MBps.

### To access your file system using Elastic Fabric Adapter (EFA):

- Use Nitro v4 (or higher) EC2 instances that support EFA, excluding the trn2 instance family. See Supported instance types in the *Amazon EC2 User Guide*.
- Run AL2023, RHEL 9.5 and newer, or Ubuntu 22+ with kernel version of 6.8 and newer. For more information, see Installing the Lustre client.
- Install the EFA modules and configure EFA interfaces on your client instances. For more information, see Configuring EFA clients.

### To access your file system using GPUDirect Storage (GDS):

- Use an Amazon EC2 P5, P5e, P5en, P6-B200, or P6e-GB200 client instance.
- Install the NVIDIA Compute Unified Device Architecture (CUDA) package, the open source NVIDIA driver, and the NVIDIA GPUDirect Storage Driver on your client instance. For more information, see Install the GDS driver (optional).

#### To access your file system using ENA Express:

- Use Amazon EC2 instances that support ENA Express. See <u>Supported instance types for ENA</u> Express in the Amazon EC2 User Guide.
- Update the settings for your Linux instance. See <u>Prerequisites for Linux instances</u> in the *Amazon EC2 User Guide*.
- Enable ENA Express on network interfaces for your client instances. For details, see <u>Review ENA</u>
   Express settings for your EC2 instance in the *Amazon EC2 User Guide*.

## Creating an EFA-enabled file system

This section contains instructions on how to create an FSx for Lustre EFA-enabled file system using the AWS CLI. For information on how to create an EFA-enabled file system using the Amazon FSx console, see Step 1: Create your FSx for Lustre file system.

### To create an EFA-enabled file system (CLI)

Use the <u>create-file-system</u> CLI command (or the equivalent <u>CreateFileSystem</u> API operation). The following example creates an FSx for Lustre EFA-enabled file system with a PERSISTENT\_2 deployment type.

```
aws fsx create-file-system\
    --storage-capacity 4800 \
    --storage-type SSD \
    --file-system-type LUSTRE \
    --file-system-type-version 2.15 \
    --subnet-ids subnet-01234567890 \
    --security-group-ids sg-0123456789abcdefg \
    --lustre-configuration '{"DeploymentType": "PERSISTENT_2", "EfaSupport": true}'
```

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

# **Using Lustre storage quotas**

You can create storage quotas for users, groups, and projects on FSx for Lustre file systems. With storage quotas, you can limit the amount of disk space and the number of files that a user, group, or project can consume. Storage quotas automatically track user-level, group-level, and project-level usage so you can monitor consumption whether or not you choose to set storage limits.

Amazon FSx enforces quotas and prevents users who have exceeded them from writing to the storage space. When users exceed their quotas, they must delete enough files to get under the quota limits so that they can write to the file system again.

#### **Topics**

- Quota enforcement
- Types of quotas

FSx for Lustre Lustre User Guide

- Quota limits and grace periods
- Setting and viewing quotas
- Quotas and Amazon S3 linked buckets
- Quotas and restoring backups

## **Quota enforcement**

User, group, and project quota enforcement is automatically enabled on all FSx for Lustre file systems. You cannot disable quota enforcement.

## Types of quotas

System administrators with AWS account root user credentials can create the following types of quotas:

- A user quota applies to an individual user. A user quota for a specific user can be different from the quotas of other users.
- A group quota applies to all users who are members of a specific group.
- A project quota applies to all files or directories associated with a project. A project can include multiple directories or individual files located in different directories within a file system.



### Note

Project quotas are only supported on Lustre version 2.15 on FSx for Lustre file systems.

- A block quota limits the amount of disk space that a user, group, or project can consume. You configure the storage size in kilobytes.
- An *inode quota* limits the number of files or directories that a user, group, or project can create. You configure the maximum number of inodes as an integer.



#### Note

Default quotas aren't supported.

Quota enforcement 183 FSx for Lustre Lustre User Guide

If you set quotas for a particular user and a group, and the user is a member of that group, the user's data usage applies to both quotas. It is also limited by both quotas. If either quota limit is reached, the user is blocked from writing to the file system.



#### Note

Quotas set for the root user are not enforced. Similarly, writing data as the root user using the sudo command bypasses enforcement of the quota.

## **Quota limits and grace periods**

Amazon FSx enforces user, group, and project quotas as a hard limit or as a soft limit with a configurable grace period.

The hard limit is the absolute limit. If users exceed their hard limit, a block or inode allocation fails with a Disk quota exceeded message. Users who have reached their quota hard limit must delete enough files or directories to get under the quota limit before they can write to the file system again. When a grace period is set, users can exceed the soft limit within the grace period if under the hard limit.

For soft limits, you configure a grace period in seconds. The soft limit must be smaller than the hard limit.

You can set different grace periods for inode and block guotas. You can also set different grace periods for a user quota, a group quota, and a project quota. When user, group, and project quotas have different grace periods, the soft limit transforms to a hard limit after the grace period of any of these quotas elapses.

When users exceed a soft limit, Amazon FSx allows them to continue exceeding their quota until the grace period has elapsed or until the hard limit is reached. After the grace period ends, the soft limit converts to a hard limit, and users are blocked from any further write operations until their storage usage returns below the defined block quota or inode quota limits. Users don't receive a notification or warning when the grace period begins.

## **Setting and viewing quotas**

You set storage quotas using Lustre file system 1fs commands in your Linux terminal. The 1fs setquota command sets quota limits, and the 1fs quota command displays quota information.

For more information about Lustre quota commands, see the *Lustre Operations Manual* on the Lustre documentation website.

### Setting user, group, and project quotas

The syntax of the setquota command for setting user, group, or project quotas is as follows.

#### Where:

- -u or --user specifies a user to set a quota for.
- -g or --group specifies a group to set a quota for.
- -p or --project specifies a project to set a quota for.
- -b sets a block quota with a soft limit. -B sets a block quota with a hard limit. Both
   block\_softlimit and block\_hardlimit are expressed in kilobytes, and the minimum value
   is 1024 KB.
- -i sets an inode quota with a soft limit. -I sets an inode quota with a hard limit. Both
   inode\_softlimit and inode\_hardlimit are expressed in number of inodes, and the
   minimum value is 1024 inodes.
- mount\_point is the directory that the file system was mounted on.

**User quota example:** The following command sets a 5,000 KB soft block limit, an 8,000 KB hard block limit, a 2,000 soft inode limit, and a 3,000 hard inode limit quota for user1 on the file system mounted to /mnt/fsx.

```
sudo lfs setquota -u user1 -b 5000 -B 8000 -i 2000 -I 3000 /mnt/fsx
```

**Group quota example:** The following command sets a 100,000 KB hard block limit for the group named group1 on the file system mounted to /mnt/fsx.

```
sudo lfs setquota -g group1 -B 100000 /mnt/fsx
```

Setting and viewing quotas 185

**Project quota example:** First make sure that you have used the project command to associate the desired files and directories with the project. For example, the following command associates all the files and sub-directories of the /mnt/fsxfs/dir1 directory with the project whose project ID is 100.

```
sudo lfs project -p 100 -r -s /mnt/fsxfs/dir1
```

Then use the setquota command to set the project quota. The following command sets a 307,200 KB soft block limit, a 309,200 KB hard block limit, a 10,000 soft inode limit, and an 11,000 hard inode limit quota for project 250 on the file system mounted to /mnt/fsx.

```
sudo lfs setquota -p 250 -b 307200 -B 309200 -i 10000 -I 11000 /mnt/fsx
```

### **Setting grace periods**

The default grace period is one week. You can adjust the default grace period for users, groups, or projects, using the following syntax.

```
lfs setquota -t {-u|-g|-p}
    [-b block_grace]
    [-i inode_grace]
    /mount_point
```

#### Where:

- -t indicates that a grace time period will be set.
- -u sets a grace period for all users.
- -g sets a grace period for all groups.
- -p sets a grace period for all projects.
- -b sets a grace period for block quotas. -i sets a grace period for inode quotas. Both
   block\_grace and inode\_grace are expressed in integer seconds or in the XXwXXdXXhXXmXXs
   format.
- mount\_point is the directory that the file system was mounted on.

The following command sets grace periods of 1,000 seconds for user block quotas and 1 week and 4 days for user inode quotas.

Setting and viewing quotas 186

sudo lfs setquota -t -u -b 1000 -i 1w4d /mnt/fsx

# **Viewing quotas**

The quota command displays information about user quotas, group quotas, project quotas, and grace periods.

| View quota command                    | Quota information displayed   |
|---------------------------------------|---|
| lfs quota /mount_point                | General quota information (disk usage and limits) for the user running the command and the user's primary group.  |
| lfs quota -u username /mount_point    | General quota informati on for a specific user. Users with AWS account root user credentials can run this command for any user, but non-root users can't run this command to get quota information about other users.   |
| lfs quota -u username -v /mount_point | General quota information for a specific user and detailed quota statistics for each object storage target (OST) and metadata target (MDT). Users with AWS account root user credentials can run this command for any user, but non-root users can't run this command to get quota information about other users. |

Setting and viewing quotas 187

| View quota command                  | Quota information displayed                       |
|-------------------------------------|---|
| lfs quota -g groupname /mount_point | General quota information for a specific group.   |
| lfs quota -p projectid /mount_point | General quota information for a specific project. |
| lfs quota -t -u /mount_point        | Block and inode grace times for user quotas.      |
| lfs quota -t -g /mount_point        | Block and inode grace times for group quotas.     |
| lfs quota -t -p /mount_point        | Block and inode grace times for project quotas.   |

## **Quotas and Amazon S3 linked buckets**

You can link your FSx for Lustre file system to an Amazon S3 data repository. For more information, see Linking your file system to an Amazon S3 bucket.

You can optionally choose a specific folder or prefix within a linked S3 bucket as an import path to your file system. When a folder in Amazon S3 is specified and imported into your file system from S3, only the data from that folder is applied towards the quota. The data of the entire bucket is not counted against the quota limits.

File metadata in a linked S3 bucket are imported into a folder with a structure matching the imported folder from Amazon S3. These files count towards the inode quotas of the users and groups who own the files.

When a user performs an hsm\_restore or lazy loads a file, the file's full size counts towards the block quota associated with the owner of the file. For example, if user A lazy loads a file that is owned by user B, the amount of storage and inode usage counts towards user B's quota. Similarly, when a user uses the Amazon FSx API to release a file, the data is freed up from the block quotas of the user or group who owns the file.

Because HSM restores and lazy loading are performed with root access, they bypass quota enforcement. Once data has been imported, it counts towards the user or group based on the ownership set in S3, which can cause users or groups to exceed their block limits. If this occurs, they'll need to free up files to be able to write to the file system again.

Similarly, file systems with automatic import enabled will automatically create new inodes for objects added to S3. These new inodes are created with root access and bypass quota enforcement while they're being created. These new inodes will count towards the users and groups, based on who owns the object in S3. If those users and groups exceed their inode quotas based on automatic import activity, they'll have to delete files in order to free up additional capacity and get below their quota limits.

## **Quotas and restoring backups**

When you restore a backup, the quota settings of the original file system are implemented in the restored file system. For example, if quotas are set in file system A, and file system B is created from a backup of file system A, file system A's quotas are enforced in file system B.

# **Managing storage capacity**

You can increase the SSD or HDD storage capacity that is configured on your FSx for Lustre file system as you need additional storage and throughput. Because the throughput of an FSx for Lustre file system scales linearly with storage capacity, you also get a comparable increase in throughput capacity. To increase the storage capacity, you can use the Amazon FSx console, the AWS Command Line Interface (AWS CLI), or the Amazon FSx API.

When you request an update to your file system's storage capacity, Amazon FSx automatically adds new network file servers and scales your metadata server. While scaling storage capacity, the file system may be unavailable for a few minutes. File operations issued by clients while the file system is unavailable will transparently retry and eventually succeed after storage scaling is complete. During the time that the file system is unavailable, the file system status is set to UPDATING. Once storage scaling is complete, the file system status is set to AVAILABLE.

Amazon FSx then runs a storage optimization process that transparently rebalances data across the existing and newly added file servers. Rebalancing is performed in the background with no impact to file system availability. During rebalancing, you might see decreased file system performance as resources are consumed for data movement. For most file systems, storage optimization takes a few hours up to a few days. You can access and use your file system during the optimization phase.

FSx for Lustre Lustre User Guide

You can track the storage optimization progress at any time using the Amazon FSx console, CLI, and API. For more information, see Monitoring storage capacity increases.

### **Topics**

- Considerations when increasing storage capacity
- When to increase storage capacity
- How concurrent storage scaling and backup requests are handled
- Increasing storage capacity
- Monitoring storage capacity increases

## Considerations when increasing storage capacity

Here are a few important items to consider when increasing storage capacity:

- Increase only You can only increase the amount of storage capacity for a file system; you cannot decrease storage capacity.
- Increase increments When you increase storage capacity, use the increments listed in the **Increase storage capacity** dialog box.
- Time between increases You can't make further storage capacity increases on a file system until 6 hours after the last increase was requested.
- Throughput capacity You automatically increase throughput capacity when you increase the storage capacity. For persistent HDD file systems with SSD cache, the read cache storage capacity is also similarly increased to maintain an SSD cache that is sized to 20 percent of the HDD storage capacity. Amazon FSx calculates the new values for the storage and throughput capacity units and lists them in the **Increase storage capacity** dialog box.



### Note

You can independently modify the throughput capacity of a persistent SSD-based file system without having to update the file system's storage capacity. For more information, see Managing provisioned throughput capacity.

• **Deployment type** – You can increase the storage capacity of all deployment types except for scratch 1 file systems.

# When to increase storage capacity

Increase your file system's storage capacity when it's running low on free storage capacity. Use the FreeStorageCapacity CloudWatch metric to monitor the amount of free storage that is available on the file system. You can create an Amazon CloudWatch alarm on this metric and get notified when it drops below a specific threshold. For more information, see Monitoring with Amazon CloudWatch.

You can use CloudWatch metrics to monitor your file system's ongoing throughput usage levels. If you determine that your file system needs a higher throughput capacity, you can use the metric information to help you decide how much to increase the storage capacity. For information about how to determine your file system's current throughput, see <a href="How to use Amazon FSx">How to use Amazon FSx for Lustre</a> <a href="CloudWatch metrics">CloudWatch metrics</a>. For information about how storage capacity affects throughput capacity, see <a href="Amazon FSx">Amazon FSx</a> for Lustre performance.

You can also view your file system's storage capacity and total throughput on the **Summary** panel of the file system details page.

# How concurrent storage scaling and backup requests are handled

You can request a backup just before a storage scaling workflow begins or while it is in progress. The sequence of how Amazon FSx handles the two requests is as follows:

- If a storage scaling workflow is in progress (storage scaling status is IN\_PROGRESS and file
  system status is UPDATING) and you request a backup, the backup request is queued. The backup
  task is started when storage scaling is in the storage optimization phase (storage scaling status is
  UPDATED\_OPTIMIZING and file system status is AVAILABLE).
- If the backup is in progress (backup status is CREATING) and you request storage scaling, the storage scaling request is queued. The storage scaling workflow is started when Amazon FSx is transferring the backup to Amazon S3 (backup status is TRANSFERRING).

If a storage scaling request is pending and a file system backup request is also pending, the backup task has higher precedence. The storage scaling task does not start until the backup task is finished.

FSx for Lustre Lustre User Guide

# **Increasing storage capacity**

You can increase a file system's storage capacity using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

### To increase storage capacity for a file system (console)

- Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/. 1.
- Navigate to **File systems**, and choose the Lustre file system that you want to increase storage 2. capacity for.
- For Actions, choose Update storage capacity. Or, in the Summary panel, choose Update next to the file system's **Storage capacity** to display the **Increase storage capacity** dialog box.
- For **Desired storage capacity**, provide a new storage capacity in GiB that is greater than the current storage capacity of the file system:
  - For a persistent SSD or scratch 2 file system, this value must be in multiples of 2400 GiB.
  - For a persistent HDD file system, this value must be in multiples of 6000 GiB for 12 MBps/ TiB file systems and multiples of 1800 GiB for 40 MBps/TiB file systems.
  - For an EFA-enabled file system, this value must be in multiples of 38400 GiB for 125 MBps/ TiB file systems, multiples of 19200 GiB for 250 MBps/TiB file systems, multiples of 9600 GiB for 500 MBps/TiB file systems, and multiples of 4800 GiB for 1000 MBps/TiB file systems.



#### Note

You cannot increase the storage capacity of scratch 1 file systems.

- 5. Choose **Update** to initiate the storage capacity update.
- 6. You can monitor the update progress on the file systems detail page in the **Updates** tab.

### To increase storage capacity for a file system (CLI)

To increase the storage capacity for an FSx for Lustre file system, use the AWS CLI command 1. update-file-system. Set the following parameters:

Set --file-system-id to the ID of the file system you are updating.

192 Increasing storage capacity

Set --storage-capacity to an integer value that is the amount, in GiB, of the storage capacity increase. For a persistent SSD or scratch 2 file system, this value must be in multiples of 2400. For a persistent HDD file system, this value must be in multiples of 6000 for 12 MBps/TiB file systems and multiples of 1800 for 40 MBps/TiB file systems. The new target value must be greater than the current storage capacity of the file system.

This command specifies a storage capacity target value of 9600 GiB for a persistent SSD or scratch 2 file system.

```
$ aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --storage-capacity 9600
```

2. You can monitor the progress of the update by using the AWS CLI command <u>describe-file-</u>systems. Look for the administrative-actions in the output.

For more information, see AdministrativeAction.

## Monitoring storage capacity increases

You can monitor the progress of a storage capacity increase using the Amazon FSx console, the API, or the AWS CLI.

### Monitoring increases in the console

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

You can view the following information:

### **Update type**

Supported types are **Storage capacity** and **Storage optimization**.

### **Target value**

The desired value to update the file system's storage capacity to.

#### **Status**

The current status of the storage capacity updates. The possible values are as follows:

• **Pending** – Amazon FSx has received the update request, but has not started processing it.

- In progress Amazon FSx is processing the update request.
- **Updated; Optimizing** Amazon FSx has increased the file system's storage capacity. The storage optimization process is now rebalancing data across the file servers.
- **Completed** The storage capacity increase completed successfully.
- **Failed** The storage capacity increase failed. Choose the question mark (?) to see details on why the storage update failed.

### **Progress %**

Displays the progress of the storage optimization process as percent complete.

### Request time

The time that Amazon FSx received the update action request.

### Monitoring increases with the AWS CLI and API

You can view and monitor file system storage capacity increase requests using the <a href="Monitoring-describe-file-systems">describe-file-systems</a> AWS CLI command and the <a href="Monitoring-DescribeFileSystems">DescribeFileSystems</a> API action. The AdministrativeActions array lists the 10 most recent update actions for each administrative action type. When you increase a file system's storage capacity, two AdministrativeActions are generated: a FILE\_SYSTEM\_UPDATE and a STORAGE\_OPTIMIZATION action.

The following example shows an excerpt of the response of a **describe-file-systems** CLI command. The file system has a storage capacity of 4800 GB, and there is a pending administrative action to increase the storage capacity to 9600 GB.

Amazon FSx processes the FILE\_SYSTEM\_UPDATE action first, adding new file servers to the file system. When the new storage is available to the file system, the FILE\_SYSTEM\_UPDATE status changes to UPDATED\_OPTIMIZING. The storage capacity shows the new larger value, and Amazon FSx begins processing the STORAGE\_OPTIMIZATION administrative action. This is shown in the following excerpt of the response of a **describe-file-systems** CLI command.

The ProgressPercent property displays the progress of the storage optimization process. After the storage optimization process completes successfully, the status of the FILE\_SYSTEM\_UPDATE action changes to COMPLETED, and the STORAGE\_OPTIMIZATION action no longer appears.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            "StorageCapacity": 9600,
            "AdministrativeActions": [
                    "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "RequestTime": 1581694764.757,
                    "Status": "UPDATED_OPTIMIZING",
                    "TargetFileSystemValues": {
                         "StorageCapacity": 9600
                }
                },
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
                    "Status": "IN_PROGRESS",
```

```
"ProgressPercent": 50,
}
```

If the storage capacity increase fails, the status of the FILE\_SYSTEM\_UPDATE action changes to FAILED. The FailureDetails property provides information about the failure, shown in the following example.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            "StorageCapacity": 4800,
            "AdministrativeActions": [
                {
                     "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                    "FailureDetails": {
                         "Message": "string"
                    },
                    "RequestTime": 1581694764.757,
                     "Status": "FAILED",
                     "TargetFileSystemValues":
                         "StorageCapacity": 9600
                }
            ]
```

# Managing 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-based read cache which provides SSD latencies for reads of your frequently-accessed data, up to 3 IOPS per GiB.

You can configure your SSD read cache for frequently-accessed data with one of these sizing mode options:

• Automatic (proportional to throughput capacity). With Automatic, Amazon FSx for Lustre automatically selects an SSD data read cache size based on provisioned throughput capacity.

SSD read caches 196

• **Custom (user-provisioned)**. 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.

• Choose No Cache if you do not want to use an SSD data read cache with your file system.

In Automatic (proportional to throughput capacity) mode, Amazon FSx automatically provisions the following default read cache size based on the throughput capacity of your file system.

SSD read caches 197

| Provisioned throughput<br>capacity (MBps) | SSD read cache in<br>Automatic (proportional<br>to throughput capacity)<br>mode (GiB) | Supported SSD read cache size | ize           |
|---|---|-------------------------------|---------------|
|   |   | minimum (GiB)                 | maximum (GiB) |
| Every 4000                                | 20000   | 32                            | 131072        |

SSD read caches 198

After your file system is created, you can modify your read cache's sizing mode and storage capacity at any time.

### **Topics**

- Considerations when updating SSD read cache
- Updating a provisioned SSD read cache
- Monitoring SSD read cache updates

## Considerations when updating SSD read cache

Here are a few important considerations when modifying your SSD data 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 increase or decrease the capacity size of an SSD read cache. However, you can only do
  this once every six hours. There is no time restriction when adding or removing an 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 a provisioned SSD read cache**

You can update your SSD data 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://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for Lustre file system that you want to update the SSD read cache for.
- 3. SSD On the **Summary** panel, choose **Update** next to the file system's **SSD read cache** value.
  - The **Update SSD read cache** dialog box appears.
- 4. Select the new sizing mode that you would like for your data read cache, as follows:
  - Choose **Automatic (proportional to throughput capacity)** to have your data 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 data 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 data read cache with your Intelligent-Tiering file system.

### 5. Choose **Update**.

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

To update the SSD data 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 --lustre-configuration
   DataReadCacheConfiguration 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 data read cache automatically sized based on your throughput capacity.

