FSx for Lustre

Lustre User Guide

FSx for Lustre: Lustre User Guide

Table of Contents

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 open-source 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 submillisecond latencies, up to hundreds of GBps 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 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

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 workloads. In persistent file systems, data is replicated, and file servers are replaced if they fail. For more information, see File system deployment options for FSx for Lustre (p. 15).

Multiple storage options

Amazon FSx for Lustre offers a choice of SSD (solid state drive) and HDD (hard disk drive) storage types that are optimized for different data processing requirements.

- SSD storage options — For low-latency, IOPS-intensive workloads that typically have small, random file operations, choose one of the SSD storage options.
- HDD storage options — For throughput-intensive workloads that typically have large, sequential file operations, choose one of the HDD storage options.

If you are provisioning a file system with the HDD storage option, you might also want to consider provisioning a read-only SSD cache automatically sized to 20 percent of your HDD storage capacity. This provides sub-millisecond latencies and higher IOPS for frequently accessed files. Both SSD-based and HDD-based file systems are provisioned with SSD-based metadata servers so that all metadata operations, which represent the majority of file system operations, are delivered with sub-millisecond latencies.

For more information about performance of these storage options, see Amazon FSx for Lustre performance (p. 46).

FSx for Lustre and data repositories

FSx for Lustre file systems can be linked to data repositories on Amazon S3 or to an on-premises data store.

Amazon S3 integration

FSx for Lustre integrates with Amazon S3, making it easy 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 enables 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 Using data repositories with FSx for Lustre (p. 18) and Data repository tasks (p. 31).

On-premises data repositories

With Amazon FSx for Lustre, you can burst your data processing workloads from on-premises into the Amazon Web Services Cloud by importing data using AWS Direct Connect or AWS VPN. For more information, see Using Amazon FSx with your on-premises data repository (p. 45).

Accessing file systems

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

FSx for Lustre is accessible from compute workloads running on Amazon Elastic Compute Cloud (Amazon EC2) instances and containers running on Amazon Elastic Kubernetes Service (Amazon EKS).

- 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.

- 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.
- You access Amazon FSx for Lustre from Amazon Elastic Container Service (Amazon ECS) Docker containers on Amazon EC2 instances. For more information, see Mounting from Amazon Elastic Container Service (p. 70).

Amazon FSx for Lustre is compatible with the most popular Linux-based AMIs, including Amazon Linux, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, and SUSE Linux. The Lustre client is included with Amazon Linux 2 and Amazon Linux. For RHEL, CentOS, and Ubuntu, an AWS Lustre client repository provides clients that are compatible with these operating systems.

Using Amazon FSx, you can burst your compute-intensive workloads from on-premises into the AWS Cloud by importing data over AWS 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.

For more information, see Accessing file systems (p. 55).

Integrations with AWS services

Amazon FSx for Lustre integrates with SageMaker as an input data source. When using SageMaker 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 What Is SageMaker? in the *Amazon SageMaker Developer Guide*.

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 What Is AWS Batch? in the *AWS Batch User Guide*.

FSx for Lustre integrates with AWS ParallelCluster. AWS ParallelCluster is an AWS-supported open-source 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 EC2 instances. For more information about data encryption in FSx for Lustre, including AWS Regions where encryption of data in transit is supported, see Data encryption in Amazon FSx for Lustre (p. 108). Amazon FSx has been assessed to comply with ISO, PCI-DSS, and SOC certifications, and is HIPAA eligible. For more information, see Security in FSx for Lustre (p. 107).

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 Amazon EC2 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 (p. 111).

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 Getting started with Amazon FSx for Lustre (p. 8).
2. For information on performance, see Amazon FSx for Lustre performance (p. 46).
3. For information on linking your file system to an Amazon S3 bucket data repository, see Using data repositories with FSx for Lustre (p. 18).
4. For Amazon FSx for Lustre security details, see Security in FSx for Lustre (p. 107).
5. For information on the scalability limits of Amazon FSx for Lustre, including throughput and file system size, see Quotas (p. 138).
6. For information on the Amazon FSx for Lustre API, see the Amazon FSx for Lustre API Reference.

Setting up

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

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

Sign up for AWS

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

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

To create an AWS account

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

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

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

Create an IAM user

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

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

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

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

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

> **Note**
>
> We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

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

    > **Note**
    >
    > You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

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

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

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

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

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

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

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

Adding permissions to use data repositories in Amazon S3

Amazon 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 mounting your Amazon FSx for Lustre file system. For more information, see Using data repositories with FSx for Lustre (p. 18).