The following example updates the SSD read cache to USER\_PROVISIONED mode and sets the size to 524288 GiB.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration
'DataReadCacheConfiguration={SizeGiB=524288,SizingMode=USER_PROVISIONED}'
```

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

For more information, see AdministrativeAction in the Amazon FSx 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 <u>describe-file-systems</u> AWS CLI command and the <u>DescribeFileSystems</u> 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.

FSx for Lustre Lustre User Guide

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": {
            "LustreConfiguration": {
                "DataReadCacheConfiguration": {
                      "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.

## Managing metadata performance

You can update the metadata configuration of your FSx for Lustre 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). The update procedure increases the number of provisioned Metadata IOPS for your file system.



### Note

Enhanced metadata is available only for 2.15 file systems. You can increase metadata performance only on FSx for Lustre file systems created with the Persistent 2 deployment type and a metadata configuration specified. You cannot add or update the metadata configuration for an FSx for Lustre file system if the metadata configuration is not specified at the time of file system creation. This also applies to file systems restored from backups

of 2.12 file systems which did not support enhanced metadata performance, or from 2.15 file systems which had no metadata configuration specified.

The increased metadata performance of your file system is available for use within minutes. You can update the metadata performance at anytime, as long as metadata performance increase requests are at least 6 hours apart. While scaling metadata performance, the file system may be unavailable for a few minutes. File operations issued by clients while the file system is unavailable will transparently retry and eventually succeed after metadata performance scaling is complete. You will be billed for the new metadata performance increase after it becomes available to you.

You can track the progress of a metadata performance increase at any time by using the Amazon FSx console, CLI, and API. For more information, see Monitoring metadata configuration updates.

### **Topics**

- · Lustre metadata performance configuration
- Considerations when increasing metadata performance
- When to increase metadata performance
- · Increasing metadata performance
- Changing the metadata configuration mode
- Monitoring metadata configuration updates

## Lustre metadata performance configuration

The number of provisioned Metadata IOPS determines the maximum rate of metadata operations that can be supported by the file system.

When you create the file system, you choose a metadata configuration mode:

- For SSD file systems, you can choose Automatic mode if you want Amazon FSx to automatically
  provision and scale the Metadata IOPS on your file system based on your file system's storage
  capacity. Note that Intelligent-Tiering file systems don't support Automatic mode.
- For SSD file systems, you can choose User-provisioned if you want to specify the number of Metadata IOPS to provision for your file system.

For Intelligent-Tiering file systems, you must choose User-provisioned mode. With User-provisioned mode, you can specify the number of Metadata IOPS to provision for your file system.

On SSD file systems, you can switch from Automatic mode to User-provisioned mode at any time. You can also switch from User-provisioned to Automatic mode if the number of Metadata IOPS provisioned on your file system matches the default number of Metadata IOPS provisioned in Automatic mode. Intelligent-Tiering file systems only support User-provisioned mode, so you can't switch metadata configuration modes.

Valid Metadata IOPS values are as follows:

- For SSD file systems, valid Metadata IOPS values are 1500, 3000, 6000, and multiples of 12000 up to a maximum of 192000.
- For Intelligent-Tiering file systems, valid Metadata IOPS values are 6000 and 12000.

If the metadata performance of your workload exceeds the number of Metadata IOPS provisioned in Automatic mode, you can use User-provisioned mode to increase the Metadata IOPS value for your file system.

You can view the current value of the file system's metadata server configuration as follows:

- Using the console On the Summary panel of the file system details page, the Metadata IOPS
  field shows the current value of the provisioned Metadata IOPS and the current metadata
  configuration mode of the file system.
- Using the CLI or API Use the <u>describe-file-systems</u> CLI command or the <u>DescribeFileSystems</u> API operation, and look for the MetadataConfiguration property.

### Considerations when increasing metadata performance

Here are a few important considerations when increasing your metadata performance:

- **Metadata performance increase only** You can only *increase* the number of Metadata IOPS for a file system; you cannot decrease the number of Metadata IOPS.
- Specifying Metadata IOPS in Automatic mode not supported You can't specify the number of Metadata IOPS on a file system that is in Automatic mode. You'll have to switch to User-

provisioned mode and then make the request. For more information, see <u>Changing the metadata</u> configuration mode.

- Metadata IOPS for data written before scaling When scaling Metadata IOPS beyond 12000,
  FSx for Lustre adds new metadata servers to your file system. New metadata is automatically
  distributed across all servers for improved performance. However, existing metadata and
  subdirectories created before scaling remain on original servers, with no increase in Metadata
  IOPS.
- **Time between increases** You can't make further metadata performance increases on a file system until 6 hours after the last increase was requested.
- **Concurrent metadata performance and SSD storage increases** You cannot scale metadata performance and file system storage capacity concurrently.

## When to increase metadata performance

Increase the number of Metadata IOPS when you need to run workloads that require higher levels of metadata performance than is provisioned by default on your file system. You can monitor your metadata performance on the AWS Management Console by using the Metadata IOPS Utilization graph which provides the percentage of provisioned metadata server performance you are consuming on your file system.

You can also monitor your metadata performance using more granular CloudWatch metrics. CloudWatch metrics include DiskReadOperations and DiskWriteOperations, which provide the volume of metadata server operations that require disk IO, as well as granular metrics for metadata operations including file and directory creation, stats, reads, and deletes. For more information, see FSx for Lustre metadata metrics.

## Increasing metadata performance

You can increase a file system's metadata performance by using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

### To increase metadata performance for a file system (console)

- 1. Open the Amazon FSx console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.
- 2. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for Lustre file system that you want to increase metadata performance for.

3. For **Actions**, choose **Update Metadata IOPS**. Or, in the **Summary** panel, choose **Update** next to the file system's **Metadata IOPS** field.

The **Update Metadata IOPS** dialog box appears.

- 4. Choose **User-provisioned**.
- 5. For **Desired Metadata IOPS**, choose the new Metadata IOPS value. The value you enter must be greater than or equal to the current Metadata IOPS value.
  - For SSD file systems, valid values are 1500, 3000, 6000, 12000, and multiples of 12000 up to a maximum of 192000.
  - For Intelligent-Tiering file systems, valid values are 6000 and 12000.
- Choose Update.

### To increase metadata performance for a file system (CLI)

To increase the metadata performance for an FSx for Lustre file system, use the AWS CLI command <u>update-file-system</u> (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 metadata performance, use the --lustre-configuration MetadataConfiguration property. This property has two parameters, Mode and Iops.
  - 1. If your file system is in USER\_PROVISIONED mode, using Mode is optional (if used, set Mode to USER\_PROVISIONED).
    - If your SSD file system is in AUTOMATIC mode, set Mode to USER\_PROVISIONED (which switches the file system mode to USER\_PROVISIONED in addition to increasing the Metadata IOPS value).
  - 2. For SSD file systems, set Iops to a value of 1500, 3000, 6000, 12000, or multiples of 12000 up to a maximum of 192000. For Intelligent-Tiering file systems, set Iops to 6000 or 12000. The value you enter must be greater than or equal to the current Metadata IOPS value.

The following example updates the provisioned Metadata IOPS to 12000.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration 'MetadataConfiguration={Mode=USER_PROVISIONED, Iops=12000}'
```

# Changing the metadata configuration mode

For SSD-based file systems, you can change the metadata configuration mode of an existing file system using the AWS console and CLI, as explained in the following procedures.

When switching from Automatic mode to User-provisioned mode, you must provide a Metadata IOPS value greater than or equal to the current file system Metadata IOPS value.

If you request to switch from User-provisioned to Automatic mode and the current Metadata IOPS value is greater than the automated default, Amazon FSx rejects the request, because downscaling Metadata IOPS is not supported. To unblock mode switch, you must increase storage capacity to match your current Metadata IOPS in Automatic mode in order to enable mode switch again.

You can change a file system's metadata configuration mode by using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

#### To change the metadata configuration mode for a file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for Lustre file system that you want to change the metadata configuration mode for.
- For Actions, choose Update Metadata IOPS. Or, in the Summary panel, choose Update next to the file system's Metadata IOPS field.

The **Update Metadata IOPS** dialog box appears.

- 4. Do one of the following.
  - To switch from User-provisioned mode to Automatic mode, choose Automatic.
  - To switch from Automatic mode to User-provisioned mode, choose **User-provisioned**. Then, for **Desired Metadata IOPS**, provide a Metadata IOPS value greater than or equal to the current file system Metadata IOPS value.
- 5. Choose **Update**.

### To change the metadata configuration mode for an SSD file system (CLI)

To change the metadata configuration mode for an SSD FSx for Lustre file system, use the AWS CLI command <u>update-file-system</u> (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 change the metadata configuration mode on SSD-based file systems, use the --lustreconfiguration MetadataConfiguration property. This property has two parameters, Mode and Iops.

 To switch your SSD file system from AUTOMATIC mode to USER\_PROVISIONED mode, set Mode to USER\_PROVISIONED and Iops to a Metadata IOPS value greater than or equal to the current file system Metadata IOPS value. For example:

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration
'MetadataConfiguration={Mode=USER_PROVISIONED, Iops=96000}'
```

 To switch from USER\_PROVISIONED mode to AUTOMATIC mode, set Mode to AUTOMATIC and do not use the Iops parameter. For example:

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration 'MetadataConfiguration={Mode=AUTOMATIC}'
```

## Monitoring metadata configuration updates

You can monitor the progress of metadata configuration updates by using the Amazon FSx console, the API, or the AWS CLI.

### Monitoring metadata configuration updates (console)

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

For metadata configuration updates, you can view the following information:

#### **Update type**

Supported types are **Metadata IOPS** and **Metadata configuration mode**.

#### Target value

The updated value for the file system's Metadata IOPS or Metadata configuration mode.

#### **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 metadata configuration updates (CLI)

You can view and monitor metadata configuration update requests using the <u>describe-file-systems</u> AWS CLI command and the <u>DescribeFileSystems</u> API operation. The AdministrativeActions array lists the 10 most recent update actions for each administrative action type. When you update a file system's metadata performance or metadata configuration mode, 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 Metadata IOPS to 96000 and the metadata configuration mode to USER\_PROVISIONED.

]

Amazon FSx processes the FILE\_SYSTEM\_UPDATE action, modifying the file system's Metadata IOPS and metadata configuration mode. When the new metadata resources are available to the file system the FILE\_SYSTEM\_UPDATE status changes to COMPLETED.

If the metadata configuration 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": 1678840205.853,
        "Status": "FAILED",
        "TargetFileSystemValues": {
            "LustreConfiguration": {
                "MetadataConfiguration": {
                     "Iops": 96000,
                     "Mode": USER_PROVISIONED
                }
            }
        },
        "FailureDetails": {
            "Message": "failure-message"
        }
    }
]
```

# Managing provisioned throughput capacity

Every FSx for Lustre file system has a throughput capacity that is configured when you create the file system. For file systems using SSD or HDD storage, the throughput capacity is measured in megabytes per second per tebibyte (MBps/TiB). For file systems using Intelligent-Tiering storage, the throughput capacity is measured in megabytes per second (MBps) for the file system. 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 Amazon FSx for Lustre performance.

Throughput capacity 210

You can modify the throughput tier of a persistent SSD-based file system by increasing or decreasing the value of the file system's throughput per unit of storage. Valid values depend on the deployment type of the file system, as follows:

- For Persistent 1 SSD-based deployment types, valid values are 50, 100, and 200 MBps/TiB.
- For Persistent 2 SSD-based deployment types, valid values are 125, 250, 500, and 1000 MBps/ TiB.

You can modify the throughput capacity of an Intelligent-Tiering file system by increasing the value of the total throughput capacity for the file system. Valid values are 4,000 MBps or increments of 4,000 MBps, up to a maximum of 2,000,000 MBps.

You can view the current value of the file system's throughput capacity as follows:

- Using the console On the Summary panel of the file system details page, the Throughput per unit of storage field shows the current value for SSD-based file systems while the Throughput capacity field shows the current value for Intelligent-Tiering file systems.
- Using the CLI or API Use the <u>describe-file-systems</u> CLI command or the <u>DescribeFileSystems</u> API operation, and look for the PerUnitStorageThroughput property.

When you modify your file system's throughput capacity, behind the scenes, Amazon FSx switches out the file system's file servers on SSD file systems or adds new file servers on Intelligent-Tiering file systems. Your file system will be unavailable for a few minutes during throughput capacity scaling. You are billed for the new amount of throughput capacity once it is available to your file system.

#### **Topics**

- Considerations when updating throughput capacity
- When to modify throughput capacity
- Modifying throughput capacity
- Monitoring throughput capacity changes

# Considerations when updating throughput capacity

Here are a few important items to consider when updating throughput capacity:

 Increase or decrease – You can increase or decrease the amount of throughput capacity for an SSD-based file system. You can only increase the amount of throughput capacity for an Intelligent-Tiering file system.

- Update increments When you modify throughput capacity, use the increments listed in the
  Update throughput tier dialog box for SSD-based file systems or in the Update throughput
  capacity dialog box for Intelligent-Tiering file systems.
- **Time between increases** You can't make further throughput capacity changes on a file system until 6 hours after the last request, or until the throughput optimization process has completed, whichever time is longer.
- Automatic scaling of SSD read cache For the SSD read cache default mode (Proportional
  to throughput capacity), Amazon FSx automatically provisions 5 GiB of data storage for every
  MBps of throughput capacity you provision. As you scale your file system's throughput capacity,
  Amazon FSx automatically scales your SSD data cache by attaching additional cache storage to
  any newly added file servers.
- **Deployment type** You can only update the throughput capacity of persistent SSD-based or Intelligent-Tiering deployment types. You cannot modify the throughput capacity of EFA-enabled SSD-based file systems.

# 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 class. For information about how to determine your file system's current throughput, see <a href="How to use Amazon FSx">How to use Amazon FSx for Lustre CloudWatch metrics</a>. For information about CloudWatch metrics, see <a href="Monitoring with Amazon CloudWatch">Monitoring with Amazon CloudWatch</a>.

## Modifying throughput capacity

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

### To modify an SSD file system's throughput capacity (console)

Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.

FSx for Lustre Lustre User Guide

Navigate to File systems, and choose the FSx for Lustre file system that you want to modify 2. the throughput capacity for.

For Actions, choose Update throughput tier. Or, in the Summary panel, choose Update next 3. to the file system's **Throughput per unit of storage**.

The **Update throughput tier** window appears.

- Choose the new value for **Desired throughput per unit of storage** from the list. 4.
- 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 an SSD 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). Set the following parameters:
  - Set --file-system-id to the ID of the file system that you are updating.
  - Set --lustre-configuration PerUnitStorageThroughput to a value of 50, 100, or 200 MBps/TiB for Persistent 1 SSD file systems, or to a value of 125, 250, 500, or 1000 MBps/TiB for Persistent 2 SSD file systems.

This command specifies that throughput capacity be set to 1000 MBps/TiB for the file system.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration PerUnitStorageThroughput=1000
```

## To modify an Intelligent-Tiering file system's throughput capacity (console)

- Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/. 1.
- 2. Navigate to **File systems**, and choose the FSx for Lustre file system that you want to modify the throughput capacity for.

FSx for Lustre Lustre User Guide

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** dialog box appears.

Choose the new value for **Desired throughput capacity** from the list.

Amazon FSx will automatically scale your data read cache to avoid clearing the cache contents.

Choose **Update** to initiate the throughput capacity update. 5.



### Note

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

#### To modify an Intelligent-Tiering 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). Set the following parameters:
  - Set --file-system-id to the ID of the file system that you are updating.
  - If your data read cache is configured in proportional to throughput capacity mode, set -lustre-configuration ThroughputCapacity to a throughput level of increments of 4000 MBps, up to a maximum of 2000000 MBps.

If your data read cache is configured in user-provisioned mode, you also need to use the -lustre-configuration DataReadCacheConfiguration property to specify the data read cache. You need to maintain the same cache storage per server ratio and specify the new SizeGiB, or the request will be rejected.

This command specifies that throughput capacity be set to 8000 MBps for a file system that uses a read cache configured in proportional to throughput capacity mode.

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration '{
      "ThroughputCapacity": 8000
      }'
```

This command specifies that throughput capacity be set to 8000 MBps for a file system that uses a read cache configured in user-provisioned mode.

# Monitoring throughput capacity changes

You can monitor the progress of a throughput capacity modification using the Amazon FSx console, the API, and the AWS CLI.

### Monitoring throughput capacity changes (console)

• On the **Updates** tab in the file system details page, you can view the 10 most recent update actions for each update action type.

For throughput capacity update actions, you can view the following information.

### **Update type**

Supported type is **Per unit storage throughput**.

### Target value

The desired value to change the file system's throughput per unit of storage to.

#### **Status**

The current status of the update. For throughput capacity updates, the possible values are as follows:

• **Pending** – Amazon FSx has received the update request, but has not started processing it.

- In progress Amazon FSx is processing the update request.
- **Updated; Optimizing** Amazon FSx has updated the file system's network I/O, CPU, and memory resources. The new disk I/O performance level is available for write operations. Your read operations will see disk I/O performance between the previous level and the new level until your file system is no longer in the this state.
- **Completed** The throughput capacity update completed successfully.
- **Failed** The throughput capacity update failed. Choose the question mark (?) to see details on why the throughput update failed.

#### Request time

The time when Amazon FSx received the update request.

### Monitoring file system updates (CLI)

You can view and monitor file system throughput capacity modification requests using the <u>describe-file-systems</u> CLI command and the <u>DescribeFileSystems</u> API action. The AdministrativeActions array lists the 10 most recent update actions for each administrative action type. When you modify a file system's throughput capacity, a FILE\_SYSTEM\_UPDATE administrative action is generated.

The following example shows the response excerpt of a describe-file-systems CLI command. The file system has a target throughput per unit of storage of 500 MBps/TiB.

}

When Amazon FSx processes the action successfully, the status changes to COMPLETED. The new throughput capacity is then available to the file system, and shows in the PerUnitStorageThroughput property.

If the throughput capacity modification fails, the status changes to FAILED, and the FailureDetails property provides information about the failure.

# **Lustre data compression**

You can use the Lustre data compression feature to achieve cost savings on your high-performance Amazon FSx for Lustre file systems and backup storage. When data compression is enabled, Amazon FSx for Lustre automatically compresses newly-written files before they are written to disk and automatically uncompresses them when they are read.

Data compression uses the LZ4 algorithm, which is optimized to deliver high levels of compression without adversely impacting file system performance. LZ4 is a Lustre community-trusted and performance-oriented algorithm that provides a balance between compression speed and compressed file size. Enabling data compression does not typically have a measurable impact on latency.

Data compression reduces the amount of data that is transferred between Amazon FSx for Lustre file servers and storage. If you are not already using compressed file formats, you will see an increase in overall file system throughput capacity when using data compression. Increases in throughput capacity that are related to data compression will be capped after you have saturated your front-end network interface cards.

For example, if your file system is a PERSISTENT-50 SSD deployment type, your network throughput has a baseline of 250 MBps per TiB of storage. Your disk throughput has a baseline of 50 MBps per TiB. With data compression, your disk throughput could increase from 50 MBps per TiB to a maximum of 250 MBps per TiB, which is the baseline network throughput limit. For more information about network and disk throughput limits, see the file system performance tables in Performance characteristics of SSD and HDD storage classes. For more information about data compression performance, see the Spend less while increasing performance with Amazon FSx for Lustre data compression post on the AWS Storage Blog.

Data compression 217

#### **Topics**

- Managing data compression
- Compressing previously written files
- Viewing file sizes
- Using CloudWatch metrics

## Managing data compression

You can turn data compression on or off when creating a new Amazon FSx for Lustre file system. Data compression is turned off by default when you create an Amazon FSx for Lustre file system from the console, AWS CLI, or API.

### To turn on data compression when creating a file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Follow the procedure for creating a new file system described in <u>Step 1: Create your FSx for</u> Lustre file system in the *Getting started* section.
- 3. In the File system details section, for Data compression type, choose LZ4.
- 4. Complete the wizard as you do when you create a new file system.
- 5. Choose Review and create.
- Review the settings you chose for your Amazon FSx for Lustre file system, and then choose Create file system.

When the file system is **Available**, data compression is turned on.

### To turn on data compression when creating a file system (CLI)

• To create an FSx for Lustre file system with data compression turned on, use the Amazon FSx CLI command <a href="mailto:create-file-system">create-file-system</a> with the DataCompressionType parameter, as shown following. The corresponding API operation is <a href="mailto:CreateFileSystem">CreateFileSystem</a>.

```
$ aws fsx create-file-system \
     --client-request-token CRT1234 \
     --file-system-type LUSTRE \
     --file-system-type-version 2.12 \
     --lustre-configuration
DeploymentType=PERSISTENT_1,PerUnitStorageThroughput=50,DataCompressionType=LZ4 \
```

Managing data compression 218

```
--storage-capacity 3600 \
--subnet-ids subnet-123456 \
--tags Key=Name, Value=Lustre-TEST-1 \
--region us-east-2
```

After successfully creating the file system, Amazon FSx returns the file system description as JSON, as shown in the following example.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            "CreationTime": 1549310341.483,
            "FileSystemId": "fs-0123456789abcdef0",
            "FileSystemType": "LUSTRE",
            "FileSystemTypeVersion": "2.12",
            "Lifecycle": "CREATING",
            "StorageCapacity": 3600,
            "VpcId": "vpc-123456",
            "SubnetIds": [
                "subnet-123456"
            ],
            "NetworkInterfaceIds": [
                "eni-039fcf55123456789"
            ],
            "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
            "ResourceARN": "arn:aws:fsx:us-east-2:123456:file-system/
fs-0123456789abcdef0",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Lustre-TEST-1"
                }
            ],
            "LustreConfiguration": {
                "DeploymentType": "PERSISTENT_1",
                "DataCompressionType": "LZ4",
                "PerUnitStorageThroughput": 50
            }
        }
```

Managing data compression 219

}

You can also change the data compression configuration of your existing file systems. When you turn data compression on for an existing file system, only newly written files are compressed, and existing files are not compressed. For more information, see Compressing previously written files.

### To update data compression on an existing file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Navigate to **File systems**, and choose the Lustre file system that you want to manage data compression for.
- 3. For Actions, choose Update data compression type.
- 4. On the **Update data compression type** dialog box, choose **LZ4** to turn on data compression, or choose **NONE** to turn it off.
- 5. Choose Update.
- 6. You can monitor the update progress on the file systems detail page in the **Updates** tab.

### To update data compression on an existing file system (CLI)

To update the data compression configuration for an existing FSx for Lustre file system, use the AWS CLI command update-file-system. Set the following parameters:

- Set --file-system-id to the ID of the file system that you are updating.
- Set --lustre-configuration DataCompressionType to NONE to turn off data compression or LZ4 to turn on data compression with the LZ4 algorithm.

This command specifies that data compression is turned on with the LZ4 algorithm.

```
$ aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration DataCompressionType=LZ4
```

# Data compression configuration when creating a file system from backup

You can use an available backup to create a new Amazon FSx for Lustre file system. When you create a new file system from backup, there is no need to specify the DataCompressionType; the setting will be applied using the backup's DataCompressionType setting. If you choose to specify

Managing data compression 220

the DataCompressionType when creating from backup, the value must match the backup's DataCompressionType setting.

To view the settings on a backup, choose it from the **Backups** tab of the Amazon FSx console. Details of the backup will be listed on the **Summary** page for the backup. You can also run the <u>describe-backups</u> AWS CLI command (the equivalent API action is <u>DescribeBackups</u>).

# Compressing previously written files

Files are uncompressed if they were created when data compression was turned off on the Amazon FSx for Lustre file system. Turning on data compression will not automatically compress your existing uncompressed data.

You can use the lfs\_migrate command that is installed as a part of the Lustre client installation to compress existing files. For an example, see FSxL-Compression which is available on GitHub.

## Viewing file sizes

You can use the following commands to view the uncompressed and compressed sizes of your files and directories.

- du displays compressed sizes.
- du --apparent-size displays uncompressed sizes.
- 1s -1 displays uncompressed sizes.

The following examples show the output of each command with the same file.

```
$ du -sh samplefile
272M samplefile
$ du -sh --apparent-size samplefile
1.0G samplefile
$ ls -lh samplefile
-rw-r--r-- 1 root root 1.0G May 10 21:16 samplefile
```

The -h option is useful for these commands because it prints sizes in a human-readable format.

## **Using CloudWatch metrics**

You can use Amazon CloudWatch Logs metrics to view your file system usage. The LogicalDiskUsage metric shows the total logical disk usage (without compression), and the

PhysicalDiskUsage metric shows the total physical disk usage (with compression). These two metrics are available only if your file system has data compression enabled or previously had it enabled.

You can determine your file system's compression ratio by dividing the Sum of the LogicalDiskUsage statistic by the Sum of the PhysicalDiskUsage statistic.

For more information about monitoring your file system's performance, see <u>Monitoring Amazon</u> FSx for Lustre file systems.

# Lustre root squash

Root squash is an administrative feature that adds an additional layer of file access control on top of the current network-based access control and POSIX file permissions. Using the root squash feature, you can restrict root level access from clients that try to access your FSx for Lustre file system as root.

Root user permissions are required to perform administrative actions, such as managing permissions on FSx for Lustre file systems. However, root access provides unrestricted access to the users, allowing them to bypass permission checks to access, modify, or delete file system objects. Using the root squash feature, you can prevent unauthorized access or deletion of data by specifying a non-root user ID (UID) and group ID (GID) for your file system. Root users accessing the file system will automatically be converted to the specified less-privileged user/group with limited permissions that are set by the storage administrator.

The root squash feature also optionally allows you to provide a list of clients who are not affected by the root squash setting. These clients can access the file system as root, with unrestricted privileges.

### **Topics**

- How root squash works
- Managing root squash

## How root squash works

The root squash feature works by re-mapping the user ID (UID) and group ID (GID) of the root user to a UID and GID specified by the Lustre system administrator. The root squash feature also lets you optionally specify a set of clients for which UID/GID re-mapping does not apply.

Root squash 222

When you create a new FSx for Lustre file system, root squash is disabled by default. You enable root squash by configuring a UID and GID root squash setting for your FSx for Lustre file system. The UID and GID values are integers that can range from 0 to 4294967294:

- A non-zero value for UID and GID enables root squash. The UID and GID values can be different, but each must be a non-zero value.
- A value of 0 (zero) for UID and GID indicates root, and therefore disables root squash.

During file system creation, you can use the Amazon FSx console to provide the root squash UID and GID values in the **Root Squash** property, as shown in <u>To enable root squash when creating a file system (console)</u>. You can also use the RootSquash parameter with the AWS CLI or API to provide the UID and GID values, as shown in <u>To enable root squash when creating a file system</u> (CLI).

Optionally, you can also specify a list of NIDs of clients for which root squash doesn't apply. A client NID is a Lustre Network Identifier used to uniquely identify a client. You can specify the NID as either a single address or a range of addresses:

- A single address is described in standard Lustre NID format by specifying the client's IP address followed by the Lustre network ID (for example, 10.0.1.6@tcp).
- An address range is described using a dash to separate the range (for example, 10.0.[2-10].
   [1-255]@tcp).
- If you don't specify any client NIDs, there will be no exceptions to root squash.

When creating or updating your file system, you can use the **Exceptions to Root Squash** property in the Amazon FSx console to provide the list of client NIDs. In the AWS CLI or API, use the NoSquashNids parameter. For more information, see the procedures in Managing root squash.

# Managing root squash

During file system creation, root squash is disabled by default. You can enable root squash when creating a new Amazon FSx for Lustre file system from the Amazon FSx console, AWS CLI, or API.

#### To enable root squash when creating a file system (console)

1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.

2. Follow the procedure for creating a new file system described in <u>Step 1: Create your FSx for</u> Lustre file system in the *Getting started* section.

- 3. Open the **Root Squash** *optional* section.
- 4. For **Root Squash**, provide the user and group IDs with which the root user can access the file system. You can specify any whole number in the range of 1–4294967294:
  - 1. For **User ID**, specify the user ID for the root user to use.
  - 2. For **Group ID**, specify the group ID for the root user to use.
- 5. (Optional) For **Exceptions to Root Squash**, do the following:
  - 1. Choose Add client address.
  - 2. In the **Client addresses** field, specify the IP address of a client to which root squash doesn't apply, For information on the IP address format, see How root squash works.
  - 3. Repeat as needed to add more client IP addresses.
- 6. Complete the wizard as you do when you create a new file system.
- 7. Choose **Review and create**.
- 8. Review the settings you chose for your Amazon FSx for Lustre file system, and then choose **Create file system**.

When the file system is **Available**, root squash is enabled.

### To enable root squash when creating a file system (CLI)

To create an FSx for Lustre file system with root squash enabled, use the Amazon FSx CLI command <u>create-file-system</u> with the RootSquashConfiguration parameter. The corresponding API operation is <u>CreateFileSystem</u>.

For the RootSquashConfiguration parameter, set the following options:

- RootSquash The colon-separated UID:GID values that specify the user ID and group ID for the root user to use. You can specify any whole number in the range of 0–4294967294 (0 is root) for each ID (for example, 65534:65534).
- NoSquashNids Specify the Lustre Network Identifiers (NIDs) of clients to which root squash doesn't apply. For information on the client NID format, see <a href="How root squash works">How root squash works</a>.

The following example creates an FSx for Lustre file system with root squash enabled:

After successfully creating the file system, Amazon FSx returns the file system description as JSON, as shown in the following example.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            "CreationTime": 1549310341.483,
            "FileSystemId": "fs-0123456789abcdef0",
            "FileSystemType": "LUSTRE",
            "FileSystemTypeVersion": "2.15",
            "Lifecycle": "CREATING",
            "StorageCapacity": 2400,
            "VpcId": "vpc-123456",
            "SubnetIds": [
                "subnet-123456"
            ],
            "NetworkInterfaceIds": [
                "eni-039fcf55123456789"
            ],
            "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
            "ResourceARN": "arn:aws:fsx:us-east-2:123456:file-system/
fs-0123456789abcdef0",
            "Tags": [
                {
                    "Key": "Name",
```

You can also update the root squash settings of your existing file system using the Amazon FSx console, AWS CLI, or API. For example, you can change the root squash UID and GID values, add or remove client NIDs, or disable root squash.

#### To update root squash settings on an existing file system (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Navigate to **File systems**, and choose the Lustre file system that you want to manage root squash for.
- 3. For **Actions**, choose **Update root squash**. Or, in the **Summary** panel, choose **Update** next to the file system's **Root Squash** field to display the **Update Root Squash Settings** dialog box.
- 4. For **Root Squash**, update the user and group IDs with which the root user can access the file system. You can specify any whole number in the range of 0–4294967294. To disable root squash, specify 0 (zero) for both IDs.
  - 1. For **User ID**, specify the user ID for the root user to use.
  - 2. For **Group ID**, specify the group ID for the root user to use.
- 5. For **Exceptions to Root Squash**, do the following:
  - 1. Choose Add client address.
  - 2. In the **Client addresses** field, specify the IP address of a client to which root squash doesn't apply,
  - 3. Repeat as needed to add more client IP addresses.
- 6. Choose **Update**.

FSx for Lustre Lustre User Guide



#### Note

If root squash is enabled and you want to disable it, choose **Disable** instead of performing steps 4-6.

You can monitor the update progress on the file systems detail page in the **Updates** tab.

#### To update root squash settings on an existing file system (CLI)

To update the root squash settings for an existing FSx for Lustre file system, use the AWS CLI command update-file-system. The corresponding API operation is UpdateFileSystem.

Set the following parameters:

- Set --file-system-id to the ID of the file system that you are updating.
- Set the --lustre-configuration RootSquashConfiguration options as follows:
  - RootSquash Set the colon-separated UID:GID values that specify the user ID and group ID for the root user to use. You can specify any whole number in the range of 0-4294967294 (0 is root) for each ID. To disable root squash, specify 0:0 for the UID:GID values.
  - NoSquashNids Specify the Lustre Network Identifiers (NIDs) of clients to which root squash doesn't apply. Use [] to remove all client NIDs, which means there will be no exceptions to root squash.

This command specifies that root squash is enabled using 65534 as the value for the root user's user ID and group ID.

```
$ aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration RootSquashConfiguration={RootSquash="65534:65534", \
          NoSquashNids=["10.216.123.47@tcp", "10.216.12.176@tcp"]}
```

If the command is successful, Amazon FSx for Lustre returns the response in JSON format.

You can view the root squash settings of your file system in the **Summary** panel of the file system details page on the Amazon FSx console or in the response of a describe-file-systems CLI command (the equivalent API action is DescribeFileSystems).

# FSx for Lustre file system status

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

| File system status | Description  |
|--------------------|--|
| AVAILABLE          | The file system is in a healthy state, and is reachable and available for use.   |
| CREATING           | Amazon FSx is creating a new file system.  |
| DELETING           | Amazon FSx is deleting an existing file system.  |
| UPDATING           | The file system is undergoing a customer-initiated update.   |
| MISCONFIGURED      | The file system is in a failed but recoverable state.  |
| FAILED             | <ul> <li>This status can mean either of the following:</li> <li>The file system has failed and Amazon FSx can't recover it.</li> <li>When creating a new file system, Amazon FSx couldn't create the file system.</li> </ul> |

# **Tag your Amazon FSx for Lustre resources**

To help you manage your file systems and other Amazon FSx for Lustre 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

File system status 228

- 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 for Lustre 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.

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 for Lustre 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 for Lustre 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

Tag basics 229

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 for Lustre service.

You can apply tag-based resource-level permissions in your IAM policies to the Amazon FSx for Lustre 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 for Lustre 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 <u>Using cost allocation tags</u> in the *AWS Billing User Guide*.

## Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length 128 Unicode characters in UTF-8
- Maximum value length 256 Unicode characters in UTF-8
- The allowed characters for Amazon FSx for Lustre tags are: letters, numbers, and spaces representable in UTF-8, and the following characters: + = . \_ : / @.
- Tag keys and values are case-sensitive.
- The aws: prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the aws: prefix do not count against your tags per resource limit.

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

Tag restrictions 230

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 <u>Grant permission to tag resources during creation</u>. For more information about using tags to restrict access to Amazon FSx resources in IAM policies, see <u>Using tags to control access to your Amazon</u> FSx resources.

## **Amazon FSx for Lustre maintenance windows**

Amazon FSx for Lustre performs routine software patching for the Lustre software it manages. Patching occurs infrequently, typically once every several weeks. The maintenance window is your opportunity to control what day and time of the week this software patching occurs. You choose the maintenance window during file system creation. If you have no time preference, then a 30-minute default window is assigned.

Patching should require only a fraction of your 30-minute maintenance window. During these few minutes of time, your file system will be temporarily unavailable. File operations issued by clients while the file system is unavailable will transparently retry and eventually succeed after maintenance is complete. Note that the in-memory cache will be erased during maintenance, leading to higher latencies until after maintenance has been completed.

FSx for Lustre 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 Lustre will proceed with maintenance on the file system to ensure its security and reliability.

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

### To change the maintenance window using the console

1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.

Permissions and tag 231

- 2. Choose **File systems** in the navigation pane.
- 3. Choose the file system that you want to change the maintenance window for. The file system details page appears.
- 4. Choose the Maintenance tab. The maintenance window Settings panel appears.
- 5. Choose **Edit** and enter the new day and time that you want the maintenance window to start.
- 6. Choose **Save** to save your changes. The new maintenance start time is displayed in the **Settings** panel.

You can change the maintenance window for your file system using the <u>update-file-system</u> CLI command. Run the following command, replacing the file system ID with the ID for your file system, and the date and time with when you want to begin the window.

```
aws fsx update-file-system --file-system-id fs-01234567890123456 --lustre-configuration WeeklyMaintenanceStartTime=1:01:30
```

# **Managing Lustre versions**

FSx for Lustre currently supports multiple long-lerm support (LTS) Lustre versions released by the Lustre community. Newer LTS versions provide benefits such as performance enhancements, new features, and support for the latest Linux kernel versions for your client instances. You can upgrade your file systems to newer Lustre versions within minutes using the AWS Management Console, AWS CLI, or AWS SDKs.

FSx for Lustre currently supports Lustre LTS versions 2.10, 2.12, and 2.15. You can determine the LTS version of your FSx for Lustre file systems using the AWS Management Console or by using the describe-file-systems AWS CLI command.

Before you perform a Lustre version upgrade, we recommend that you follow the steps described in Best practices for Lustre version upgrades.

#### **Topics**

- Best practices for Lustre version upgrades
- Performing the upgrade

Lustre versions 232

# **Best practices for Lustre version upgrades**

We recommend following these best practices before upgrading the Lustre version of your FSx for Lustre file system:

- **Test in a non-production environment:** Test a Lustre version upgrade on a duplicate of your production file system before upgrading your production file system. This ensures a smooth upgrade process for your production workload.
- Ensure client compatibility: Verify that the Linux kernel versions running on your client instances are compatible with the Lustre version you plan to upgrade to. See <a href="Lustre file system">Lustre file system</a> and client kernel compatibility for details.

#### · Back up your data:

- For file systems not linked to S3: We recommend that you create an FSx backup before upgrading the Lustre version so that you have a known restore point for your file system. If automatic daily backups are enabled on your file system, Amazon FSx will automatically create a backup of your file system before upgrading.
- For file systems linked to S3 We recommend ensuring that all changes have been exported to S3 before upgrading. If you have enabled automatic export, check that the <a href="AgeOfOldestQueuedMessage">AgeOfOldestQueuedMessage</a> AutoExport metric is zero to confirm that all changes have been successfully exported to S3. If you have not enabled automatic export, you can run a manual data repository task (DRT) export to synchronize your file system with the S3 bucket before upgrading.

# Performing the upgrade

To upgrade your FSx for Lustre file system to a newer version, follow the listed steps:

 Unmount all clients: Before initiating the upgrade, you must unmount the file system from all client instances accessing your file system. You can verify that all clients are successfully unmounted by using the ClientConnections metric on Amazon CloudWatch - this metric should display zero connections. The upgrade process will not proceed if any clients remain connected to the file system.

You can view the list of client network identifiers (NIDs) connected to the file system in the .fsx/clientConnections file stored at the root of your file system. This file is updated every 5 minutes. You can use the cat command to display the contents of the file, as in this example:

#### cat /test/.fsx/clientConnections

2. **Upgrade the Lustre version:** You can upgrade the Lustre version of your FSx for Lustre file system using the Amazon FSx console, the AWS CLI, or the Amazon FSx API. We recommend upgrading your file systems to the latest Lustre version supported by FSx for Lustre.

#### To update the Lustre version of a file system (console)

- a. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- b. In the left navigation pane, choose **File systems**. In the **File systems** list, choose the FSx for Lustre file system that you want to update the Lustre version for.
- c. For Actions, choose Update file system Lustre version. Or, in the Summary panel, choose Update next to the file system's Lustre version field. The Update file system Lustre version dialog box appears. The Update file system Lustre version dialog box appears.
- d. For the **Select a new Lustre version** field, choose a Lustre version. The value you choose must be newer than the current Lustre version.
- e. Choose **Update**.

### To update the Lustre version of a file system (CLI)

To update the Lustre version of an FSx for Lustre file system, use the AWS CLI command <u>update-file-system</u>. (The equivalent API action is <u>UpdateFileSystem</u>.) Set the following parameters:

- Set --file-system-id to the ID of the file system that you are updating.
- Set --file-system-type-version to a newer Lustre version for the file system that you are updating.

The following example updates the file system's Lustre version from 2.12 to 2.15:

```
aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --file-system-type-version "2.15"
```

3. **Mount all clients:** You can monitor the progress of Lustre version updates by using the **Updates** tab in the Amazon FSx console or describe-file-systems in the AWS CLI. Once

Performing the upgrade 234

the Lustre version upgrade status shows as Completed, you can safely remount the file system on your client instances and resume your workload.

# Deleting a file system

You can delete an Amazon FSx for Lustre file system using the Amazon FSx console, the AWS CLI, and the Amazon FSx API. Before deleting an FSx for Lustre file system, you should unmount it from every connected Amazon EC2 instance. On S3-linked file systems, to ensure all of your data is written back to S3 before deleting your file system, you can either monitor for the AgeOfOldestQueuedMessage metric to be zero (if using automatic export) or you can run an export data repository task. If you have automatic export enabled and want to use an export data repository task, you have to disable automatic export before executing the export data repository task.

To delete a file system after unmounting from every Amazon EC2 instance:

- Using the console Follow the procedure described in Step 5: Clean up resources.
- Using the API or CLI Use the the <u>DeleteFileSystem</u> API operation or the <u>delete-file-system</u> CLI command.

Deleting a file system 235

# Protecting your data with backups

With Amazon FSx for Lustre, you can take automatic daily backups and user-initiated backups of persistent file systems that are not linked to an Amazon S3 durable data repository. Amazon FSx backups are file-system-consistent, highly durable, and incremental. To ensure high durability, Amazon FSx for Lustre stores backups in Amazon Simple Storage Service (Amazon S3) with 99.9999999% (11 9's) durability.

FSx for Lustre file system backups are block-based, incremental backups, whether they are generated using the automatic daily backup or the user-initiated backup feature. This means that when you take a backup, Amazon FSx compares the data on your file system to your previous backup at the block level. Then Amazon FSx stores a copy of all block-level changes in the new backup. Block-level data that remains unchanged since the previous backup is not stored in the new backup. The duration of the backup process depends on how much data has changed since the last backup was taken and is independent of the storage capacity of the file system. The following list illustrates backup times under different circumstances:

- The initial backup of a brand new file system with very little data takes minutes to complete.
- The initial backup of a brand new file system taken after loading TBs of data takes hours to complete.
- A second backup taken of the file system with TBs of data with minimal changes to the blocklevel data (relatively few creates/modifications) takes seconds to complete.
- A third backup of the same file system after a large amount of data has been added and modified takes hours to complete.

When you delete a backup, only the data unique to that backup is removed. Each FSx for Lustre backup contains all of the information that is needed to create a new file system from the backup, effectively restoring a point-in-time snapshot of the file system.

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

Backups are not supported on scratch file systems because these file systems are designed for temporary storage and shorter-term processing of data. Backups are not supported on file systems

linked to an Amazon S3 bucket because the S3 bucket serves as the primary data repository, and the Lustre file system does not necessarily contain the full dataset at any given time.

### **Topics**

- Backup support in FSx for Lustre
- · Working with automatic daily backups
- Working with user-initiated backups
- Using AWS Backup with Amazon FSx
- Copying backups
- Copying backups within the same AWS account
- Restoring backups
- Deleting backups

# **Backup support in FSx for Lustre**

Backups are supported only on FSx for Lustre persistent file systems that are not linked to an Amazon S3 data repository.

Amazon FSx does not support backups on scratch file systems because scratch file systems are designed for temporary storage and shorter-term processing of data. Amazon FSx does not support backups on file systems linked to an Amazon S3 bucket because the S3 bucket serves as the primary data repository and the file system does not necessarily contain the full dataset at any given time. For more information, see <a href="Deployment and storage class options">Deployment and storage class options</a> and <a href="Using data repositories">Using data repositories</a>.

# Working with automatic daily backups

Amazon FSx for Lustre can take 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*. You can set the retention period to be between 0–90 days. Setting the retention period to 0 (zero) days

FSx for Lustre Lustre User Guide

turns off automatic daily backups. The default retention period for automatic daily backups is 0 days. Automatic daily backups are deleted when the file system is deleted.



#### Note

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

You can use the AWS CLI or one of the AWS SDKs to change the backup window and backup retention period for your file systems. Use the UpdateFileSystem API operation or the updatefile-system CLI command.

# Working with user-initiated backups

Amazon FSx for Lustre enables you to manually take backups of your file systems at any time. You can do so using the Amazon FSx for Lustre console, API, or the AWS Command Line Interface (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 for Lustre console, API, or CLI, and 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

- Open the Amazon FSx for Lustre console at https://eusc-de-east-1.console.amazonawseusc.eu/fsx/.
- From the console dashboard, choose the name of the file system that you want to back up. 2.
- 3. From **Actions**, choose **Create backup**.
- In the **Create backup** dialog box that opens, provide a name for your backup. Backup names 4. 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 for Lustre 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 while Amazon FSx creates the backup. The status changes to **Transferring** while the backup is transferred to Amazon S3, until it is fully available.

# **Using AWS Backup with Amazon FSx**

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

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

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

Backups created by AWS Backup have backup type AWS\_BACKUP and are incremental relative to any other Amazon FSx backups you take of your file system. Backups taken by AWS Backup are considered user-initiated backups, and they count toward the user-initiated backup quota for

Amazon FSx. You can see and restore backups taken by AWS Backup in the Amazon FSx console, CLI, and API. However, you can't delete the backups taken by AWS Backup in the Amazon FSx console, CLI, or API. For more information about how to use AWS Backup to back up your Amazon FSx file systems, see Working with Amazon FSx File Systems in the AWS Backup Developer Guide.

# **Copying backups**

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

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

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

Cross-account backup copies are valuable for meeting your regulatory compliance requirements to copy backups to an isolated account. They also provide an additional layer of data protection to help prevent accidental or malicious deletion of backups, loss of credentials, or compromise of AWS KMS keys. Cross-account backups support fan-in (copy backups from multiple primary accounts to one isolated backup copy account) and fan-out (copy backups from one primary account to multiple isolated backup copy accounts).

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

Copying backups 240

# **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.

**JSON** 

Backup copy limitations 241

) }

For more information on IAM policies, see <u>Policies and permissions in IAM</u> 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.

# Copying backups within the same AWS account

You can copy backups of FSx for Lustre file systems using the AWS Management Console, CLI, and API, as described in the following procedures.

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

- 1. Open the Amazon FSx console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.
- 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.

Full and incremental copies 242

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

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

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

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

The response shows the description of the copied backup.

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

# **Restoring backups**

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



#### Note

You can only restore your backup to a file system of the same deployment type, storage class, throughput capacity, storage capacity, data compression type, and AWS Region as the original. You can increase your restored file system's storage capacity after it becomes available.

### To restore a file system from a backup using the console

- Open the Amazon FSx for Lustre console at https://eusc-de-east-1.console.amazonaws-1. eusc.eu/fsx/.
- From the console dashboard, choose **Backups** from the left side navigation.

Restoring backups 243

3. Choose the backup that you want to restore from the **Backups** table, and then choose **Restore** backup.

The file system creation wizard opens with most settings pre-populated based on the configuration of the file system the backup was created from. You can optionally modify the Virtual Private Cloud (VPC) configuration, or choose a newer Lustre version. Note that other configuration settings, such as Deployment Type and throughput per unit of storage, cannot be modified during restore.

- 4. Complete the wizard as you do when you create a new file system.
- 5. Choose Review and create.
- 6. Review the settings you chose for your Amazon FSx for Lustre file system, and then choose **Create file system**.

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

# **Deleting backups**

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

### To delete a backup

- 1. Open the Amazon FSx for Lustre console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.
- 2. From the console dashboard, choose **Backups** from the left side navigation.
- 3. Choose the backup that you want to delete from the **Backups** table, and then choose **Delete** backup.
- 4. In the **Delete backups** dialog box that opens, confirm that the ID of the backup identifies the backup that you want to delete.
- 5. Confirm that the check box is checked for the backup that you want to delete.
- 6. Choose **Delete backups**.

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

Deleting backups 244

# Monitoring Amazon FSx for Lustre file systems

Monitoring is an important part of maintaining the reliability, availability, and performance of your FSx for Lustre 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 Lustre 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 for Lustre
  instances and automatically launch new instances when needed.
- **Lustre logging** Monitors the enabled logging events for your file system. Lustre logging writes these events to Amazon CloudWatch Logs.
- 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 the tools with your FSx for Lustre file systems.

### **Topics**

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

# Monitoring with Amazon CloudWatch

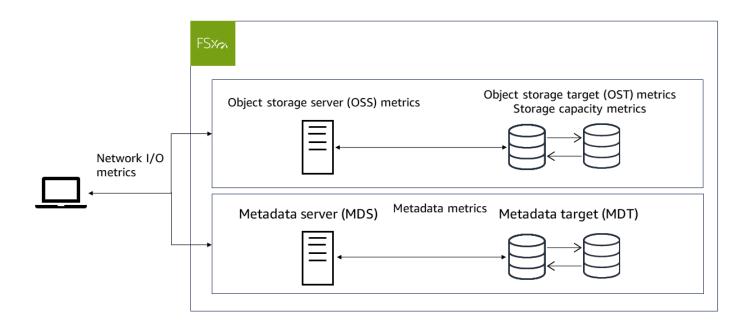
You can monitor Amazon FSx for Lustre using CloudWatch, which collects and processes raw data from Amazon FSx for Lustre into readable, near real-time metrics. These statistics are retained for a period of 15 months, so that you can access historical information and gain a better perspective on how your application or service is performing. For more information about CloudWatch, see <a href="What is Amazon CloudWatch">What is Amazon CloudWatch</a>? in the Amazon CloudWatch User Guide.

Monitoring with CloudWatch 245

CloudWatch metrics for FSx for Lustre are organized into six categories:

- Network I/O metrics Measure activity between clients and your file system.
- **Object storage server metrics** Measure object storage server (OSS) network throughput and disk throughput utilization.
- **Object storage target metrics** Measure object storage target (OST) disk throughput and disk IOPS utilization.
- **Metadata metrics** Measure metadata server (MDS) CPU utilization, metadata target (MDT) IOPS utilization, and client metadata operations.
- **Storage capacity metrics** Measure storage capacity utilization.
- **S3 data repository metrics** Measure age of oldest message waiting to be imported or exported, and renames processed by the file system.

The following diagram illustrates an FSx for Lustre file system, its components, and its metric categories.



FSx for Lustre sends metric data to CloudWatch at 1-minute intervals.

Monitoring with CloudWatch 246



### Note

Metrics may not be published during file system maintenance windows for your Amazon FSx for Lustre file system.

### **Topics**

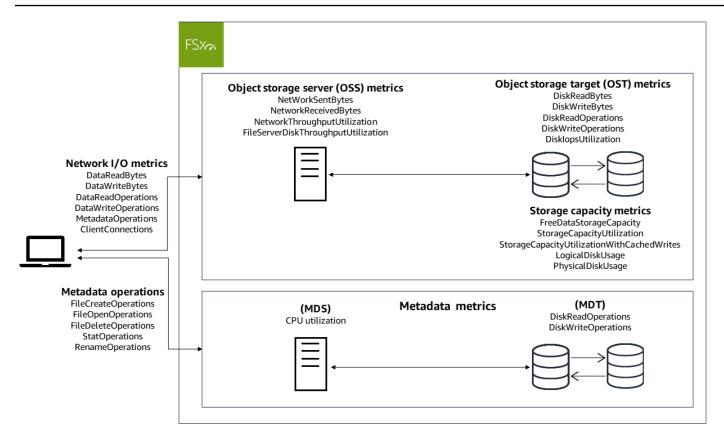
- How to use Amazon FSx for Lustre CloudWatch metrics
- Accessing CloudWatch metrics
- Amazon FSx for Lustre metrics and dimensions
- Performance warnings and recommendations
- Creating CloudWatch alarms to monitor metrics

### How to use Amazon FSx for Lustre CloudWatch metrics

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

- One or more **object storage servers (OSSs)** that serve data to clients that access the file system. Each OSS is attached to one or more storage volumes, known as object storage targets (OSTs), that host the data in your file system.
- One or more **metadata servers (MDSs)** that serve metadata to clients that access the file system. Each MDS is attached to a storage volume, known as a metadata target (MDT), that stores metadata such as filenames, directories, access permissions, and file layouts.

FSx for Lustre reports metrics in CloudWatch that track performance and resource utilization for your file system's storage and metadata servers, and their associated storage volumes. The following diagram illustrates an Amazon FSx for Lustre file system with its architectural components, and the performance and resource CloudWatch metrics that are available for monitoring.



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

### File system activity (in Summary tab)

| How do I  | Chart                                 | Relevant metrics   |
|---|---------------------------------------|--|
| determine the amount of available storage capacity on my file system? | Available storage capacity (bytes)    | FreeDataStorageCap<br>acity                              |
| determine my file system's total client throughput?                   | Total client throughpu t (bytes/se c) | SUM(DataReadBytes + DataWriteBytes )/PERIOD (in seconds) |

FSx for Lustre User Guide

| How do I   | Chart                               | Relevant metrics  |
|--|-------------------------------------|---|
| determine my file system's total client IOPS?  | Total client IOPS (operatio ns/sec) | <pre>SUM(DataReadOperations     + DataWriteOperation s    + MetadataO perations )/PERIOD (in seconds)</pre> |
| determine the number of connections that are established between clients and my file server? | Client<br>connectio<br>ns (count)   | ClientConnections   |
| determine my file system's metadata performance utilization?                                 | Metadata IOPS utilization (percent) | MAX(MDT Disk IOPS)  |

# Storage tab

| How do I  | Chart  | Relevant metrics                                   |
|---|--|--|
| determine how much storage is available?  | Available storage capacity (bytes)           | FreeDataStorageCap<br>acity                        |
| determine the percentage of used storage<br>for my file system, excluding space reserved for<br>cached writes on clients? | Total storage capacity utilization (percent) | StorageCapacityUti<br>lization                     |
| determine the percentage of used storage<br>for my file system, including space reserved for<br>cached writes on clients? | Total storage capacity utilization (percent) | StorageCapacityUti<br>lizationWithCached<br>Writes |

FSx for Lustre User Guide

| How do I   | Chart   | Relevant metrics   |
|--|---|--|
| determine the percentage of used storage for<br>my file system's OSTs excluding space reserved<br>for cached writes on clients?  | Total storage capacity utilization per OST (percent)                    | StorageCapacityUti<br>lization   |
| determine the percentage of used storage for<br>my file system's OSTs, including space reserved<br>for cached writes on clients? | Total storage capacity utilization per OST with client grants (percent) | StorageCapacityUti<br>lizationWithCached<br>Writes                       |
| determine my file system's data compression ratio?   | Compressi<br>on savings   | 100*(LogicalDi<br>skUsage - PhysicalD<br>iskUSage )/LogicalDi<br>skUsage |

# **Object storage performance (in Performance tab)**

| How do I  | Chart                               | Relevant metrics                        |
|---|-------------------------------------|---|
| determine the network throughput between<br>the clients and the OSSs as a percentage of the<br>provisioned limit? | Network<br>throughpu<br>t (percent) | NetworkThroughputU<br>tilization        |
| determine the disk throughput between<br>the OSS and its OSTs as a percentage of the<br>provisioned limit?        | Disk<br>throughpu<br>t (percent)    | FileServerDiskThro<br>ughputUtilization |
| determine the IOPS for operations that access OSTs as a percentage of the provisioned limit?                      | Disk IOPS<br>(percent)              | DiskIopsUtilization                     |

### Metadata performance (in Performance tab)

| How do I  | Chart                           | Relevant metrics   |
|---|---------------------------------|--------------------|
| determine the metadata server's CPU utilizati on percentage?                      | CPU<br>utilization<br>(percent) | CPUUtilization     |
| determine the metadata IOPS utilization as a percentage of the provisioned limit? | Metadata<br>IOPS<br>utilization | MAX(MDT Disk IOPS) |

## **Accessing CloudWatch metrics**

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

- The Amazon FSx for Lustre 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 for Lustre console

#### To view metrics using the Amazon FSx for Lustre console

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. From the navigation pane, choose **File systems**, then choose the file system that has the metrics that you want to view.
- On the Summary page, choose Monitoring & performance to see the metrics for your file system.

There are four tabs on the **Monitoring & performance** panel.

- Choose **Summary** (the default tab) to display any active warnings, CloudWatch alarms, and graphs for **File system activity**.
- Choose Storage to view storage capacity, utilization metrics, and active warnings.

 Choose Performance to view file server and storage performance metrics, and active warnings.

• Choose **CloudWatch alarms** to view graphs of any alarms configured 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.

### **Using the AWS CLI**

#### To access metrics from the AWS CLI

• Use the <u>list-metrics</u> 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 Lustre metrics and dimensions

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

#### **Topics**

- FSx for Lustre network I/O metrics
- FSx for Lustre object storage server metrics
- FSx for Lustre object storage target metrics

- FSx for Lustre metadata metrics
- FSx for Lustre storage capacity metrics
- FSx for Lustre S3 repository metrics
- FSx for Lustre dimensions

## FSx for Lustre network I/O metrics

The AWS/FSx namespace includes the following network I/O metrics. All of these metrics take one dimension, FileSystemId.

| Metric         | Description   |
|----------------|---|
| DataReadBytes  | The number of bytes from reads by clients to the file system.   |
|                | The Sum statistic is the total number of bytes associated with read operations during the specified period. The Minimum statistic is the minimum number of bytes associated with read operations on a single OST. The Maximum statistic is the maximum number of bytes associated with read operations on the OST. The Average statistic is the average number of bytes associated with read operations per OST. The SampleCount statistic is the number of OSTs. |
|                | To calculate the average throughput (bytes per second) for a period, divide the Sum statistic by the number of seconds in the period.   |
|                | Units:  |
|                | • Bytes for Sum, Minimum, Maximum, Average.   |
|                | • Count for SampleCount .   |
|                | Valid statistics: Sum, Minimum, Maximum, Average, SampleCount   |
| DataWriteBytes | The number of bytes from writes by clients to the file system.  |
|                | The Sum statistic is the total number of bytes associated with write operations. The Minimum statistic is the minimum number of bytes associated with write operations on a single OST. The Maximum   |

| Metric    | Description  |
|-----------|--|
|           | statistic is the maximum number of bytes associated with write operations on the OST. The Average statistic is the average number of bytes associated with write operations per OST. The SampleCount statistic is the number of OSTs.  To calculate the average throughput (bytes per second) for a period,                                    |
|           | divide the Sum statistic by the number of seconds in the period.  Units:   |
|           | <ul><li>Bytes for Sum, Minimum, Maximum, Average.</li><li>Count for SampleCount .</li></ul>  |
|           | Valid statistics: Sum, Minimum, Maximum, Average, SampleCount  |
| DataReadO | The number of read operations.   |
| perations | The Sum statistic is the total number of read operations. The Minimum statistic is the minimum number of read operations on a single OST. The Maximum statistic is the maximum number of read operation s on the OST. The Average statistic is the average number of read operations per OST. The SampleCount statistic is the number of OSTs. |
|           | To calculate the average number of read operations (operations per second) for a period, divide the Sum statistic by the number of seconds in the period.  |
|           | Units:   |
|           | • Count for Sum, Minimum, Maximum, Average, SampleCount .  |
|           | Valid statistics: Sum, Minimum, Maximum, Average, SampleCount  |

| Metric                  | Description   |
|-------------------------|---|
| DataWrite<br>Operations | The number of write operations.  The Sum statistic is the total number of write operations. The Minimum statistic is the minimum number of write operations on a single OST. The Maximum statistic is the maximum number of write operation s on the OST. The Average statistic is the average number of write operations per OST. The SampleCount statistic is the number of OSTs.  To calculate the average number of write operations (operations per second) for a period, divide the Sum statistic by the number of seconds in the period.  Units:  Count for Sum, Minimum, Maximum, Average, SampleCount. |

| Metric    | Description  |
|-----------|--|
| MetadataO | The number of metadata operations.   |
| perations | The Sum statistic is the count of metadata operations. The Minimum statistic is the minimum number of metadata operations per MDT. The Maximum statistic is the maximum number of metadata operations per MDT. The Average statistic is the average number of metadata operations per MDT. The SampleCount statistic is the number of MDTs.  To calculate the average number of metadata operations (operations per second) for a period, divide the Sum statistic by the number of seconds in the period.  Units:  Count for Sum, Minimum, Maximum, Average, SampleCount. |
| ClientCon |  |
| nections  | The number of active connections between clients and the file system.  Unit: Count   |
|           |  |

### **FSx for Lustre object storage server metrics**

The AWS/FSx namespace includes the following object storage server (OSS) metrics. All of these metrics take two dimensions, FileSystemId and FileServer.

- FileSystemId Your file system's AWS resource ID.
- FileServer The name of the object storage server (OSS) in your Lustre file system. Each
  OSS is provisioned with one or more object storage targets (OSTs). OSS use the naming
  convention of OSS<HostIndex>, where HostIndex represents a 4-digit hexadecimal value (for
  example, OSS0001). An OSS's ID is the ID of the first OST attached to it. For example, the first
  OSS attached to OST0000 and OST0001, will use OSS0000, and the second OSS attached to
  OST0002, OST0003 will use OSS0002.

| throughput utilization as a percentage of network throughput for your file system. Fic is equivalent to the sum of NetworkSe and NetworkReceivedBytes as a ge of the network throughput capacity of one four file system. There is one metric emitted ute for each of your file system's OSSs.   |
|--|
| rage statistic is the average network ut utilization for the given OSS over the period.  mum statistic is the lowest network throughpu on for the given OSS over one minute, for the period.  mum statistic is the highest network throughpu on for the given OSS over one minute, for the period.  ent istics: Average, Minimum, Maximum                        |
| per of bytes sent by the file system. All traffic ered in this metric, including data movement to linked data repositories. There is one metric each minute for each of your file system's OSSs. Statistic is the total number of bytes sent over ork by the given OSS over the specified period. The requirement of bytes the network by the given OSS over the |
|  |

| Metric | Description   |
|--------|---|
|        | The Minimum statistic is the lowest number of bytes sent over the network by the given OSS over the specified period.                 |
|        | The Maximum statistic is the highest number of bytes sent over the network by the given OSS over the specified period.                |
|        | To calculate sent throughput (bytes per second) for<br>any statistic, divide the statistic by the seconds in the<br>specified period. |
|        | Unit: Bytes   |
|        | Valid statistics: Sum, Average, Minimum, Maximum  |

| Metric               | Description   |
|----------------------|---|
| NetworkReceivedBytes | The number of bytes received by the file system. All traffic is considered in this metric, including data movement to and from linked data repositories. There is one metric emitted each minute for each of your file system's OSSs. |
|                      | The Sum statistic is the total number of bytes received over the network by the given OSS over the specified period.  |
|                      | The Average statistic is the average number of bytes received over the network by the given OSS over the specified period.  |
|                      | The Minimum statistic is the lowest number of bytes received over the network by the given OSS over the specified period.   |
|                      | The Maximum statistic is the highest number of bytes received over the network by the given OSS over the specified period.  |
|                      | To calculate throughput (bytes per second) for any statistic, divide the statistic by the seconds in the specified period.  |
|                      | Unit: Bytes   |
|                      | Valid statistics: Sum, Average, Minimum, Maximum  |

| The disk throughput between your OSS and associate ughputUtilization  The disk throughput between your OSS and associate d OSTs, as a percentage of the provisioned limit determined by throughput capacity. This metric is equivalent to the sum of DiskReadBytes and DiskWriteBytes as a percentage of the OSS's disk throughput capacity for your file system. There is one metric emitted each minute for each of your file system's OSSs.  The Average statistic is the average OSS disk throughput utilization for the given OSS over the | Metric | Description   |
|---|--------|---|
| The Minimumstatistic is the lowest OSS disk throughput utilization for the given OSS over the specified period.  The Maximum statistic is the highest OSS disk throughput utilization for the given OSS over the specified period.  Unit: Percent  Valid statistics: Average, Minimum, Maximum  |        | d OSTs, as a percentage of the provisioned limit determined by throughput capacity. This metric is equivalent to the sum of DiskReadBytes and DiskWriteBytes as a percentage of the OSS's disk throughput capacity for your file system. There is one metric emitted each minute for each of your file system's OSSs.  The Average statistic is the average OSS disk throughput utilization for the given OSS over the specified period.  The Minimumstatistic is the lowest OSS disk throughput utilization for the given OSS over the specified period.  The Maximum statistic is the highest OSS disk throughput utilization for the given OSS over the specified period.  Unit: Percent |

## **FSx for Lustre object storage target metrics**

The AWS/FSx namespace includes the following object storage target (OST) metrics. All of these metrics take two dimensions, FileSystemId and StorageTargetId.



### Note

DiskReadOperations and DiskWriteOperations metrics are not available on Scratch file systems, and DiskIopsUtilization metrics are not available on Scratch and Persistent HDD file systems.

| Metric         | Description   |
|----------------|---|
| DiskReadBytes  | The number of bytes (disk IO) from any disk reads from this OST. There is one metric emitted each minute for each of your file system's OSTs.   |
|                | The Sum statistic is the total number of bytes read in a minute from the given OST over the specified period.                                   |
|                | The Average statistic is the average number of bytes read each minute from the given OST over the specified period.                             |
|                | The Minimum statistic is the lowest number of bytes read each minute from the given OST over the specified period.                              |
|                | The Maximum statistic is the highest number of bytes read each minute from the given OST over the specified period.                             |
|                | To calculate read disk throughput (bytes per second) for any statistic, divide the statistic by the seconds in the period.                      |
|                | Unit: Bytes   |
|                | Valid statistics: Sum, Average, Minimum, and, Maximum   |
| DiskWriteBytes | The number of bytes (disk IO) from any disk writes from this OST.  There is one metric emitted each minute for each of your file system's OSTs. |
|                | The Sum statistic is the total number of bytes written each minute from the given OST over the specified period.                                |
|                | The Average statistic is the average number of bytes written each minute from the given OST over the specified period.                          |
|                | The Minimum statistic is the lowest number of bytes written each minute from the given OST over the specified period.                           |
|                | The Maximum statistic is the highest number of bytes written each minute from the given OST over the specified period.                          |

| Metric                 | Description   |
|------------------------|---|
|                        | To calculate read disk throughput (bytes per second) for any statistic, divide the statistic by the seconds in the period         |
|                        | Unit: Bytes   |
|                        | Valid statistics: Sum, Average, Minimum, and, Maximum   |
| DiskReadO<br>perations | The number of read operations (disk IO) to this OST. There is one metric emitted each minute for each of your file system's OSTs. |
|                        | The Sum statistic is the total number of read operations performed by the given OST over the specified period.                    |
|                        | The Average statistic is the average number of read operations performed each minute by the given OST over the specified period.  |
|                        | The Minimum statistic is the lowest number of read operations performed each minute by the given OST over the specified period.   |
|                        | The Maximum statistic is the highest number of read operations performed each minute by the given OST over the specified period.  |
|                        | To calculate average disk IOPS over the period, use the Average statistic and divide the result by 60 (seconds).                  |
|                        | Units: Count  |
|                        | Valid statistics: Sum, Average, Minimum, and Maximum  |

| Metric                  | Description  |
|-------------------------|--|
| DiskWrite<br>Operations | The number of write operations (disk IO) to this OST. There is one metric emitted each minute for each of your file system's OSTs.                               |
|                         | The Sum statistic is the total number of write operations performed by the given OST over the specified period.  |
|                         | The Average statistic is the average number of write operations performed each minute by the given OST over the specified period.                                |
|                         | The Minimum statistic is the lowest number of write operations performed each minute by the given OST over the specified period.                                 |
|                         | The Maximum statistic is the highest number of write operations performed each minute by the given OST over the specified period.                                |
|                         | To calculate average disk IOPS over the period, use the Average statistic and divide the result by 60 (seconds).   |
|                         | Units: Count   |
|                         | Valid statistics: Sum, Average, Minimum, and Maximum   |
| DiskIopsU<br>tilization | The disk IOPS utilization of one OST, as a percentage of the OST's disk IOPS limit. There is one metric emitted each minute for each of your file system's OSTs. |
|                         | The Average statistic is the average disk IOPS utilization for the given OST over the specified period.  |
|                         | The Minimum statistic is the lowest disk IOPS utilization for the given OST over the specified period.   |
|                         | The Maximum statistic is the highest disk IOPS utilization for the given OST over the specified period.  |
|                         | Unit: Percent  |
|                         | Valid statistics: Average, Minimum, and Maximum  |

### **FSx for Lustre metadata metrics**

The AWS/FSx namespace includes the following metadata metrics. The CPUUtilization metric takes the FileSystemId and FileServer dimensions, while the other metrics take the FileSystemId and StorageTargetId dimensions.

- FileSystemId Your file system's AWS resource ID.
- StorageTargetId The name of the metadata target (MDT). MDTs use the naming convention
  of MDT<MDTIndex> (for example, MDT0001).
- FileServer The name of the metadata server (MDS) in your Lustre file system. Each
  MDS is provisioned with one metadata target (MDT). MDS use the naming convention of
  MDS<HostIndex>, where HostIndex represents a 4-digit hexadecimal value derived using
  the MDT index on the server. For example, the first MDS provisioned with MDT0000 will use
  MDS0000, and the second MDS provisioned with MDT0001 will use MDS00001. Your file system
  contains multiple metadata servers if your file system has a metadata configuration specified.

| Metric         | Description   |
|----------------|---|
| CPUUtilization | The percent utilization of your file system's MDS CPU resources. There is one metric emitted each minute for each of your file system's MDSs. |
|                | The Average statistic is the average CPU utilization of the MDS over a specified period.  |
|                | The Minimum statistic is the lowest CPU utilization for the given MDS over the specified period.  |
|                | The Maximum statistic is the highest CPU utilization for the given MDS over the specified period.   |
|                | Unit: Percent   |

| Metric               | Description  |
|----------------------|--|
|                      | Valid statistics: Average, Minimum and Maximum   |
| FileCreateOperations | Total number of file create operations.  |
|                      | Unit: Count  |
| FileOpenOperations   | Total number of file open operations.  |
|                      | Unit: Count  |
| FileDeleteOperations | Total number of file delete operations.  |
|                      | Unit: Count  |
| StatOperations       | Total number of stat operations.   |
|                      | Unit: Count  |
| RenameOperations     | Total number of directory renames, whether in-place directory renames or cross directory renames.  Unit: Count |
|                      | orne. Court  |

# **FSx for Lustre storage capacity metrics**

The AWS/FSx namespace includes the following storage capacity metrics. All of these metrics take two dimensions, FileSystemId and StorageTargetId except LogicalDiskUsage and PhysicalDiskUsage which take the FileSystemId dimension.

| Metric                  | Description   |
|-------------------------|---|
| FreeDataStorageCapacity | The amount of available storage capacity in this OST.  There is one metric emitted each minute for each of your file system's OSTs. |

| Metric                     | Description  |
|----------------------------|--|
|                            | The Sum statistic is the total number of bytes available in the given OST over the specified period.                                       |
|                            | The Average statistic is the average number of bytes available in the given OST over the specified period.                                 |
|                            | The Minimum statistic is the lowest number of bytes available in the given OST over the specified period.                                  |
|                            | The Maximum statistic is the highest number of bytes available in the given OST over the specified period.                                 |
|                            | Unit: Bytes  |
|                            | Valid statistics: Sum, Average, Minimum, and Maximum   |
| StorageCapacityUtilization | The storage capacity utilization for a given file system OST. There is one metric emitted each minute for each of your file system's OSTs. |
|                            | The Average statistic is the average amount of storage capacity utilization for a given OST over a specified period.                       |
|                            | The Minimum statistic is the minimum amount of storage capacity utilization for a given OST over a specified period.                       |
|                            | The Maximum statistic is the maximum amount of storage capacity utilization for a given OST over a specified period.                       |
|                            | Unit: Percent  |
|                            | Valid statistics: Average, Minimum, Maximum  |

| Metric   | Description   |
|--|---|
| StorageCapacityUti<br>lizationWithCachedWrites | The storage capacity utilization for a given file system OST including space reserved for cached writes on the client. There is one metric emitted each minute for each of your file system's OSTs.  The Average statistic is the average amount of storage |
|  | capacity utilization for a given OST over a specified period.   |
|  | The Minimum statistic is the minimum amount of storage capacity utilization for a given OST over a specified period.  |
|  | The Maximum statistic is the maximum amount of storage capacity utilization for a given OST over a specified period.  |
|  | Unit: Percent   |
|  | Valid statistics: Average, Minimum, Maximum   |

| Metric           | Description   |
|------------------|---|
| LogicalDiskUsage | The amount of logical data stored (uncompressed).   |
|                  | The Sum statistic is the total number of logical bytes stored in the file system. The Minimum statistic is the least number of logical bytes stored in an OST in the file system. The Maximum statistic is the largest number of logical bytes stored in an OST in the file system. The Average statistic is the average number of logical bytes stored per OST. The SampleCount statistic is the number of OSTs.  Units:  • Bytes for Sum, Minimum, Maximum. |
|                  | • Count for SampleCount .   |
|                  | Valid statistics: Sum, Minimum, Maximum, Average, SampleCount   |

| Metric            | Description  |
|-------------------|--|
| PhysicalDiskUsage | The amount of storage physically occupied by file system data (compressed).  |
|                   | The Sum statistic is the total number of bytes occupied in OSTs in the file system. The Minimum statistic is the total number of bytes occupied in the emptiest OST. The Maximum statistic is the total number of bytes occupied in the fullest OST. The Average statistic is the average number of bytes occupied per OST. The SampleCount statistic is the number of OSTs. |
|                   | Units:   |
|                   | • Bytes for Sum, Minimum, Maximum.   |
|                   | • Count for SampleCount .  |
|                   | Valid statistics: Sum, Minimum, Maximum, Average, SampleCount  |

## **FSx for Lustre S3 repository metrics**

FSx for Lustre publishes the following AutoImport (automatic import) and AutoExport (automatic export) metrics into the FSx namespace in CloudWatch. These metrics use dimensions to enable more granular measurements of your data. All AutoImport and AutoExport metrics have the FileSystemId and Publisher dimensions.

| Metric  | Description   |
|---|---|
| AgeOfOldestQueuedMessage  Dimension: AutoExport | The age, in seconds, of the oldest message waiting to be exported.  |
|   | The Average statistic is the average age of the oldest message waiting to be exported.  The Maximum statistic is the maximum number of seconds a message lived in the |

| Metric  | Description  |
|---|--|
|   | export queue. The Minimum statistic is the minimum number of seconds a message lived in the export queue. A value of zero indicates that no messages are waiting to be exported.  Units: Seconds   |
|   | Valid statistics: Average, Minimum,<br>Maximum   |
| RepositoryRenameOperations  Dimension: AutoExport | The number of renames processed by the file system in response to a larger directory rename.   |
|   | The Sum statistic is the total number of rename operations that result from a directory rename. The Average statistic is the average number of rename operations for the file system. The Maximum statistic is the maximum number of rename operation s associated with a directory rename on the file system. The Minimum statistic is the minimum number of renames associated with a directory rename on the file system.  Units: Count |
|   | Valid statistics: Sum, Average, Minimum, Maximum,  |

| Metric  | Description  |
|---|--|
| AgeOfOldestQueuedMessage  Dimension: AutoImport | The age, in seconds, of the oldest message waiting to be imported.  The Average statistic is the average age of the oldest message waiting to be imported. The Maximum statistic is the maximum number of seconds a message lived in the import queue. The Minimum statistic is the minimum number of seconds a message lived in the import queue. A value of zero indicates that no messages are waiting to be imported.  Units: Seconds  Valid statistics: Average, Minimum, Maximum |

### **FSx for Lustre dimensions**

Amazon FSx for Lustre metrics use the AWS/FSx namespace and use the following dimensions.

- The FileSystemId dimension denotes a file system's ID and filters the metrics that you request to that individual file system. You can find the ID on the Amazon FSx console on the **Summary** panel of the file system details page, in the **File system ID** field. The file system ID takes the form of *fs-01234567890123456*. You can also see the ID in the response of a <u>describe-file-systems</u> CLI command (the equivalent API action is <u>DescribeFileSystems</u>).
- The StorageTargetId dimension denotes which OST (object storage target) or MDT (metadata target) published the metadata metrics. A StorageTargetId takes the form of OSTxxxx (for example, OST0001) or MDTxxxx (for example, MDT0001).
- The FileServer dimension denotes the following
  - For OSS metrics: the name of the object storage server (OSS). OSS use the OSSxxxx naming convention (for example, OSS0002).

• For the CPUUtilization metric: the name of a metadata server (MDS). MDS use the MDSxxxx naming convention (for example, MDS0002).

• The Publisher dimension is available in CloudWatch and AWS CLI for the AutoImport and AutoImport metrics to denote which service published the metrics.

For more information about dimensions, see Dimensions in the Amazon CloudWatch User Guide.

### Performance warnings and recommendations

FSx for Lustre 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 for Lustre 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. These warnings automatically disappear from the dashboard 24 hours after the underlying metrics fall below the warning threshold.

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, storage capacity, or metadata IOPS, as described in the following table.

| Dashboard section | If there's a warning for this metric | Do this                                       |
|-------------------|--------------------------------------|---|
| Storage           | Storage capacity utilization         | Increase your file system's storage capacity. |

FSx for Lustre User Guide

| Dashboard section                    | If there's a warning for this metric            | Do this  |
|--------------------------------------|---|--|
|                                      |   | If your storage capacity utilizati on is only higher for a subset of your file system's Object Storage Targets (OSTs), then you can also rebalance your workload so that your storage capacity utilization is more evenly balanced across your file system.                                  |
|                                      | Storage capacity utilization with cached writes | Reduce the size of your client write cache by configuring the max_dirty_mb parameter on your clients.  |
| Object<br>storage<br>performan<br>ce | Network throughput                              | Increase your file system's throughput capacity.  If your throughput utilization is higher for a subset of your file system's Object Storage Servers (OSSs), then you can also rebalance your workload so that your throughput utilizati on is more evenly balanced across your file system. |

| Dashboard section | If there's a warning for this metric | Do this  |
|-------------------|--------------------------------------|--|
|                   | Disk throughput                      | Increase your file system's throughput capacity.  If your disk throughput utilizati on is higher for a subset of your file system's Object Storage Servers (OSSs), then you can also rebalance your workload so that your disk throughput utilization is more evenly balanced across your file system. |
|                   | Disk IOPS                            | Increase your file system's storage capacity.  If your disk IOPS utilization is higher for a subset of your file system's Object Storage Targets (OSTs), then you can also rebalance your workload so that your disk IOPS utilizati on is more evenly balanced across your file system.                |

| Dashboard section           | If there's a warning for this metric | Do this   |
|-----------------------------|--------------------------------------|---|
| Metadata<br>performan<br>ce | CPU utilization                      | Increase your file system's storage capacity.  If you need to scale metadata performance independent of the storage capacity, you can migrate to a new file system that supports provision ing metadata performance independent of the storage capacity using the MetadataC onfiguration parameter. |
|                             | Metadata IOPS                        | Increase your file system's metadata IOPS.  |

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

## **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 for Lustre using the console, AWS CLI, and API.

### To set alarms using the Amazon FSx for Lustre console

1. Open the Amazon FSx console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.

Creating CloudWatch alarms 275

From the navigation pane, choose **File systems**, and then choose the file system that you want to create the alarm for.

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

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.



### Marning

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.

- 10. Fill in the Name, Description, and Whenever values for the metric, and choose Next.
- 11. On the **Preview and create** page, review the alarm and choose **Create Alarm**.

### To set alarms using the CloudWatch console

Sign in to the AWS Management Console and open the CloudWatch console at https://euscde-east-1.console.amazonaws-eusc.eu/cloudwatch/.

- Choose Create Alarm to start the Create Alarm Wizard.
- 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**.
- Enter a **Name** and a **Description**, and choose a **Whenever** value for the metric. 4.
- 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.



#### Marning

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.

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 with Amazon CloudWatch Logs

FSx for Lustre supports logging of error and warning events for data repositories associated with your file system to Amazon CloudWatch Logs.



### Note

Logging with Amazon CloudWatch Logs is only available on Amazon FSx for Lustre file systems created after 3pm PST on November 30, 2021.

### **Topics**

- Logging overview
- Log destinations
- Managing logging
- Viewing logs

## **Logging overview**

If you have data repositories linked to your FSx for Lustre file system, you can enable logging of data repository events to Amazon CloudWatch Logs. Error and warning events can be logged for import, export, and restore events. For more information on these operations and on linking to data repositories, see Using data repositories with Amazon FSx for Lustre.

You can configure the log levels that Amazon FSx logs; that is, whether Amazon FSx will log only error events, only warning events, or both error and warning events. You can also turn event logging off at any time.



#### Note

We highly recommend that you enable logs for file systems that have any level of critical functionality associated with them.

### Log destinations

When logging is enabled, FSx for Lustre must be configured with an Amazon CloudWatch Logs destination. The event log destination is an Amazon CloudWatch Logs log group, and Amazon FSx creates a log stream for your file system within this log group. CloudWatch Logs allows you to store, view, and search audit event logs in the Amazon CloudWatch console, run queries on the logs using CloudWatch Logs Insights, and trigger CloudWatch alarms or Lambda functions.

278 Logging overview

You choose the log destination when you create your FSx for Lustre file system or afterwards by updating it. For more information, see Managing logging.

By default, Amazon FSx will create and use a default CloudWatch Logs log group in your account as the event log destination. If you want to use a custom CloudWatch Logs log group as the event log destination, here are the requirements for the name and location of the event log destination:

- The name of the CloudWatch Logs log group must begin with the /aws/fsx/ prefix.
- If you don't have an existing CloudWatch Logs log group when you create or update a file system on the console, Amazon FSx for Lustre can create and use a default log stream in the CloudWatch Logs /aws/fsx/lustre log group. The log stream will be created with the format datarepo\_file\_system\_id (for example, datarepo\_fs-0123456789abcdef0).
- If you don't want to use the default log group, the configuration UI lets you create a CloudWatch Logs log group when you create or update your file system on the console.
- The destination CloudWatch Logs log group must be in the same AWS partition, AWS Region, and AWS account as your Amazon FSx for Lustre file system.

You can change the event log destination at any time. When you do so, new event logs are sent only to the new destination.

# **Managing logging**

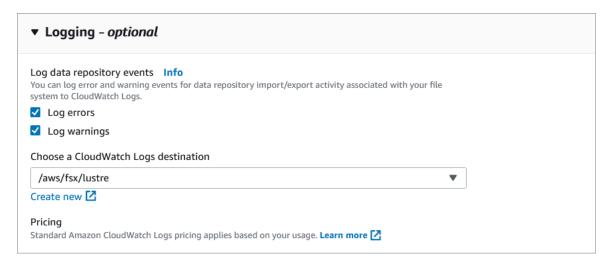
You can enable logging when you create a new FSx for Lustre file system or afterwards by updating it. Logging is turned on by default when you create a file system from the Amazon FSx console. However, logging is turned off by default when you create a file system with the AWS CLI or Amazon FSx API.

On existing file systems that have logging enabled, you can change the event logging settings, including which log level to log events for, and the log destination. You can perform these tasks using the Amazon FSx console, AWS CLI, or Amazon FSx API.

#### To enable logging when creating a file system (console)

- 1. Open the Amazon FSx console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/">https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/</a>.
- 2. Follow the procedure for creating a new file system described in <u>Step 1: Create your FSx for</u> Lustre file system in the Getting Started section.
- 3. Open the **Logging optional** section. Logging is enabled by default.

Managing logging 279



4. Continue with the next section of the file system creation wizard.

When the file system becomes **Available**, logging will be enabled.

#### To enable logging when creating a file system (CLI)

1. When creating a new file system, use the LogConfiguration property with the CreateFileSystem operation to enable logging for the new file system.

2. When the file system becomes **Available**, logging feature will be enabled.

#### To change the logging configuration (console)

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. Navigate to **File systems**, and choose the Lustre file system that you want to manage logging for.
- 3. Choose the **Data repository** tab.
- 4. On the **Logging** panel, choose **Update**.
- 5. On the **Update logging configuration** dialog, change the desired settings.

Managing logging 280

a. Choose **Log errors** to log only error events, or **Log warnings** to log only warning events, or both. Logging is disabled if you don't make a selection.

- b. Choose an existing CloudWatch Logs log destination or create a new one.
- 6. Choose **Save**.

#### To change the logging configuration (CLI)

Use the <u>update-file-system</u> CLI command or the equivalent <u>UpdateFileSystem</u> API operation.

# **Viewing logs**

You can view the logs after Amazon FSx has started emitting them. You can view the logs as follows:

- You can view logs by going to the Amazon CloudWatch console and choosing the log group and log stream to which your event logs are sent. For more information, see <u>View log data sent to</u> <u>CloudWatch Logs in the Amazon CloudWatch Logs User Guide</u>.
- You can use CloudWatch Logs Insights to interactively search and analyze your log data. For more information, see <u>Analyzing log data with CloudWatch Logs Insights</u>, in the *Amazon CloudWatch Logs User Guide*.
- You can also export logs to Amazon S3. For more information, see <a href="Exporting log data to Amazon S3"><u>Exporting log data to Amazon S3</u></a>, in the *Amazon CloudWatch Logs User Guide*.

To learn about failure reasons, see Data repository event logs.

# Logging FSx for Lustre API calls with AWS CloudTrail

Amazon FSx for Lustre 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 for Lustre. CloudTrail captures all API calls

Viewing logs 281

for Amazon FSx for Lustre as events. Captured calls include calls from the Amazon FSx for Lustre console and from code calls to Amazon FSx for Lustre 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 for Lustre. 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 for Lustre. 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 for Lustre information in CloudTrail

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

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

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

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

 Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.

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

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

## **Understanding Amazon FSx for Lustre log file entries**

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

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

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T22:36:07Z"
            }
        }
    },
    "eventTime": "2018-11-14T22:36:07Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
```

```
"resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-
ab12cd34ef56gh789"
    },
    "responseElements": null,
    "requestID": "aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51",
    "eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": "111122223333"
}
```

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

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

```
"eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
    "eventType": "AwsApiCall",
    "apiVersion": "2018-03-01",
    "recipientAccountId": "111122223333"
}
```

FSx for Lustre Lustre User Guide

# Migrating to Amazon FSx for Lustre using AWS DataSync

You can use AWS DataSync to transfer data between FSx for Lustre 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 Direct Connect. DataSync can transfer your file system data and metadata, such as ownership, timestamps, and access permissions.

# How to migrate existing files to FSx for Lustre using AWS **DataSync**

You can use DataSync with FSx for Lustre file systems to perform one-time data migrations, periodically ingest data for distributed workloads, and schedule replication for data protection and recovery. For information about specific transfer scenarios, see Where can I transfer my data with AWS DataSync? in the AWS DataSync User Guide.

# **Prerequisites**

To migrate data into your FSx for Lustre setup, you need a server and network that meet the DataSync requirements. To learn more, see Setting up with AWS DataSync in the AWS DataSync User Guide.

- You have created a destination FSx for Lustre file system. For more information, see Step 1: Create your FSx for Lustre 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 Site-to-Site VPN. For more information, see What is VPC peering? in the Amazon VPC Peering Guide.



#### Note

DataSync can only transfer across AWS accounts to or from FSx for Lustre if the other transfer location is Amazon S3.

# Basic steps for migrating files using DataSync

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

1. Download and deploy an agent in your environment and activate it (not required if transferring between AWS services).

- 2. Create a source and destination location.
- 3. Create a task.
- 4. 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:

- Transferring between on-premises storage and AWS
- Configuring AWS DataSync transfers with Amazon FSx for Lustre.
- Deploying your Amazon EC2 agent

# **Security in Amazon FSx for Lustre**

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 <u>shared responsibility model</u> describes this as security *of* the cloud and security *in* the cloud:

- Security of the cloud AWS is responsible for protecting the infrastructure that runs AWS services in the Amazon Web Services Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the <u>AWS Compliance Programs</u>. To learn about the compliance programs that apply to Amazon FSx for Lustre, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon FSx for Lustre. The following topics show you how to configure Amazon FSx to meet your security and compliance objectives. You also learn how to use other Amazon services that help you to monitor and secure your Amazon FSx for Lustre resources.

Following, you can find a description of security considerations for working with Amazon FSx.

#### **Topics**

- Data protection in Amazon FSx for Lustre
- · Identity and access management for Amazon FSx for Lustre
- File system access control with Amazon VPC
- Amazon VPC network ACLs
- Compliance Validation for Amazon FSx for Lustre
- Amazon FSx for Lustre and interface VPC endpoints (AWS PrivateLink)

# **Data protection in Amazon FSx for Lustre**

The AWS applies to data protection in Amazon FSx for Lustre. 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 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 <u>Working with CloudTrail trails</u> 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**

- Data encryption in Amazon FSx for Lustre
- Internetwork traffic privacy

Data protection 289

FSx for Lustre Lustre User Guide

# **Data encryption in Amazon FSx for Lustre**

Amazon FSx for Lustre supports two forms of encryption for file systems, encryption of data at rest and encryption in transit. Encryption of data at rest is automatically enabled when creating an Amazon FSx file system. Encryption of data in transit is automatically enabled when you access an Amazon FSx file system from Amazon EC2 instances that support this feature.

### When to use encryption

If your organization is subject to corporate or regulatory policies that require encryption of data and metadata at rest, we recommend creating an encrypted file system and mounting your file system using encryption of data in transit.

For more information about creating a file system encrypted at rest using the console, see Create your Amazon FSx for Lustre file system.

#### **Topics**

- Encrypting data at rest
- Encrypting data in transit

### **Encrypting data at rest**

Encryption of data at rest is automatically enabled when you create an Amazon FSx for Lustre 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. If you create a persistent file system, you can specify the AWS KMS key to encrypt the data with. If you create a scratch file system, the data is encrypted using keys managed by Amazon FSx. For more information about creating a file system encrypted at rest using the console, see Create your Amazon FSx for Lustre file system.



#### (i) 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.

Data encryption 290

For more information on how FSx for Lustre uses AWS KMS, see <u>How Amazon FSx for Lustre uses</u> 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 Lustre, so you don't have to modify your applications.

Amazon FSx for Lustre uses industry-standard AES-256 encryption algorithm to encrypt file system data at rest. For more information, see <u>Cryptography Basics</u> in the AWS Key Management Service Developer Guide.

#### **How Amazon FSx for Lustre uses AWS KMS**

Amazon FSx for Lustre encrypts data automatically before it is written to the file system, and automatically decrypts data as it is read. Data is encrypted using an XTS-AES-256 block cipher. All scratch FSx for Lustre file systems are encrypted at rest with keys managed by AWS KMS. Amazon FSx for Lustre integrates with AWS KMS for key management. The keys used to encrypt scratch file systems at-rest are unique per file system and destroyed after the file system is deleted. For persistent file systems, you choose the KMS key used to encrypt and decrypt data. You specify which key to use when you create a persistent file system. 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 <a href="AWS Key Management Service pricing">AWS Key Management Service pricing</a>.
- 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, reenable, delete, or revoke access to your customer managed key at any time.

Data encryption 291

FSx for Lustre Lustre User Guide

#### Important

Amazon FSx accepts only symmetric encryption 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 plaintext into ciphertext. This permission is included in the default key policy.
- kms:Decrypt (Required) Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted. This permission is included in the default key policy.
- kms:ReEncrypt (Optional) Encrypts data on the server side with a new KMS key, without exposing the plaintext of the data on the client side. The data is first decrypted and then reencrypted. 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.

Data encryption 292

### **Encrypting data in transit**

Scratch 2 and persistent file systems can automatically encrypt data in transit when the file system is accessed from Amazon EC2 instances that support encryption in transit, and also for all communications between hosts within the file system. To learn which EC2 instances support encryption in transit, see <a href="Encryption in transit">Encryption in transit</a> in the Amazon EC2 User Guide.

For a list of AWS Regions in which Amazon FSx for Lustre is available, see <u>Deployment type</u> availability.

## Internetwork traffic privacy

This topic describes how Amazon FSx secures connections from the service to other locations.

## Traffic between Amazon FSx and on-premises clients

You have two connectivity options between your private network and AWS:

- An AWS Site-to-Site VPN connection. For more information, see What is AWS Site-to-Site VPN?
- An AWS Direct Connect connection. For more information, see What is AWS Direct Connect?

You can access FSx for Lustre over the network to reach AWS-published API operations for performing administrative tasks and Lustre ports to interact with the file system.

### **Encrypting API traffic**

To access AWS-published API operations, clients must support Transport Layer Security (TLS) 1.2 or later. We require TLS 1.2 and recommend TLS 1.3. Clients must also support cipher suites with Perfect Forward Secrecy (PFS), such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). 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 <a href="AWS Security Token Service">AWS Security Token Service</a> (STS) to generate temporary security credentials to sign requests.

### **Encrypting data traffic**

Encryption of data in transit is enabled from supported EC2 instances accessing the file systems from within the AWS Cloud. For more information, see <a href="Encrypting data in transit">Encrypting data in transit</a>. FSx for Lustre does not natively offer encryption in transit between on-premise clients and file systems.

Internetwork traffic privacy 293

# Identity and access management for Amazon FSx for Lustre

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 Lustre works with IAM
- Identity-based policy examples for Amazon FSx for Lustre
- AWS managed policies for Amazon FSx for Lustre
- Troubleshooting Amazon FSx for Lustre identity and access
- Using tags with Amazon FSx
- Using service-linked roles for Amazon FSx

#### **Audience**

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** request permissions from your administrator if you cannot access features (see Troubleshooting Amazon FSx for Lustre identity and access)
- **Service administrator** determine user access and submit permission requests (see <u>How Amazon</u> FSx for Lustre works with IAM)
- IAM administrator write policies to manage access (see <u>Identity-based policy examples for</u> Amazon FSx for Lustre)

# **Authenticating with identities**

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

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see <a href="How to sign in to your AWS account">How to sign in to your AWS account</a> in the AWS Sign-In User Guide.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

#### AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see <u>Tasks</u> that require root user credentials in the *IAM User Guide*.

### **Federated identity**

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

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the AWS IAM Identity Center User Guide.

## IAM users and groups

An <u>IAM user</u> is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see <u>Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.</u>

An <u>IAM group</u> specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

Authenticating with identities 295

#### IAM roles

An <u>IAM role</u> is an identity with specific permissions that provides temporary credentials. You can assume a role by <u>switching from a user to an IAM role (console)</u> or by calling an AWS CLI or AWS API operation. For more information, see <u>Methods to assume a role</u> in the <u>IAM User Guide</u>.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the IAM User Guide.

## Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see <a href="Overview of JSON policies">Overview of JSON policies</a> in the IAM User Guide.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

## **Identity-based policies**

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see <a href="Define custom IAM">Define custom IAM</a> permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see <a href="Choose between managed policies and inline policies">Choose between managed policies and inline policies</a> in the *IAM User Guide*.

## **Resource-based policies**

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-

based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

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

### Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see <u>Permissions boundaries for IAM entities</u> in the *IAM User Guide*.
- **Service control policies (SCPs)** Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see <u>Service control policies</u> in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** Set the maximum available permissions for resources in your accounts. For more information, see <u>Resource control policies (RCPs)</u> in the *AWS Organizations User Guide*.
- **Session policies** Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

## Multiple policy types

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

## How Amazon FSx for Lustre 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 Lustre

| IAM feature                   | Amazon FSx support |
|-------------------------------|--------------------|
| Identity-based policies       | Yes                |
| Resource-based policies       | No                 |
| Policy actions                | Yes                |
| Policy resources              | Yes                |
| Policy condition keys         | Yes                |
| ACLs                          | No                 |
| ABAC (tags in policies)       | Yes                |
| Temporary credentials         | Yes                |
| Forward access sessions (FAS) | 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. 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 <u>Identity-based policy examples for</u> Amazon FSx for Lustre.

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

To see a list of Amazon FSx actions, see <u>Actions defined by Amazon FSx for Lustre</u> in the <u>Service</u> 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"
    ]
```

To view examples of Amazon FSx identity-based policies, see <u>Identity-based policy examples for</u> Amazon FSx for Lustre.

# 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. As a best practice, specify a resource using its <u>Amazon Resource Name (ARN)</u>. For actions that don't support resource-level permissions, use a wildcard (\*) to indicate that the statement applies to all resources.

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

To see a list of Amazon FSx resource types and their ARNs, see <u>Resources defined by Amazon FSx</u> <u>for Lustre</u> in the <u>Service Authorization Reference</u>. To learn with which actions you can specify the ARN of each resource, see <u>Actions defined by Amazon FSx</u> for <u>Lustre</u>.

To view examples of Amazon FSx identity-based policies, see <u>Identity-based policy examples for</u> Amazon FSx for Lustre.

### 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 specifies when statements execute based on defined criteria. You can create conditional expressions that use <u>condition operators</u>, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Amazon FSx condition keys, see <u>Condition keys for Amazon FSx for Lustre</u> in the Service Authorization Reference. To learn with which actions and resources you can use a condition key, see Actions defined by Amazon FSx for Lustre.

To view examples of Amazon FSx identity-based policies, see <u>Identity-based policy examples for</u> Amazon FSx for Lustre.

## Access control lists (ACLs) in Amazon FSx

Supports ACLs: No

### 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 called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the <u>condition element</u> of a policy using the aws:ResourceTag/<u>key-name</u>, aws:RequestTag/<u>key-name</u>, 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 <u>Define permissions with ABAC authorization</u> in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see <u>Use attribute-based access control</u> (ABAC) in the *IAM User Guide*.

For more information about tagging Amazon FSx resources, see <u>Tag your Amazon FSx for Lustre</u> 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

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see <a href="Temporary security credentials in IAM">Temporary security credentials in IAM</a> and <a href="AWS services that work with IAM">AWS services that work with IAM</a> in the IAM User Guide.

#### Forward access sessions for Amazon FSx

### **Supports forward access sessions (FAS):** Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

FSx for Lustre Lustre User Guide

#### 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.



#### Marning

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 more information about creating and managing Amazon FSx service-linked roles, see Using service-linked roles for Amazon FSx.

### Identity-based policy examples for Amazon FSx for Lustre

By default, users and roles don't have permission to create or modify Amazon FSx resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

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

For details about actions and resource types defined by Amazon FSx, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Amazon FSx for Lustre in the Service Authorization Reference.

#### **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 <u>AWS managed policies</u> or <u>AWS</u>
  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 <a href="Policies and permissions in IAM">Policies and permissions in IAM</a> in the
  IAM User Guide.
- Use conditions in IAM policies to further restrict access You can add a condition to your
  policies to limit access to actions and resources. For example, you can write a policy condition to
  specify that all requests must be sent using SSL. You can also use conditions to grant access to
  service actions if they are used through a specific AWS service, such as CloudFormation. For more
  information, see <a href="IAM JSON policy elements: Condition">IAM User Guide</a>.
- 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 <u>Validate policies with IAM Access Analyzer</u> 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 <a href="Secure API">Secure API</a>
  access with MFA in the IAM User Guide.

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

### Using the Amazon FSx console

To access the Amazon FSx for Lustre 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 AmazonFSxConsoleReadOnlyAccess AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

You can see the AmazonFSxConsoleReadOnlyAccess and other Amazon FSx managed service policies in AWS managed policies for Amazon FSx for Lustre.

### 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.

```
{
            "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 Lustre

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 <a href="customer managed policies">customer managed policies</a> 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 <u>AWS managed policies</u> in the *IAM User Guide*.

### AmazonFSxServiceRolePolicy

Allows Amazon FSx to manage AWS resources on your behalf. See <u>Using service-linked roles for</u> Amazon FSx 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 <u>Using service-linked roles</u> for Amazon FSx.

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 <u>AmazonFSxDeleteServiceLinkedRoleAccess</u> in the AWS Managed Policy Reference Guide.

#### AWS managed policy: AmazonFSxFullAccess

You can attach AmazonFSxFullAccess to your IAM entities. Amazon FSx also attaches this policy to a service role that allows Amazon FSx to perform actions on your behalf.

Provides full access to Amazon FSx and access to related AWS services.

#### **Permissions details**

This policy includes the following permissions.

- fsx Allows principals full access to perform all Amazon FSx actions, except for BypassSnaplockEnterpriseRetention.
- ds Allows principals to view information about the Directory Service directories.
- ec2
  - Allows principals to create tags under the specified conditions.
  - To provide enhanced security group validation of all security groups that can be used with a VPC.
- iam Allows principles to create an Amazon FSx service linked role on the user's behalf. This is required so that Amazon FSx can manage AWS resources on the user's behalf.

• firehose – Allows principals to write records to a Amazon Data Firehose. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to Firehose.

logs – Allows principals to create log groups, log streams, and write events to log streams. This
is required so that users can monitor FSx for Windows File Server file system access by sending
audit access logs to CloudWatch Logs.

To view the permissions for this policy, see <u>AmazonFSxFullAccess</u> in the AWS Managed Policy Reference Guide.

### AWS managed policy: AmazonFSxConsoleFullAccess

You can attach the AmazonFSxConsoleFullAccess policy to your IAM identities.

This policy grants administrative permissions that allow full access to Amazon FSx and access to related AWS services via the AWS Management Console.

#### **Permissions details**

This policy includes the following permissions.

- fsx Allows principals to perform all actions in the Amazon FSx management console, except for BypassSnaplockEnterpriseRetention.
- cloudwatch Allows principals to view CloudWatch Alarms and metrics in the Amazon FSx management console.
- ds Allows principals to list information about an Directory Service directory.
- ec2
  - Allows principals to create tags on route tables, list network interfaces, route tables, security groups, subnets and the VPC associated with an Amazon FSx file system.
  - Allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.
  - Allows principals to view the elastic network interfaces associated with an Amazon FSx file system.
- kms Allows principals to list aliases for AWS Key Management Service keys.
- s3 Allows principals to list some or all of the objects in an Amazon S3 bucket (up to 1000).

• 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 <u>AmazonFSxConsoleFullAccess</u> in the AWS Managed Policy Reference Guide.

#### AWS managed policy: AmazonFSxConsoleReadOnlyAccess

You can attach the AmazonFSxConsoleReadOnlyAccess policy to your IAM identities.

This policy grants read-only permissions to Amazon FSx and related AWS services so that users can view information about these services in the AWS Management Console.

#### **Permissions details**

This policy includes the following permissions.

- fsx Allows principals to view information about Amazon FSx file systems, including all tags, in the Amazon FSx Management Console.
- cloudwatch Allows principals to view CloudWatch Alarms and metrics in the Amazon FSx Management Console.
- ds Allows principals to view information about an Directory Service directory in the Amazon FSx Management Console.
- ec2
  - Allows principals to view network interfaces, security groups, subnets and the VPC associated with an Amazon FSx file system in the Amazon FSx Management Console.
  - Allows principals to provide enhanced security group validation of all security groups that can be used with a VPC.
  - Allows principals to view the elastic network interfaces associated with an Amazon FSx file system.
- kms Allows principals to view aliases for AWS Key Management Service keys in the Amazon FSx Management Console.
- log Allows principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.

• 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 <u>AmazonFSxConsoleReadOnlyAccess</u> 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 <u>AmazonFSxReadOnlyAccess</u> in the AWS Managed Policy Reference Guide.

### Amazon FSx updates to AWS managed policies

View details about updates to AWS managed policies for Amazon FSx since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon FSx Document history page.

| Change   | Description  | Date          |
|--|--|---------------|
| AmazonFSxServiceRolePolicy  – Update to an existing policy | Amazon FSx added a new permission, ec2:Assig nIpv6Addresses that allows principals to assign IPv6 addresses to customer network interfaces that have an AmazonFSx.FileSyst emId tag. | July 22, 2025 |

| Change   | Description  | Date          |
|--|--|---------------|
| AmazonFSxServiceRolePolicy  – Update to an existing policy | Amazon FSx added a new permission, ec2:Unass ignIpv6Addresses that allows principals to unassign IPv6 addresses from customer network interfaces that have an AmazonFSx.FileSyst emId tag. | July 22, 2025 |
| AmazonFSxConsoleFullAccess  – Update to an existing policy | Amazon FSx added a new permission, fsx:Creat eAndAttachS3Access Point that allows principal s to create an S3 access point and attach it to an FSx volume.                                 | June 25, 2025 |
| AmazonFSxConsoleFullAccess  – Update to an existing policy | Amazon FSx added a new permission, fsx:Descr ibeS3AccessPointAt tachments that allows principals to list all S3 access points in an AWS account in an AWS Region.                          | June 25, 2025 |
| AmazonFSxConsoleFullAccess  – Update to an existing policy | Amazon FSx added a new permission, fsx:Detac hAndDeleteS3Access Point that allows principals to delete an S3 access point.   | June 25, 2025 |

| Change   | Description   | Date              |
|--|---|-------------------|
| AmazonFSxFullAccess – Update to an existing policy             | Amazon FSx added a new permission, fsx:Creat eAndAttachS3Access Point that allows principal s to create an S3 access point and attach it to an FSx volume.        | June 25, 2025     |
| AmazonFSxFullAccess – Update to an existing policy             | Amazon FSx added a new permission, fsx:Descr ibeS3AccessPointAt tachments that allows principals to list all S3 access points in an AWS account in an AWS Region. | June 25, 2025     |
| AmazonFSxFullAccess – Update to an existing policy             | Amazon FSx added a new permission, fsx:Detac hAndDeleteS3Access Point that allows principals to delete an S3 access point.  | June 25, 2025     |
| AmazonFSxConsoleRe adOnlyAccess – Update to an existing policy | Amazon FSx added new permission, ec2:Descr ibeNetworkInterfac es that allows principals to view the elastic network interfaces associated with their file system. | February 25, 2025 |

| Change   | Description  | Date              |
|--|--|-------------------|
| AmazonFSxConsoleFullAccess  – Update to an existing policy     | Amazon FSx added new permission, ec2:Descr ibeNetworkInterfac es that allows principals to view the elastic network interfaces associated with their file system.                      | February 07, 2025 |
| AmazonFSxServiceRolePolicy  – Update to an existing policy     | Amazon FSx added new permission, ec2: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: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   |
| AmazonFSxConsoleFullAccess  – 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   |
| AmazonFSxFullAccess – Update to an existing policy         | Amazon FSx added new permission to enable users to perform cross-region and cross-account data replicati on 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 replicati on 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    |

FSx for Lustre User Guide

| Change  | Description  | Date               |
|---|--|--------------------|
| AWS managed policy:  AmazonFSxServiceRolePolicy  – Update to an existing policy | Amazon FSx modified the existing cloudwatc h:PutMetricData permission so that Amazon FSx publishes CloudWatc h 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      |
| AmazonFSxConsoleRe adOnlyAccess – Update to an existing policy                  | Amazon FSx added new permissions to enable users to view enhanced performan ce 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 performan ce metrics and recommended actions for FSx for Windows File Server file systems in the Amazon FSx console. | September 21, 2022 |

FSx for Lustre User Guide

| Change  | Description  | Date              |
|---|--|-------------------|
| AmazonFSxReadOnlyAccess – Started tracking policy                 | This policy grants read-<br>only access to all Amazon<br>FSx resources and any tags<br>associated with them.                             | February 4, 2022  |
| AmazonFSxDeleteSer viceLinkedRoleAccess – Started tracking policy | This policy grants administr<br>ative permissions that allow<br>Amazon FSx to delete its<br>Service Linked Role for<br>Amazon S3 access. | January 7, 2022   |
| AmazonFSxServiceRolePolicy  – Update to an existing policy        | Amazon FSx added new permissions to allow Amazon FSx to manage network configurations for Amazon FSx for NetApp ONTAP file systems.      | September 2, 2021 |
| AmazonFSxFullAccess – Update to an existing policy                | Amazon FSx added new permissions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.                           | September 2, 2021 |
| AmazonFSxConsoleFullAccess  – Update to an existing policy        | Amazon FSx added new permissions to allow Amazon FSx to create Amazon FSx for NetApp ONTAP Multi-AZ file systems.                        | September 2, 2021 |
| AmazonFSxConsoleFullAccess  – Update to an existing policy        | Amazon FSx added new permissions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.                           | September 2, 2021 |

| Change   | Description   | Date         |
|--|---|--------------|
| AmazonFSxServiceRolePolicy  – Update to an existing policy | Amazon FSx added new permissions to allow Amazon FSx to describe and write to CloudWatch Logs log streams.  This is required so that users can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.  | June 8, 2021 |
| AmazonFSxServiceRolePolicy  – Update to an existing policy | Amazon FSx added new permissions to allow Amazon FSx to describe and write to Amazon Data Firehose delivery streams.  This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Data Firehose.                                 | June 8, 2021 |
| AmazonFSxFullAccess – Update to an existing policy         | Amazon FSx added new permissions to allow principal s to describe and create CloudWatch Logs log groups, log streams, and write events to log streams.  This is required so that principals can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs. | June 8, 2021 |

| Change   | Description   | Date         |
|--|---|--------------|
| AmazonFSxFullAccess – Update to an existing policy         | Amazon FSx added new permissions to allow principal s to describe and write records to a Amazon Data Firehose.  This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Data Firehose.   | June 8, 2021 |
| AmazonFSxConsoleFullAccess  - Update to an existing policy | Amazon FSx added new permissions to allow principal s to describe the Amazon CloudWatch Logs log groups associated with the account making the request.  This is required so that principals can choose an existing CloudWatch Logs log group when configuring file access auditing for an FSx for Windows File Server file system. | June 8, 2021 |

| Change   | Description  | Date         |
|--|--|--------------|
| AmazonFSxConsoleFullAccess  – Update to an existing policy     | Amazon FSx added new permissions to allow principal s 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 |
| AmazonFSxConsoleRe adOnlyAccess – Update to an existing policy | Amazon FSx added new permissions to allow principal s to describe the Amazon CloudWatch Logs log groups associated with the account making the request.  This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.                                  | June 8, 2021 |

| Change   | Description   | Date         |
|--|---|--------------|
| AmazonFSxConsoleRe adOnlyAccess – Update to an existing policy | Amazon FSx added new permissions to allow principal s 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 Lustre identity and access

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.

Troubleshooting 320

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

Troubleshooting 321

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 <a href="How Amazon FSx for Lustre works">How Amazon FSx for Lustre works</a> 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 <a href="Providing access to AWS accounts owned by third parties in the IAM User Guide">IAM User Guide</a>.
- To learn how to provide access through identity federation, see <u>Providing access to externally</u> authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the IAM User Guide.

## **Using tags with Amazon FSx**

You can use tags to control access to Amazon FSx resources and to implement attribute-based access control (ABAC). To apply tags to Amazon FSx resources during creation, users must have certain AWS Identity and Access Management (IAM) permissions.

## Grant permission to tag resources during creation

With some resource-creating Amazon FSx for Lustre API actions, you can specify tags when you create the resource. You can use these resource tags to implement attribute-based access control (ABAC). For more information, see What is ABAC for AWS? in the IAM User Guide.

For users to tag resources on creation, they must have permission to use the action that creates the resource, such as fsx:CreateFileSystem. If tags are specified in the resource-creating action, IAM 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 policy allows users to create file systems and apply tags to them during creation in a specific AWS account.

{

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "fsx:CreateFileSystem",
            "fsx:TagResource"
        ],
        "Resource": [
            "arn:aws:fsx:region:account-id:file-system/*"
        ]
    }
}
```

Similarly, the following policy allows users to create backups on a specific file system and apply any tags to the backup during backup creation.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateBackup"
      "Resource": "arn:aws:fsx:region:account-id:file-system/file-system-id*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "fsx:TagResource"
      "Resource": "arn:aws:fsx:region:account-id:backup/*"
    }
  ]
}
```

The fsx:TagResource action is evaluated only if tags are applied during the resource-creating action. Therefore, a user who has permissions to create a resource (assuming there are no tagging conditions) does not require permission 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 <u>Tag your Amazon FSx for Lustre</u> <u>resources</u>. For more information about using tags to control access to Amazon FSx for Lustre resources, see <u>Using tags to control access to your Amazon FSx resources</u>.

#### Using tags to control access to your Amazon FSx resources

To control access to Amazon FSx resources and actions, you can use IAM policies based on tags. You can provide this control in two ways:

- You can control access to Amazon FSx resources based on the tags on those resources.
- You can control which tags can be passed in an IAM request condition.

For information about how to use tags to control access to AWS resources, see <u>Controlling access</u> <u>using tags</u> in the *IAM User Guide*. For more information about tagging Amazon FSx resources at creation, see <u>Grant permission to tag resources during creation</u>. For more information about tagging resources, see Tag your Amazon FSx for Lustre resources.

#### Controlling access based on tags on a resource

To control which actions a user or role can perform on an Amazon FSx resource, you can use tags on the resource. For example, you might want to allow or deny specific API operations on a file system resource based on the key-value pair of the tag on the resource.

#### Example Example policy - Create a file system on when providing a specific tag

This policy allows the user to create a file system only when they tag it with a specific tag key value pair, in this example, key=Department, value=Finance.

```
{
    "Effect": "Allow",
    "Action": [
        "fsx:CreateFileSystem",
        "fsx:TagResource"
],
    "Resource": "arn:aws:fsx:region:account-id:file-system/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Department": "Finance"
        }
    }
}
```

}

#### Example Example policy – Create backups only on file systems with a specific tag

This policy allows users to create backups only on file systems that are tagged with the key value pair key=Department, value=Finance, and the backup will be created with the tag Department=Finance.

**JSON** 

```
}
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                     "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource",
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                     "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

## Example Example policy – Create a file system with a specific tag from backups with a specific tag

This policy allows users to create file systems that are tagged with Department=Finance only from backups that are tagged with Department=Finance.

**JSON** 

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        },
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        }
    ]
}
```

#### Example Example policy – Delete file systems with specific tags

This policy allows a user to delete only file systems that are tagged with Department=Finance. If they create a final backup, then it must be tagged with Department=Finance. For FSx for Lustre file systems, users need the fsx:CreateBackup privilege to create the final backup.

**JSON** 

```
}
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:DeleteFileSystem"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                     "aws:ResourceTag/Department": "Finance"
                }
            }
        },
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:backup/*",
            "Condition": {
                "StringEquals": {
                     "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

#### Example Example policy - Create data repository tasks on file systems with specific tag

This policy allows users to create data repository tasks tagged with Department=Finance, and only on file systems tagged with Department=Finance.

**JSON** 

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateDataRepositoryTask"
            "Resource": "arn:aws:fsx:us-east-1:111122223333:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateDataRepositoryTask",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:us-east-1:111122223333:task/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

## Using service-linked roles for Amazon FSx

Amazon FSx uses AWS Identity and Access Management (IAM) <u>service-linked roles</u>. 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 <u>AWS services that work</u> with <u>IAM</u> and look for the services that have **Yes** in the **Service-linked roles** 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 two service-linked roles named AWSServiceRoleForAmazonFSx and AWSServiceRoleForFSxS3Access\_fs-01234567890 that perform certain actions in your account. Examples of these actions are creating elastic network interfaces for your file systems in your VPC and accessing your data repository in an Amazon S3 bucket. For AWSServiceRoleForFSxS3Access\_fs-01234567890, this service-linked role is created for each Amazon FSx for Lustre file system you create that is linked to an S3 bucket.

#### AWSServiceRoleForAmazonFSx permissions details

For AWSServiceRoleForAmazonFSx, the role permissions policy allows Amazon FSx to complete the following administrative actions on the user's behalf on all applicable AWS resources:

For updates to this policy, see AmazonFSxServiceRolePolicy.



#### Note

The AWSServiceRoleForAmazonFSx is used by all Amazon FSx file system types; some of the listed permissions are not applicable to FSx for Lustre.

 ds – Allows Amazon FSx to view, authorize, and unauthorize applications in your 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.
  - Assign IPv6 addresses to customer network interfaces that have an AmazonFSx.FileSystemId tag.
  - Unassign IPv6 addresses from customer network interfaces that have an AmazonFSx.FileSystemId tag.
  - To provide enhanced security group validation of all security groups that can be used with a VPC.
  - Create a permission for an AWS-authorized user to perform certain operations on a network interface.
- cloudwatch Allows 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.

#### **JSON**

```
"Version": "2012-10-17",
```

```
"Statement": [
    {
        "Sid": "CreateFileSystem",
        "Effect": "Allow",
        "Action": [
            "ds:AuthorizeApplication",
            "ds:GetAuthorizedApplicationDetails",
            "ds:UnauthorizeApplication",
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DescribeAddresses",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVPCs",
            "ec2:DisassociateAddress",
            "ec2:GetSecurityGroupsForVpc",
            "route53:AssociateVPCWithHostedZone"
        ],
        "Resource": "*"
    },
        "Sid": "PutMetrics",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:PutMetricData"
        ],
        "Resource": [
            II * II
        ],
        "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 <a href="Service-Linked Role Permissions">Service-Linked Role Permissions</a> in the IAM User Guide.

#### AWSServiceRoleForFSxS3Access permissions details

For AWSServiceRoleForFSxS3Access\_*file-system-id*, the role permissions policy allows Amazon FSx to complete the following actions on an Amazon S3 bucket hosting the data repository for an Amazon FSx for Lustre file system.

- s3:AbortMultipartUpload
- s3:DeleteObject

- s3:Get\*
- s3:List\*
- s3:PutBucketNotification
- s3:PutObject

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 AWS CLI, or the AWS 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 servicelinked role for you again.

#### Editing a service-linked role for Amazon FSx

Amazon FSx does not allow you to edit these service-linked roles. 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.

## File system access control with Amazon VPC

An Amazon FSx file system is accessible through an elastic network interface that resides in the virtual private cloud (VPC) based on the Amazon VPC service that you associate with your file system. You access your Amazon FSx file system through its DNS name, which maps to the file system's network interface. Only resources within the associated VPC, or a peered VPC, can access your file system's network interface. For more information, see What is Amazon VPC? in the Amazon VPC User Guide.



#### Marning

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

## **Amazon VPC Security Groups**

To further control network traffic going through your file system's network interface within your VPC, you use security groups to limit access to your file systems. A security group acts as a virtual

firewall to control the traffic for its associated resources. In this case, the associated resource is your file system's network interface. You also use VPC security groups to control network traffic for your Lustre clients.

#### **EFA-enabled security groups**

If you are going to create an EFA-enabled FSx for Lustre, you should first create an EFA-enabled security group and specify it as the security group for the file system. An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself and the security group of the clients if clients reside in a different security group. For more information, see Step 1: Prepare an EFA-enabled security group in the *Amazon EC2 User Guide*.

#### **Controlling Access Using Inbound and Outbound Rules**

To use a security group to control access to your Amazon FSx file system and Lustre clients, you add the inbound rules to control incoming traffic and outbound rules to control the outgoing traffic from your file system and Lustre clients. Make sure to 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 <u>Security Group Rules</u> in the *Amazon EC2 User Guide*.

#### To create a security group for your Amazon FSx file system

- 1. Open the Amazon EC2 console at <a href="https://eusc-de-east-1.console.amazonaws-eusc.eu/ec2">https://eusc-de-east-1.console.amazonaws-eusc.eu/ec2</a>.
- 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 VPC associated with your Amazon FSx file system to create the security group within that VPC.
- 6. Choose **Create** to create the security group.

Next, you add inbound rules to the security group that you just created to enable Lustre traffic between your FSx for Lustre file servers.

#### To add inbound rules to your security group

1. Select the security group you just created if it's not already selected. For **Actions**, choose **Edit inbound rules**.

2. Add the following inbound rules.

| Туре               | Protocol | Port Range | Source   | Description  |
|--------------------|----------|------------|--|--|
| Custom TCP<br>rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group ID of the security group that you just created                 | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers                       |
| Custom TCP<br>rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security groups associated with your Lustre clients | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |
| Custom TCP<br>rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group ID of the security group that you just created                 | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers                       |
| Custom TCP<br>rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the   | Allows Lustre<br>traffic between<br>FSx for Lustre                                       |

| Туре | Protocol | Port Range | Source   | Description                        |
|------|----------|------------|--|------------------------------------|
|      |          |            | security groups<br>associated with<br>your Lustre<br>clients | file servers and<br>Lustre clients |

3. Choose **Save** to save and apply the new inbound rules.

By default, security group rules allow all outbound traffic (All, 0.0.0.0/0). If your security group doesn't allow all outbound traffic, add the following outbound rules to your security group. These rules allow traffic between FSx for Lustre file servers and Lustre clients, and between Lustre file servers.

#### To add outbound rules to your security group

- 1. Choose the same security group to which you just added the inbound rules. For **Actions**, choose **Edit outbound rules**.
- 2. Add the following outbound rules.

| Туре               | Protocol | Port Range | Source   | Description   |
|--------------------|----------|------------|--|---|
| Custom TCP<br>rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group ID of the security group that you just created | Allow Lustre<br>traffic between<br>FSx for Lustre<br>file servers                       |
| Custom TCP<br>rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security group associated with      | Allow Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |

| Туре               | Protocol | Port Range | Source   | Description  |
|--------------------|----------|------------|--|--|
|                    |          |            | your Lustre<br>clients   |  |
| Custom TCP<br>rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group ID of the security group that you just created                 | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers                       |
| Custom TCP<br>rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the security groups associated with your Lustre clients | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |

3. Choose **Save** to save and apply the new outbound rules.

#### To associate a security group with your Amazon FSx file system

- 1. Open the Amazon FSx console at https://eusc-de-east-1.console.amazonaws-eusc.eu/fsx/.
- 2. On the console dashboard, choose your file system to view its details.
- 3. On the **Network & Security** tab, click on the **Amazon EC2 console** link under **Network Interface(s)** to view all network interfaces for your file system.
- 4. For each network interface, choose **Actions**, then choose **Change security groups**.
- 5. In the **Change security groups** dialog box, choose the security groups that you want to associate with the network interface.
- 6. Choose **Save**.

## Lustre client VPC security group rules

You use VPC security groups to control access to your Lustre clients by adding inbound rules to control incoming traffic and outbound rules to control the outgoing traffic from your Lustre clients. Make sure to have the right network traffic rules in your security group to ensure that Lustre traffic can flow between your Lustre clients and your Amazon FSx file systems.

Add the following inbound rules to the security groups applied to your Lustre clients.

| Туре            | Protocol | Port Range | Source   | Description  |
|-----------------|----------|------------|--|--|
| Custom TCP rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security groups that are applied to your Lustre clients           | Allows Lustre<br>traffic between<br>Lustre clients                                       |
| Custom TCP rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security groups associate d with your FSx for Lustre file systems | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |
| Custom TCP rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the security groups that are applied                                  | Allows Lustre<br>traffic between<br>Lustre clients                                       |

| Туре            | Protocol | Port Range | Source   | Description  |
|-----------------|----------|------------|--|--|
|                 |          |            | to your Lustre<br>clients  |  |
| Custom TCP rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the security groups associate d with your FSx for Lustre file systems | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |

Add the following outbound rules to the security groups applied to your Lustre clients.

| Туре            | Protocol | Port Range | Source   | Description   |
|-----------------|----------|------------|--|---|
| Custom TCP rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security groups that are applied to your Lustre clients           | Allows Lustre<br>traffic between<br>Lustre clients                                      |
| Custom TCP rule | TCP      | 988        | Choose <b>Custom</b> and enter the security group IDs of the security groups associate d with your FSx for Lustre file systems | Allow Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |

| Туре            | Protocol | Port Range | Source   | Description  |
|-----------------|----------|------------|--|--|
| Custom TCP rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the security groups that are applied to your Lustre clients           | Allows Lustre<br>traffic between<br>Lustre clients                                       |
| Custom TCP rule | TCP      | 1018-1023  | Choose <b>Custom</b> and enter the security group IDs of the security groups associate d with your FSx for Lustre file systems | Allows Lustre<br>traffic between<br>FSx for Lustre<br>file servers and<br>Lustre clients |

#### **Amazon VPC network ACLs**

Another option for securing access to the file system within your VPC is to establish network access control lists (network ACLs). Network ACLs are separate from security groups, but have similar functionality to add an additional layer of security to the resources in your VPC. For more information on implementing access control using network ACLs, see <a href="Control traffic to subnets">Control traffic to subnets</a> using Network ACLs in the Amazon VPC User Guide.

## **Compliance Validation for Amazon FSx for Lustre**

To learn whether an AWS service is within the scope of specific compliance programs, see and choose the compliance program that you are interested in. For general information, see .

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more

Amazon VPC network ACLs 342

information about your compliance responsibility when using AWS services, see <u>AWS Security</u> Documentation.

# Amazon FSx for Lustre and interface VPC endpoints (AWS PrivateLink)

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 <u>AWS PrivateLink</u>, a technology that enables you to privately access Amazon FSx APIs without an internet gateway, NAT device, VPN connection, or Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon FSx APIs. Traffic between your VPC and Amazon FSx does not leave the AWS network.

Each interface VPC endpoint is represented by one or more elastic network interfaces in your subnets. A network interface provides a private IP address that serves as an entry point for traffic to the Amazon FSx API.

## **Considerations for Amazon FSx interface VPC endpoints**

Before you set up an interface VPC endpoint for Amazon FSx, be sure to review <u>Interface VPC</u> 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 Lustre file system by calling the CreateFileSystem API from within your VPC. For the full list of Amazon FSx APIs, see <u>Actions</u> 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.

Interface VPC endpoints 343

## 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 <u>Creating an interface VPC endpoint</u> in the *Amazon VPC User Guide*.

For a complete list of Amazon FSx endpoints, see <u>Amazon FSx endpoints and quotas</u> in the *Amazon Web Services General Reference*.

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 <u>Viewing and updating DNS support for your VPC</u> 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 <u>Accessing a service through an interface VPC endpoint</u> in the *Amazon VPC User Guide*.

## Creating a VPC endpoint policy for Amazon FSx

To further control access to the Amazon FSx API, you can optionally attach an AWS Identity and Access Management (IAM) policy to your VPC endpoint. The policy specifies the following:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see <u>Controlling access to services with VPC endpoints</u> in the *Amazon VPC User Guide*.

## **Service quotas for Amazon FSx for Lustre**

Following, you can find out about quotas when working with Amazon FSx for Lustre.

#### **Topics**

- Quotas that you can increase
- Resource quotas for each file system
- Additional considerations

## **Quotas that you can increase**

Following are the quotas for Amazon FSx for Lustre per AWS account, per AWS Region, which you can increase.

| Resource   | Default | Description   |
|--|---------|---|
| Lustre Persistent 1 file systems                         | 100     | The maximum number of Amazon FSx for Lustre Persistent 1 file systems that you can create in this account.                      |
| Lustre Persistent 2 file systems                         | 100     | The maximum number of Amazon FSx for Lustre Persistent 2 file systems that you can create in this account.                      |
| Lustre Persistent HDD storage capacity (per file system) | 102000  | The maximum amount of HDD storage capacity (in GiB) that you can configure for an Amazon FSx for Lustre persistent file system. |
| Lustre Persistent 1 file storage capacity                | 100800  | The maximum amount of storage capacity (in GiB) that you can configure for all Amazon FSx for Lustre                            |

Quotas that you can increase 346

| Resource   | Default | Description   |
|--|---------|---|
|  |         | Persistent 1 file systems in this account.  |
| Lustre Persistent 2 file storage capacity                                    | 100800  | The maximum amount of storage capacity (in GiB) that you can configure for all Amazon FSx for Lustre Persistent 2 file systems in this account.                                   |
| Lustre Scratch file systems  | 100     | The maximum number of<br>Amazon FSx for Lustre scratch<br>file systems that you can<br>create in this account.  |
| Lustre Scratch storage capacity  | 100800  | The maximum amount of storage capacity (in GiB) that you can configure for all Amazon FSx for Lustre scratch file systems in this account.  |
| Lustre Persistent Intelligent-<br>Tiering throughput capacity                | 100000  | The total amount of throughput capacity (in MBps) allowed for all Amazon FSx for Lustre Intelligent-Tiering file systems in this account.   |
| Lustre Persistent Intellige<br>nt-Tiering SSD read cache<br>storage capacity | 100800  | The maximum amount of provisioned SSD read cache storage capacity (in GiB) that you can configure for all Amazon FSx for Lustre Intelligent-Tiering file systems in this account. |

| Resource       | Default | Description  |
|----------------|---------|--|
| Lustre backups | 500     | The maximum number of user-initiated backups that you can have for all Amazon FSx for Lustre file systems in this account. |

#### To request a quota increase

- 1. Open the Service Quotas console.
- 2. In the navigation pane, choose **AWS services**.
- 3. Choose **Lustre**.
- 4. Choose a quota.
- 5. Choose **Request quota increase**, and follow the directions to request a quota increase.
- 6. To view the status of the quota request, choose **Quota request history** in the console navigation pane.

For more information, see Requesting a quota increase in the Service Quotas User Guide.

## Resource quotas for each file system

Following are the limits on Amazon FSx for Lustre resources for each file system in an AWS Region.

| Resource   | Limit per file system |
|--|-----------------------|
| Maximum number of tags   | 50                    |
| Maximum retention period for automated backups   | 90 days               |
| Maximum number of backup copy requests in progress to a single destination Region per account. | 5                     |
| Number of file updates from linked S3 bucket per file systems                                  | 10 million / month    |

| Resource                                    | Limit per file system |
|---|-----------------------|
| Minimum storage capacity, SSD file systems  | 1.2 TiB               |
| Minimum storage capacity, HDD file systems  | 6 TiB                 |
| Minimum throughput per unit of storage, SSD | 50 MBps               |
| Maximum throughput per unit of storage, SSD | 1000 MBps             |
| Minimum throughput per unit of storage, HDD | 12 MBps               |
| Maximum throughput per unit of storage, HDD | 40 MBps               |

## **Additional considerations**

In addition, note the following:

- You can use each AWS Key Management Service (AWS KMS) key on up to 125 Amazon FSx for Lustre file systems.
- For a list of AWS Regions where you can create file systems, see <u>Amazon FSx Endpoints and</u> Quotas in the *AWS General Reference*.

Additional considerations 349

## **Troubleshooting Amazon FSx for Lustre**

This section covers various troubleshooting scenarios and solutions for Amazon FSx for Lustre file systems.

If you encounter problems not listed following, try asking a question in the <u>Amazon FSx for Lustre</u> forum.

#### **Topics**

- Creating an FSx for Lustre file system fails
- Troubleshooting file system mount issues
- You cannot access your file system
- Unable to validate access to an S3 bucket when creating a DRA
- Renaming directories takes a long time
- Troubleshooting a misconfigured linked S3 bucket
- Troubleshooting storage issues
- Troubleshooting FSx for Lustre CSI driver issues

## Creating an FSx for Lustre file system fails

There are a number of potential causes when a file system creation request fails, as described in the following topics.

# Cannot create an EFA-enabled file system because of misconfigured security group

Creating an FSx for Lustre EFA-enabled file system fails with the following error message:

Insufficient security group permissions to create an EFA-enabled file system. Update security group to allow all internal inbound and outbound traffic.

#### Action to take

Make sure that the VPC security group you are using for the creation operation is configured as described in EFA-enabled security groups. An EFA requires a security group that allows all inbound

Creating a file system fails 350

and outbound traffic to and from the security group itself and the security group of the clients if clients reside in a different security group.

### Cannot create a file system because of misconfigured security group

Creating an FSx for Lustre 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 Lustre LNET network traffic on port 988

#### Action to take

Make sure that the VPC security group you are using for the creation operation is configured as described in <u>File system access control with Amazon VPC</u>. You must set up the security group to allow inbound traffic on ports 988 and 1018-1023 from the security group itself or the full subnet CIDR, which is required to allow the file system hosts to communicate with each other.

# Cannot create a file system because of insufficient capacity errors

You may receive an insufficient capacity error when attempting to create a new file system, update storage capacity, or modify throughput capacity.

#### Cause

This error occurs when FSx for Lustre does not currently have enough available hardware capacity in the requested Availability Zone to fulfill your request.

#### Solution

To resolve the issue, try the following:

- Wait a few minutes and retry your request, as capacity availability changes frequently.
- Try your request in a different Availability Zone.
- Attempt the operation with a smaller storage size or lower throughput level

## Cannot create a file system that is linked to an S3 bucket

If creating a new file system that is linked to an S3 bucket fails with an error message similar to the following.

```
User: arn:aws:iam::012345678901:user/username is not authorized to perform: iam:PutRolePolicy on resource: resource ARN
```

This error can happen if you try to create a file system linked to an Amazon S3 bucket without the necessary IAM permissions. The required IAM permissions support the Amazon FSx for Lustre service-linked role that is used to access the specified Amazon S3 bucket on your behalf.

#### Action to take

Ensure that your IAM entity (user, group, or role) has the appropriate permissions to create file systems. Doing this includes adding the permissions policy that supports the Amazon FSx for Lustre service-linked role. For more information, see <u>Adding permissions to use data repositories in Amazon S3</u>.

For more information about service-linked roles, see <u>Using service-linked roles for Amazon FSx</u>.

# Troubleshooting file system mount issues

There are a number of potential causes when a file system mount command fails, as described in the following topics.

### File system mount fails right away

The file system mount command fails right away. The following code shows an example.

```
mount.lustre: mount fs-0123456789abcdef0.fsx.us-east-1.aws@tcp:/fsx at /lustre
failed: No such file or directory

Is the MGS specification correct?
Is the filesystem name correct?
```

This error can occur if you aren't using the correct mountname value when mounting a persistent or scratch 2 file system by using the **mount** command. You can get the mountname value from the response of the **describe-file-systems** AWS CLI command or the **DescribeFileSystems** API operation.

# File system mount hangs and then fails with timeout error

The file system mount command hangs for a minute or two, and then fails with a timeout error.

File system mount fails 352

The following code shows an example.

```
sudo mount -t lustre file_system_dns_name@tcp:/mountname /mnt/fsx

[2+ minute wait here]
Connection timed out
```

This error can occur because the security groups for the Amazon EC2 instance or the file system aren't configured properly.

#### Action to take

Make sure that your security groups for the file system have the inbound rules specified in <u>Amazon VPC Security Groups</u>.

## Automatic mounting fails and the instance is unresponsive

In some cases, automatic mounting might fail for a file system and your Amazon EC2 instance might stop responding.

This issue can occur if the \_netdev option wasn't declared. If \_netdev is missing, your Amazon EC2 instance can stop responding. This result is because network file systems need to be initialized after the compute instance starts its networking.

#### Action to take

If this issue occurs, contact AWS Support.

# File system mount fails during system boot

The file system mount fails during the system boot. The mounting is automated using /etc/fstab. When the file system is not mounted, the following error is seen in the syslog for the instance booting time frame.

```
LNetError: 3135:0:(lib-socket.c:583:lnet_sock_listen()) Can't create socket: port 988 already in use
LNetError: 122-1: Can't start acceptor on port 988: port already in use
```

This error can occur when port 988 is not available. When the instance is configured to mount NFS file systems, it is possible that the NFS mounts will bind its client port to port 988

#### Action to take

You can work around this problem by tuning the NFS client's noresvport and noauto mount options where possible.

## File system mount using DNS name fails

Misconfigured Domain Name Service (DNS) names can cause file system mount failures, as shown in the following scenarios.

**Scenario 1:** A file system mount that is using a Domain Name Service (DNS) name fails. The following code shows an example.

```
sudo mount -t lustre file_system_dns_name@tcp:/mountname /mnt/fsx
mount.lustre: Can't parse NID
'file_system_dns_name@tcp:/mountname'
```

#### Action to take

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 <u>Using DNS with Your VPC</u> in the *Amazon VPC User Guide*.

To specify a DNS name in the mount command, do the following:

- Ensure that the Amazon EC2 instance is in the same VPC as your Amazon FSx for Lustre file system.
- Connect your Amazon EC2 instance inside a VPC configured to use the DNS server provided by Amazon. For more information, see DHCP Options Sets in the Amazon VPC User Guide.
- Ensure that the Amazon VPC of the connecting Amazon EC2 instance has DNS host names enabled. For more information, see <u>Updating DNS Support for Your VPC</u> in the *Amazon VPC User Guide*.

**Scenario 2:** A file system mount that is using a Domain Name Service (DNS) name fails. The following code shows an example.

```
mount -t lustre file_system_dns_name@tcp:/mountname /mnt/fsx
mount.lustre: mount file_system_dns_name@tcp:/mountname at /mnt/fsx failed: Input/
output error Is the MGS running?
```

#### Action to take

Make sure that the client's VPC security groups have the correct outbound traffic rules applied. This recommendation holds true especially if you aren't using the default security group, or if you have modified the default security group. For more information, see Amazon VPC Security Groups.

## You cannot access your file system

There are a number of potential causes for being unable to access your file system, each with their own resolution, as follows.

# The Elastic IP address attached to the file system elastic network interface was deleted

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

## The file system elastic network interface was modified or deleted

You must not modify or delete the file system's elastic network interface. Modifying or deleting the network interface can cause a permanent loss of connection between your VPC and your file system. Create a new file system, and do not modify or delete the FSx elastic network interface. For more information, see File system access control with Amazon VPC.

# Unable to validate access to an S3 bucket when creating a DRA

Creating a data repository association (DRA) from the Amazon FSx console or using the createdata-repository-association CLI command (<u>CreateDataRepositoryAssociation</u> is the equivalent API action) fails with the following error message.

Amazon FSx is unable to validate access to the S3 bucket. Ensure the IAM role or user you are using has s3:Get\*, s3:List\* and s3:PutObject permissions to the S3 bucket prefix.

FSx for Lustre Lustre User Guide



#### Note

You can also get the above error when creating a Scratch 1, Scratch 2, or Persistent 1 file system that is linked to a data repository (S3 bucket or prefix) using the Amazon FSx console or the create-file-system CLI command (CreateFileSystem is the equivalent API action).

#### Action to take

If the FSx for Lustre file system is in the same account as the S3 bucket, this error means the IAM role you used for the create request doesn't have the necessary permissions to access the S3 bucket. Make sure the IAM role has the permissions listed in the error message. These permissions support the Amazon FSx for Lustre service-linked role that is used to access the specified Amazon S3 bucket on your behalf.

If the FSx for Lustre file system is in a different account as the S3 bucket (cross-account case), in additional to making sure the IAM role you used has the required permissions, the S3 bucket policy should be configured to allow the access from the account that the FSx for Lustre is created in.

For more information about S3 cross-account bucket permissions, see Example 2: Bucket owner granting cross-account bucket permissions in the Amazon Simple Storage Service User Guide.

# Renaming directories takes a long time

#### Question

I renamed a directory on a file system linked to an Amazon S3 bucket and have automatic export enabled. Why are the files inside this directory taking a long time to get renamed on the S3 bucket?

#### **Answer**

When you rename a directory on the file system, FSx for Lustre creates new S3 objects for all of the files and directories inside the directory that was renamed. The amount of time it takes to propagate the directory rename to S3 is directly correlated to the quantity of files and directories that are descendants of the directory being renamed.

# Troubleshooting a misconfigured linked S3 bucket

In some cases, an FSx for Lustre file system's linked S3 bucket might have a misconfigured data repository lifecycle state.

#### Possible cause

This error can occur if Amazon FSx does not have the necessary AWS Identity and Access Management (IAM) permissions that are required to access the linked data repository. The required IAM permissions support the Amazon FSx for Lustre service-linked role that is used to access the specified Amazon S3 bucket on your behalf.

#### Action to take

- 1. Ensure that your IAM entity (user, group, or role) has the appropriate permissions to create file systems. Doing this includes adding the permissions policy that supports the Amazon FSx for Lustre service-linked role. For more information, see <a href="Adding permissions to use data repositories in Amazon S3">Adding permissions to use data repositories in Amazon S3</a>.
- Using the Amazon FSx CLI or API, refresh the file system's AutoImportPolicy with the update-file-system CLI command (<u>UpdateFileSystem</u> is the equivalent API action), as follows.

```
aws fsx update-file-system \
--file-system-id fs-0123456789abcdef0 \
--lustre-configuration AutoImportPolicy=the_existing_AutoImportPolicy
```

For more information about service-linked roles, see Using service-linked roles for Amazon FSx.

#### **Possible Cause**

This error can occur if the linked Amazon S3 data repository has an existing event notification configuration with event types that overlap with the Amazon FSx event notification configuration (s3:ObjectCreated:\*, s3:ObjectRemoved:\*).

This can also occur if the Amazon FSx event notification configuration on the linked S3 bucket was deleted or modified.

#### Action to take

Remove any existing event notification on the linked S3 bucket that uses either or both
of the event types that the FSx event configuration uses, s3:0bjectCreated:\* and
s3:0bjectRemoved:\*.

- 2. Ensure that there is an S3 Event Notification Configuration in your linked S3 bucket with the name FSx, event types s3:0bjectCreated:\* and s3:0bjectRemoved:\*, and send to the SNS topic with ARN: topic\_arn\_returned\_in\_API\_response.
- 3. Reapply the FSx event notification configuration on the S3 bucket by using the Amazon FSx CLI or API, to refresh the file system's AutoImportPolicy. Do so with the update-file-system CLI command (UpdateFileSystem is the equivalent API action), as follows.

```
aws fsx update-file-system \
--file-system-id fs-0123456789abcdef0 \
--lustre-configuration AutoImportPolicy=the_existing_AutoImportPolicy
```

# **Troubleshooting storage issues**

In some cases, you may experience storage issues with your file system. You can troubleshoot these issues by using 1fs commands, such as the 1fs migrate command.

## Write error due to no space on storage target

You can check the storage usage of your file system by using the lfs df -h command, as described in <u>File system storage layout</u>. The filesystem\_summary field reports the total file system storage usage.

If the file system disk usage is 100%, consider increasing the storage capacity of your file system. For more information, see <u>Managing storage capacity</u>.

If the file system storage usage is not 100% and you still get write errors, the file you are writing to may be striped on an OST that is full.

#### Action to take

- If many of your OSTs are full, increase the storage capacity of your file system. Check for unbalanced storage on OSTs by following the actions of the <u>Unbalanced storage on OSTs</u> section.
- If your OSTs are not full, tune the client dirty page buffer size by applying the following tuning to all your client instances:

Storage issues 358

```
sudo lctl set_param osc.*.max_dirty_mb=64
```

## **Unbalanced storage on OSTs**

Amazon FSx for Lustre distributes new file stripes evenly across OSTs. However, your file system may still become unbalanced due to I/O patterns or file storage layout. As a result, some storage targets can become full while others remain relatively empty.

You use the lfs migrate command to move files or directories from more-full to less-full OSTs. You can use the lfs migrate command in either block or non-block mode.

- Block mode is the default mode for the lfs migrate command. When run in block mode, lfs migrate first acquires a group lock on the files and directories before data migration to prevent modifications to the files, then releases the lock when migration completes. By preventing other processes from modifying the files, block mode prevents these processes from interrupting the migration. The downside is that preventing an application from modifying a file may result in delays or errors for the application.
- Non-block mode is enabled for the lfs migrate command with the -n option. When running lfs migrate in non-block mode, other processes can still modify the files that are being migrated. If a process modifies a file before lfs migrate finishes migrating it, lfs migrate will fail to migrate that file, leaving the file with its original stripe layout.

We recommend you use non-block mode, as it is less likely to interfere with your application.

#### Action to take

- 1. Launch a relatively large client instance (such as the Amazon EC2 c5n.4xlarge instance type) to mount to the file system.
- 2. Before running the non-block mode script pr the block-mode script, first run the following commands on each client instance to speed up the process:

```
sudo lctl set_param 'mdc.*.max_rpcs_in_flight=60'
sudo lctl set_param 'mdc.*.max_mod_rpcs_in_flight=59'
```

3. Start a screen session and run the non-block mode script or the block mode script. Make sure to change the appropriate variables in the scripts:

Unbalanced storage on OSTs 359

• Non-block mode script:

```
#!/bin/bash
# UNCOMMENT THE FOLLOWING LINES:
# TRY_COUNT=0
# MAX_MIGRATE_ATTEMPTS=100
# OSTS="fsname-OST0000_UUID"
# DIR_OR_FILE_MIGRATED="/mnt/subdir/"
# BATCH_SIZE=10
# PARALLEL_JOBS=16 # up to max-procs processes, set to 16 if client is
 c5n.4xlarge with 16 vcpu
# LUSTRE_STRIPING_CONFIG="-E 100M -c 1 -E 10G -c 8 -E 100G -c 16 -E -1 -c 32" #
should be consistent with the existing striping setup
if [ -z "$TRY_COUNT" -o -z "$MAX_MIGRATE_ATTEMPTS" -o -z "$OSTS" -o -z
"$DIR_OR_FILE_MIGRATED" -o -z "$BATCH_SIZE" -o -z "$PARALLEL_JOBS" -o -z
"$LUSTRE_STRIPING_CONFIG" ]; then
    echo "Some variables are not set."
    exit 1
fi
echo "lfs migrate starts"
while true; do
    output=$(sudo lfs find ! -L released --ost $OSTS --print0
 $DIR_OR_FILE_MIGRATED | shuf -z | /bin/xargs -0 -P $PARALLEL_JOBS -n $BATCH_SIZE
 sudo lfs migrate -n $LUSTRE_STRIPING_CONFIG 2>&1)
    if [[ $? -eq 0 ]]; then
        echo "lfs migrate succeeds for $DIR_OR_FILE_MIGRATED at the $TRY_COUNT
 attempt, exiting."
        exit 0
    elif [[ $? -eq 123 ]]; then
        echo "WARN: Target data objects are not located on these OSTs. Skipping
 lfs migrate"
       exit 1
    else
        echo "lfs migrate fails for $DIR_OR_FILE_MIGRATED at the $TRY_COUNT
 attempt, retrying..."
        if (( ++TRY_COUNT >= MAX_MIGRATE_ATTEMPTS )); then
            echo "WARN: Exceeds max retry attempt. Skipping lfs migrate for
 $DIR_OR_FILE_MIGRATED. Failed with the following error"
```

Unbalanced storage on OSTs 360

```
echo $output
exit 1
fi
fi
done
```

- Block mode script:
  - Replace the values in OSTS with the values of your OSTs.
  - Provide an integer value to nproc to set the number of max-procs processes to run in parallel. For example, the Amazon EC2 c5n.4xlarge instance type has 16 vCPUs, so you can use 16 (or a value < 16) for nproc.
  - Provide your mount directory path in mnt\_dir\_path.

```
# find all OSTs with usage above a certain threshold; for example, greater than
  or equal to 85% full
for OST in $(lfs df -h |egrep '( 8[5-9]| 9[0-9]|100)%'|cut -d' ' -f1); do echo
  ${OST};done|tr '\012' ','

# customer can also just pass OST values directly to OSTS variable
OSTS='dzfevbmv-OST0000_UUID,dzfevbmv-OST0002_UUID,dzfevbmv-OST0004_UUID,dzfevbmv-
OST0005_UUID,dzfevbmv-OST0006_UUID,dzfevbmv-OST0008_UUID'

nproc=<Run up to max-procs processes if client is c5n.4xlarge with 16 vcpu, this
  value can be set to 16>

mnt_dir_path=<mount dir, e.g. '/my_mnt'>

lfs find ${mnt_dir_path} --ost ${OSTS}| xargs -P ${nproc} -n2 lfs migrate -E 100M
  -c 1 -E 10G -c 8 -E 100G -c 16 -E -1 -c 32
```

#### **Notes**

- If you notice that there is an impact on the performance of the reads of the file system, you can stop the migrations at any time by using ctrl-c or kill -9, and reduce the number of threads (nproc value) back to a lower number (such as 8), and resume migrating files.
- The lfs migrate command will fail on a file that is also opened by the client workload. It will throw an error and move to the next file; therefore, it is possible if there are many files being accessed, the script will not be able to migrate any files, and it will be reflected as the migration is making very slow progress.

Unbalanced storage on OSTs 361

- You can monitor OST usage using either of the following methods
  - On client mount, run the following command to monitor OST usage and find the OST with usage greater than 85%:

```
lfs df -h |egrep '( 8[5-9]| 9[1-9]|100)%'
```

- Check the Amazon CloudWatch metric, OST FreeDataStorageCapacity, check Minimum.
   If your script is finding OSTs that are over 85% full, then when the metric is close to 15%, use ctrl-c or kill -9 to stop the migration.
- You may also consider changing the stripe configuration of your file system or a directory, so
  that new files are striped across multiple storage targets. For more information, see in <a href="Striping">Striping</a>
  data in your file system.

## **Troubleshooting FSx for Lustre CSI driver issues**

Amazon FSx for Lustre supports access from containers running on Amazon EKS using the opensource FSx for Lustre CSI driver. For deployment information, see <u>Use Amazon FSx for Lustre</u> Storage in the Amazon EKS User Guide.

If you're experiencing issues with the FSx for Lustre CSI driver for containers running on Amazon EKS, see Troubleshooting CSI Driver (Common Issues) which is available on GitHub.

CSI driver issues 362

## **Additional information**

This section provides a reference of supported, but deprecated Amazon FSx features.

#### **Topics**

Setting up a custom backup schedule

# Setting up a custom backup schedule

We recommend using AWS Backup to set up a custom backup schedule for your file system. The information provided here is for reference purposes if you need to schedule backups more frequently than you can when using AWS Backup.

When enabled, Amazon FSx automatically takes a backup of your file system once a day during a daily backup window. Amazon FSx enforces a retention period that you specify for these automatic backups. It also supports user-initiated backups, so you can make backups at any point.

Following, you can find the resources and configuration to deploy custom backup scheduling. Custom backup scheduling performs user-initiated backups on an Amazon FSx for Lustre file system on a custom schedule that you define. Examples might be once every six hours, once every week, and so on. This script also configures deleting backups older than your specified retention period.

The solution automatically deploys all the components needed, and takes in the following parameters:

- · The file system
- A CRON schedule pattern for performing backups
- The backup retention period (in days)
- The backup name tags

For more information on CRON schedule patterns, see <u>Schedule Expressions for Rules</u> in the Amazon CloudWatch User Guide.

## **Architecture overview**

This solution does the following:

FSx for Lustre Lustre User Guide

1. The CloudFormation template deploys an CloudWatch Event, a Lambda function, an Amazon SNS queue, and an IAM role. The IAM role gives the Lambda function permission to invoke the Amazon FSx for Lustre API operations.

- 2. The CloudWatch event runs on a schedule you define as a CRON pattern, during the initial deployment. This event invokes the solution's backup manager Lambda function that invokes the Amazon FSx for Lustre CreateBackup API operation to initiate a backup.
- 3. The backup manager retrieves a list of existing user-initiated backups for the specified file system using DescribeBackups. It then deletes backups older than the retention period, which you specify during the initial deployment.
- 4. The backup manager sends a notification message to the Amazon SNS queue on a successful backup if you choose the option to be notified during the initial deployment. A notification is always sent in the event of a failure.

## **CloudFormation template**

This solution uses CloudFormation to automate the deployment of the Amazon FSx for Lustre custom backup scheduling solution. To use this solution, download the fsx-scheduledbackup.template CloudFormation template.

## **Automated deployment**

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



#### (i) Note

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

#### To launch the custom backup solution stack

Download the fsx-scheduled-backup.template CloudFormation template. For more information on creating an CloudFormation stack, see Creating a Stack on the AWS CloudFormation Console in the AWS CloudFormation User Guide.

CloudFormation template 364

Lustre User Guide FSx for Lustre



#### Note

By default, this template launches in the US East (N. Virginia) AWS Region. Amazon FSx for Lustre is currently only available in specific AWS Regions. You must launch this solution in an AWS Region where Amazon FSx for Lustre is available. For more information, see the Amazon FSx section of AWS Regions and Endpoints in the AWS General Reference.

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

| Parameter                            | Default               | Description   |
|--------------------------------------|-----------------------|---|
| Amazon FSx for Lustre file system ID | No default value      | The file system ID for the file system that you want to back up.  |
| CRON schedule pattern for backups.   | 0 0/4 * * ? *         | The schedule to run the CloudWatch event, triggerin g a new backup and deleting old backups outside of the retention period.          |
| Backup retention (days)              | 7                     | The number of days to keep user-initiated backups. The Lambda function deletes user-initiated backups older than this number of days. |
| Name for backups                     | user-scheduled backup | The name for these backups, which appears in the <b>Backup Name</b> column of the Amazon FSx for Lustre Management Console.           |
| Backup notifications                 | Yes                   | Choose whether to be notified when backups are  |

Automated deployment 365

| Parameter     | Default          | Description  |
|---------------|------------------|--|
|               |                  | successfully initiated. A notification is always sent if there's an error. |
| Email address | No default value | The email address to subscribe to the SNS notifications.                   |

- Choose Next.
- 4. For **Options**, choose **Next**.
- 5. For **Review**, review and confirm the settings. You must select the check box acknowledging that the template create IAM resources.
- 6. Choose **Create** to deploy the stack.

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

## **Additional options**

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

```
{
  "start-backup": "true",
  "purge-backups": "true",
  "filesystem-id": "${FileSystemId}",
  "notify_on_success": "${SuccessNotification}"
}
```

To schedule backups for an additional Amazon FSx for Lustre file system, create another CloudWatch event rule. You do so using the Schedule event source, with the Lambda function created by this solution as the target. Choose **Constant (JSON text)** under **Configure Input**. For the JSON input, simply substitute the file system ID of the Amazon FSx for Lustre file

Additional options 366

system to back up in place of \${FileSystemId}. Also, substitute either Yes or No in place of \${SuccessNotification} in the JSON above.

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

Additional options 367

# **Document history**

• API version: 2018-03-01

• Latest documentation update: September 30, 2025

The following table describes important changes to the *Amazon FSx for Lustre User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

| Change  | Description  | Date               |
|---|--|--------------------|
| Additional AWS Region support added for Persistent deployment type  | Persistent 2 SSD FSx for Lustre file systems are now available in the US West (Phoenix) Local Zone. For more information, see Deployment type availability.                  | September 30, 2025 |
| Lustre client support added<br>for Ubuntu 24 Kernel 6.14.0          | The FSx for Lustre client now supports Amazon EC2 instances running Ubuntu 24.04 Kernel 6.14.0. For more information, see <u>Installing the Lustre client</u> .              | September 24, 2025 |
| Lustre client support added<br>for Amazon Linux 2023<br>Kernel 6.12 | The FSx for Lustre client now supports Amazon EC2 instances running Amazon Linux 2023 Kernel 6.12. For more information, see Installing the Lustre client.                   | September 9, 2025  |
| Additional AWS Region support added                                 | FSx for Lustre file systems are now available in Asia Pacific (Taipei). For more informati on, see <a href="Deployment type availability">Deployment type availability</a> . | August 18, 2025    |

Amazon FSx updated the

AmazonFSxServiceRolePolicy

AWS managed policy

Amazon FSx added the ec2:AssignIpv6Addr esses and ec2:UnassignIpv6Addresses permissions to the AmazonFSxServiceRolePolicy. For more information, see Amazon FSx updates to AWS managed policies.

July 22, 2025

Lustre client support for
Rocky Linux and Red Hat
Enterprise Linux (RHEL) 9.6
added

The FSx for Lustre client now supports Amazon EC2 instances running Rocky Linux and Red Hat Enterpris e Linux (RHEL) 9.6. For more information, see <u>Installing the Lustre client</u>.

July 1, 2025

Amazon FSx updated the AmazonFSxFullAccess AWS managed policy The AmazonFSxFullAccess
managed policy was updated
to add the fsx:Creat
eAndAttachS3Access
Point ,fsx:Descr
ibeS3AccessPointAt
tachments , and
fsx:DetachAndDelet
eS3AccessPoint
permissions.

June 25, 2025

Amazon FSx updated the
AmazonFSxConsoleFullAccess
AWS managed policy

The AmazonFSxConsoleFu

LlAccess managed policy
was updated to add the
fsx:CreateAndAttac
hS3AccessPoint ,
fsx:DescribeS3Acce
ssPointAttachments ,
and fsx:DetachAndDelet
eS3AccessPoint

permissions.

June 25, 2025

Support added for Intelligent-Tiering storage class You can now create FSx for Lustre 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. For more information, see Performance characteristics of Intelligent-Tiering storage class.

May 29, 2025

Additional AWS Region support added

FSx for Lustre file systems are now available in Asia Pacific (Thailand) and Mexico (Central). For more informati on, see <a href="Deployment type">Deployment type</a> availability.

May 8, 2025

Lustre client support for Ubuntu 24 added

The FSx for Lustre client now supports Amazon EC2 instances running Ubuntu 24.04. For more informati on, see <u>Installing the Lustre</u>

March 19, 2025

| Amazon FSx updated th | <u>1e</u> |
|-----------------------|-----------|
| AmazonFSxConsoleRe    |           |
| adOnlyAccess AWS mar  | naged     |
| policy                |           |

Amazon FSx updated the AmazonFSxConsoleRe adOnlyAccess policy to add the ec2:DescribeNetwor kInterfaces permissio n. For more information, see the AmazonFSxConsoleRe adOnlyAccess policy.

February 25, 2025

### Support added for upgrading Lustre version

You can now upgrade the Lustre version of your FSx for Lustre file system to a newer version. For more informati on, see Managing Lustre version.

February 12, 2025

# Amazon FSx updated the AmazonFSxConsoleFullAccess AWS managed policy

Amazon FSx updated the AmazonFSxConsoleFu llAccess policy to add the ec2:DescribeNetwor kInterfaces permission. For more information, see the AmazonFSxConsoleFullAccess policy.

February 7, 2025

# Additional AWS Region support added for Persistent deployment type

Persistent 2 SSD FSx for Lustre file systems are now available in the Asia Pacific (Malaysia) AWS Region. For more information, see Deployment type availability.

January 2, 2025

Lustre client support for
Rocky Linux and Red Hat
Enterprise Linux (RHEL) 9.5
added

The FSx for Lustre client now supports Amazon EC2 instances running Rocky Linux and Red Hat Enterpris e Linux (RHEL) 9.5. For more information, see <u>Installing the Lustre client</u>.

December 26, 2024

Support added for EFA

You can now create an FSx for Lustre Persistent 2 file system with support for Elastic Fabric Adapter (EFA) that provides increased network performan ce for client instances that support EFA. Enabling EFA also provides support for GPUDirect Storage (GDS) and ENA Express. For more information, see Working with EFA-enabled file systems.

November 27, 2024

Additional AWS Region
support added for Persistent
deployment type

Persistent 2 SSD FSx for Lustre file systems are now available in the US West (N. California) AWS Region. For more information, see Deployment type availability. November 27, 2024

Lustre client support added for Ubuntu 22 Kernel 6.8.0

The FSx for Lustre client now supports Amazon EC2 instances running Ubuntu 22.04 Kernel 6.8.0. For more information, see <u>Installing the Lustre client</u>.

November 8, 2024

Support added for additional Amazon CloudWatch metrics and an enhanced monitoring dashboard FSx for Lustre now provides additional network, performance, and storage metrics, and an enhanced monitoring dashboard for improved visibility into file system activity. For more information, see Monitoring with Amazon CloudWatch.

September 25, 2024

Additional AWS Region
support added for Persistent
deployment type

Persistent 2 SSD FSx for Lustre file systems are now available in the US East (Dallas) Local Zone. For more information, see <u>Deployment</u> type availability.

September 20, 2024

Lustre client support added for Ubuntu 22 Kernel 6.5.0

The FSx for Lustre client now supports Amazon EC2 instances running Ubuntu 22.04 Kernel 6.5.0. For more information, see <u>Installing the Lustre client</u>.

August 1, 2024

Lustre client support for
CentOS, Rocky Linux, and Red
Hat Enterprise Linux (RHEL)
8.10 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS, Rocky Linux, and Red Hat Enterprise Linux (RHEL) 8.10. For more information, see Installing the Lustre client.

June 18, 2024

Support added for increasing metadata performance

You can now create an FSx for Lustre Persistent 2 file system with a metadata configuration that provides the ability to increase metadata performance. For more information, see File system metadata performance and Managing metadata performance.

June 6, 2024

Additional AWS Region
support added for Persistent
deployment type

Persistent 2 SSD FSx for Lustre file systems are now available in the US East (Atlanta) Local Zone. For more information, see <u>Deployment</u> type availability.

May 29, 2024

Lustre client support for
Rocky Linux and Red Hat
Enterprise Linux (RHEL) 9.4
added

The FSx for Lustre client now supports Amazon EC2 instances running Rocky Linux and Red Hat Enterpris e Linux (RHEL) 9.4. For more information, see <u>Installing the Lustre client</u>.

May 16, 2024

Additional AWS Region
support added for Persistent
deployment type

Persistent 2 SSD FSx for Lustre file systems are now available in the Canada West (Calgary) AWS Region. For more information, see Deployment type availability.

May 3, 2024

<u>Lustre client support for</u> Amazon Linux 2023 added The FSx for Lustre client now supports Amazon EC2 instances running Amazon Linux 2023. For more information, see <u>Installing the Lustre Client</u>.

March 25, 2024

<u>Lustre client support for</u>
<u>CentOS, Rocky Linux, and Red</u>
<u>Hat Enterprise Linux (RHEL)</u>
8.9 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS, Rocky Linux, and Red Hat Enterprise Linux (RHEL) 8.9. For more information, see Installing the Lustre client. January 9, 2024

Amazon FSx updated the
AmazonFSxFullAccess,
AmazonFSxConsoleFu
llAccess, AmazonFSx
ReadOnlyAccess, AmazonFSx
ConsoleReadOnlyAccess, and
AmazonFSxServiceRolePolicy
AWS managed policies

Amazon FSx updated the AmazonFSxFullAccess, AmazonFSxConsoleFu llAccess, AmazonFSx ReadOnlyAccess, AmazonFSx ConsoleReadOnlyAccess, and AmazonFSxServiceRo lePolicy policies to add the ec2:GetSecurityGro upsForVpc permission. For more information, see Amazon FSx updates to AWS managed policies.

January 9, 2024

Lustre client support for
Rocky Linux and Red Hat
Enterprise Linux (RHEL) 9.0
and 9.3 added

The FSx for Lustre client now supports Amazon EC2 instances running Rocky Linux and Red Hat Enterprise Linux (RHEL) 9.0 and 9.3. For more information, see <u>Installing the Lustre client</u>.

December 20, 2023

Amazon FSx for Lustre
updated the AmazonFSx
FullAccess and the
AmazonFSxConsoleFullAccess
AWS managed policies

Amazon FSx updated the
AmazonFSxFullAccess and
AmazonFSxConsoleFu
llAccess policies to add the
ManageCrossAccount
DataReplication action.
For more information, see
Amazon FSx updates to AWS

managed policies.

December 20, 2023

Amazon FSx updated the
AmazonFSxFullAccess and the
AmazonFSxConsoleFullAccess
AWS managed policies

Amazon FSx updated the AmazonFSxFullAccess and AmazonFSxConsoleFu llAccess policies to add the fsx:CopySnapshotAn dUpdateVolume permissio n. For more information, see Amazon FSx updates to AWS managed policies.

November 26, 2023

Support added for throughpu t capacity scaling

You can now modify the throughput capacity for existing FSx for Lustre persistent SSD-based file systems as your throughput requirements evolve. For more information, see Managing throughput capacity.

November 16, 2023

Amazon FSx updated the
AmazonFSxFullAccess and the
AmazonFSxConsoleFullAccess
AWS managed policies

Amazon FSx updated the AmazonFSxFullAccess and AmazonFSxConsoleFu llAccess policies to add the fsx:DescribeShared VPCConfiguration and fsx:UpdateSharedVP CConfiguration permissions. For more information, see Amazon FSx updates to AWS managed policies.

November 14, 2023

Support added for project quotas

You can now create storage quotas for projects. A project quota applies to all files or directories associated with a project. For more information, see Storage quotas.

August 29, 2023

Support added for Lustre version 2.15

All FSx for Lustre file systems are now built on Lustre version 2.15 when created using the Amazon FSx console. For more informati on, see <a href="Step 1: Create your Amazon FSx">Step 1: Create your Amazon FSx for Lustre file system.</a>

August 29, 2023

Additional AWS Region
support added for Persistent
deployment type

Persistent 2 FSx for Lustre file systems are now available in the Israel (Tel Aviv) AWS Region. For more information, see <u>Deployment options for FSx for Lustre file systems.</u>

August 24, 2023

Support added for release data repository tasks

FSx for Lustre now provides release data repository tasks to release archived files from a file system linked to an S3 data repository. Releasing a file retains the file listing and metadata, but removes the local copy of that file's contents. For more informati on, see <u>Using data repository</u> tasks to release files.

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 <a href="MazonFSxFullAccess"><u>AmazonFSxFullAccess</u></a> policy.

July 13, 2023

Amazon FSx updated the

AmazonFSxConsoleFullAccess

AWS managed policy

Amazon FSx updated the AmazonFSxConsoleFu llAccess policy to remove the fsx:\* permission and add specific fsx actions. For more information, see <a href="MazonFSx"><u>AmazonFSx</u></a> ConsoleFullAccess policy.

July 13, 2023

Lustre client support for
CentOS, Rocky Linux, and Red
Hat Enterprise Linux (RHEL)
8.8 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS, Rocky Linux, and Red Hat Enterprise Linux (RHEL) 8.8. For more information, see Installing the Lustre client. May 25, 2023

Support added for AutoImport and AutoExport metrics

FSx for Lustre now provides
Amazon CloudWatch metrics
that monitor automatic
import and automatic export
updates for file systems
linked to data repositories.
For more information, see
Monitoring with Amazon
CloudWatch.

March 31, 2023

DRA support for Persistent

1 and Scratch 2 deployment

types added

You can now create data repository associations to link data repositories to Lustre 2.12 file systems with Persistent 1 or Scratch 2 deployment types. For more information, see <u>Using data repositories with Amazon FSx for Lustre</u>.

March 29, 2023

Lustre client support for
CentOS, Rocky Linux, and Red
Hat Enterprise Linux (RHEL)
8.7 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS, Rocky Linux, and Red Hat Enterprise Linux (RHEL) 8.7. For more information, see Installing the Lustre client. December 5, 2022

Additional AWS Region
support added for Persistent
deployment type

Next-generation Persisten t 2 SSD FSx for Lustre file systems are now available in the Europe (Stockholm), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), and Asia Pacific (Seoul) AWS Regions. For more information, see Deployment options for FSx for Lustre file systems. November 10, 2022

Lustre client support for
CentOS, Rocky Linux, and Red
Hat Enterprise Linux (RHEL)
8.6 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS, Rocky Linux, and Red Hat Enterprise Linux (RHEL) 8.6. For more information, see Installing the Lustre client.

September 8, 2022

<u>Lustre client support for</u> Ubuntu 22 added The FSx for Lustre client now supports Amazon EC2 instances running Ubuntu 22.04. For more informati on, see <u>Installing the Lustre client</u>.

July 28, 2022

Lustre client support for Rocky Linux added

The FSx for Lustre client now supports Amazon EC2 instances running Rocky Linux. For more informati on, see <u>Installing the Lustre client</u>.

July 8, 2022

<u>Support added for Lustre root</u> squash

You can now use the Lustre root squash feature to restrict root level access from clients that try to access your FSx for Lustre file system as root. For more information, see <u>Lustre</u> root squash.

May 25, 2022

Additional AWS Region
support added for Persistent
deployment type

Next-generation Persisten t 2 SSD FSx for Lustre file systems are now available in the Europe (London), Asia Pacific (Singapore), and Asia Pacific (Sydney) AWS Regions. For more information, see Deployment options for FSx for Lustre file systems. April 19, 2022

Support added for using AWS
DataSync to migrate files to
your Amazon FSx for Lustre
file systems.

You can now use AWS
DataSync to migrate files
from existing file systems to
FSx for Lustre file systems.
For more information, see
How to migrate existing files
to FSx for Lustre 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 <a href="Market Market No.1"><u>Amazon FSx and interface VPC endpoints.</u></a>

April 5, 2022

# Support added for Lustre DRA queuing

You can now create a DRA (data repository associati on) when you create an FSx for Lustre file system. The request will be queued and the DRA will be created once the file system is available. For more information, see Linking your file system to an S3 bucket.

February 28, 2022

<u>CentOS and Red Hat Enterpris</u> e Linux (RHEL) 8.5 added The FSx for Lustre client now supports Amazon EC2 instances running CentOS and Red Hat Enterprise Linux (RHEL) 8.5. For more information, see <u>Installing the</u> Lustre client.

December 20, 2021

Support for exporting changes from FSx for Lustre into a linked data repository

You can now configure FSx for Lustre to automatically export new, changed, and deleted files from your file system to a linked Amazon S3 data repository. You can use data repository tasks to export data and metadata changes to the data repositor y. You can also configure links to multiple data repositories. For more information, see Exporting changes to the data repository.

November 30, 2021

| Support | added | for | Lustre |
|---------|-------|-----|--------|
| logging |       |     |        |

You can now configure FSx for Lustre to log error and warning events for data repositories associated with your file system to Amazon CloudWatch Logs. For more information, see Logging with Amazon CloudWatch Logs.

November 30, 2021

Persistent SSD file systems support higher throughput and smaller storage capacity

Next-generation Persisten t SSD FSx for Lustre file systems have higher throughput options and have a lower minimum storage capacity. For more informati on, see <a href="Deployment options">Deployment options</a> for FSx for Lustre file systems.

November 30, 2021

Support added for Lustre version 2.12

You can now choose Lustre version 2.12 when you create an FSx for Lustre file system. For more information, see <a href="Step 1: Create your Amazon FSx">Step 1: Create your Amazon FSx for Lustre file system</a>.

October 5, 2021

<u>Lustre client support for</u>
<u>CentOS and Red Hat Enterpris</u>
e Linux (RHEL) 8.4 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS and Red Hat Enterprise Linux (RHEL) 8.4. For more information, see <u>Installing the Lustre client</u>.

June 9, 2021

| Support adde | d for | data |
|--------------|-------|------|
| compression  |       |      |

You can now enable data compression when you create an FSx for Lustre file system. You can also enable or disable data compression on an existing FSx for Lustre file system. For more information, see Lustre data compression.

May 27, 2021

# Support added for copying backups

You can now use Amazon
FSx to copy backups within
the same AWS account to
another AWS Region (crossRegion copies) or within the
same AWS Region (in-Region
copies). For more information,
see Copying backups.

April 12, 2021

## Lustre client support for Lustre filesets

The FSx for Lustre client now supports using filesets to mount only a subset of the file system namespace . For more information, see Mounting Specific Filesets.

March 18, 2021

# Support added for clients access using non-private IP addresses

You can access FSx for Lustre file systems from an on-premises client using non-private IP addresses. For more information, see Mounting Amazon FSx file systems from on-premises or a peered Amazon VPC.

December 17, 2020

<u>Lustre client support for Arm-</u>based CentOS 7.9 added

The FSx for Lustre client now supports Amazon EC2 instances running Armbased CentOS 7.9. For more information, see <u>Installing the Lustre client</u>.

December 17, 2020

<u>Lustre client support for</u>
<u>CentOS and Red Hat Enterpris</u>
e Linux (RHEL) 8.3 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS and Red Hat Enterprise Linux (RHEL) 8.3. For more information, see <u>Installing the Lustre client</u>.

December 16, 2020

Support added for storage and throughput capacity scaling

You can now increase the storage and throughput capacity for existing FSx for Lustre file systems as your storage and throughpu t requirements evolve. For more information, see Managing storage and throughput capacity.

November 24, 2020

Support added for storage quotas

You can now create storage quotas for users and groups. Storage quotas limit the amount of disk space and the number of files that a user or group can consume on your FSx for Lustre file system. For more information, see Storage quotas.

November 9, 2020

# Amazon FSx is now integrated with AWS Backup

You can now use AWS Backup to back up and restore your FSx file systems in addition to using the native Amazon FSx backups. For more informati on, see <u>Using AWS Backup</u> with Amazon FSx.

November 9, 2020

# Support added for the HDD (hard disk drive) storage options

In addition to the SSD (solid state drive) storage option, FSx for Lustre now supports the HDD (hard disk drive) storage option. You can configure your file system to use HDD for throughpu t-intensive workloads that typically have large, sequentia I file operations. For more information, see Multiple Storage Options.

August 12, 2020

# Support for importing linked data repository changes into FSx for Lustre

You can now configure your FSx for Lustre file system to automatically import new files added to and files that have changed in a linked data repository after file system creation. For more information, see <u>Automatic ally import updates from the data repository</u>.

July 23, 2020

Linux SP4 and SP5 added

The FSx for Lustre client now supports Amazon EC2 instances running SUSE Linux SP4 and SP5. For more information, see <u>Installing the</u> Lustre client.

July 20, 2020

<u>Lustre client support for</u>
<u>CentOS and Red Hat Enterpris</u>
e Linux (RHEL) 8.2 added

The FSx for Lustre client now supports Amazon EC2 instances running CentOS and Red Hat Enterprise Linux (RHEL) 8.2. For more information, see <u>Installing the Lustre client</u>.

July 20, 2020

Support for automatic and manual file system backups added

You can now take automatic daily backups and manual backups of file systems not linked to an Amazon S3 durable data repository. For more information, see Working with backups.

June 23, 2020

Two new file system
deployment types released

Scratch file systems are designed for temporary storage and shorter-term processing of data. Persisten t file systems are designed for longer-term storage and workloads. For more information, see <u>FSx for</u> Lustre Deployment options.

February 12, 2020

| Support for | POSIX | metadata |
|-------------|-------|----------|
| added       |       |          |

FSx for Lustre retains associated POSIX metadata when importing and exporting files to a linked durable data repository on Amazon S3. For more information, see POSIX metadata support for data repositories.

December 23, 2019

# New data repository tasks feature released

You can now export changed data and associated POSIX metadata to a linked durable data repository on Amazon S3 using data repository tasks. For more information, see Data repository tasks.

December 23, 2019

# Additional AWS Region support added

FSx for Lustre is now available in the Europe (London)
Region AWS Region. For FSx for Lustre region-specific limits, see Limits.

July 9, 2019

# Additional AWS Region support added

FSx for Lustre is now available in the Asia Pacific (Singapor e) AWS Region. For FSx for Lustre region-specific limits, see Limits.

June 26, 2019

# Lustre client support for Amazon Linux and Amazon Linux 2 added

The FSx for Lustre client now supports Amazon EC2 instances running Amazon Linux and Amazon Linux 2. For more information, see Installing the Lustre Client.

March 11, 2019

# <u>User-defined data export</u> path support added

Users now have the option to overwrite the original objects in your Amazon S3 bucket or write the new or changed files to a prefix that you specify. With this option, you have additiona I flexibility to incorpora te FSx for Lustre into your data processing workflows . For more information, see Exporting Data to Your Amazon S3 Bucket.

February 6, 2019

# Total storage default limit increased

The default total storage for all FSx for Lustre file systems increased to 100,800 GiB. For more information, see Limits.

January 11, 2019

# Amazon FSx for Lustre is now generally available

Amazon FSx for Lustre is a fully managed file system that is optimized for compute-intensive workloads, such as high-perf ormance computing, machine learning, and media processin g workflows.

November 28, 2018