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

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com//iam/.
2. In the navigation pane, choose **Roles**.
3. In the list, choose the name of the role to embed a policy in.
4. Choose the **Permissions** tab.
5. 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.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:CreateServiceLinkedRole",
            "iam:AttachRolePolicy",
            "iam:PutRolePolicy"
        ],
        "Resource": "arn:aws:iam::*:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*"
    }
}
```

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 for Lustre (p. 123).

Next step

Getting started with Amazon FSx for Lustre (p. 8)

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

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 (p. 5).
- An Amazon EC2 instance running a supported Linux release in your virtual private cloud (VPC) based on the Amazon VPC service. You will install the Lustre client on this EC2 instance, and then mount your FSx for Lustre file system on the EC2 instance. The Lustre client supports Amazon Linux, Amazon Linux 2, CentOS and Red Hat Enterprise Linux 7.5, 7.6, 7.7, 7.8, 7.9, 8.2, and 8.3, SUSE Linux Enterprise Server 12 SP3, SP4, and SP5, and Ubuntu 16.04, 18.04, and 20.04. For this getting started exercise, we recommend using Amazon Linux 2.

  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.
- 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.
- Determine which type of Amazon FSx for Lustre file system you want to create, *scratch* or *persistent*. For more information, see File system deployment options for FSx for Lustre (p. 15).

Step 1: Create your Amazon FSx for Lustre file system

Next, you create your file system in the console.

To create your file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
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.
4. Provide the information in the **File system details** section:

   - 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 **+ - = . _ : /**.
   - For **Deployment and storage type**, choose one of the options.

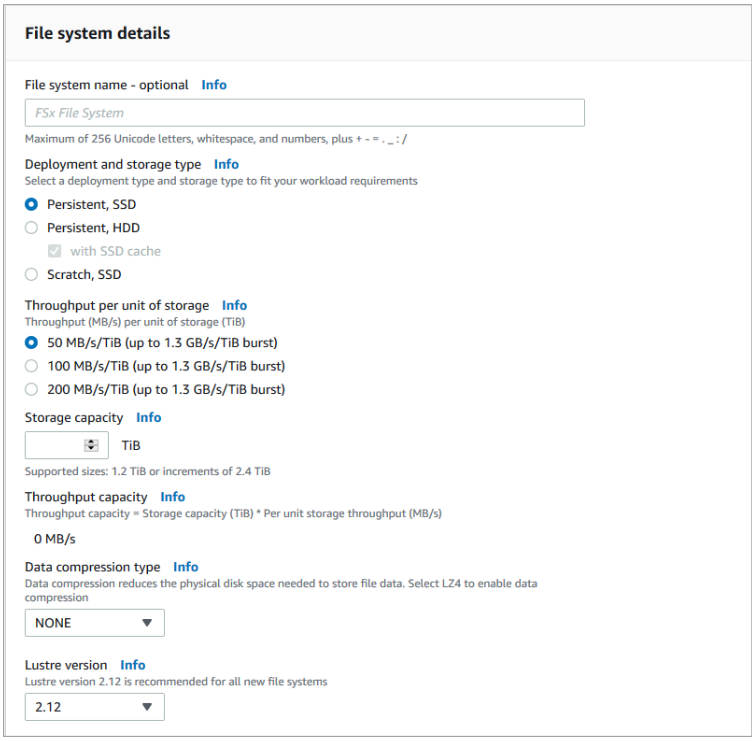
     SSD storage provides low-latency, IOPS-intensive workloads that typically have small, random file operations. **HDD** storage provides throughput-intensive workloads that typically have large, sequential file operations.

     - Choose the **Persistent, SSD** deployment type for longer-term storage and workloads. The file servers are highly available, data is automatically replicated within the file system's Availability Zone, and this type supports encrypting data in transit. To learn in which AWS Regions encrypting data in transit is available, see Encrypting data in transit (p. 109).
     - Choose the **Persistent, HDD** deployment type for longer-term storage and workloads. The file servers are highly available, data is automatically replicated within the file system's Availability Zone, and this type supports encrypting data in transit. To learn in which AWS Regions encrypting data in transit is available, see Encrypting data in transit (p. 109).

       If you choose **with SSD cache**, this cache is automatically sized to 20 percent of your HDD storage capacity to provide sub-millisecond latencies and higher IOPS for frequently accessed files.
     - Choose the **Scratch** deployment type for temporary storage and shorter-term processing of data. **Scratch 2** is the latest generation of scratch file systems, and offers higher burst throughput over baseline throughput and also in-transit encryption of data.
   - Choose the **Throughput per unit of storage** that you want for your file system. This option is only valid for **Persistent** deployment types.

     For SSD storage, set a value for **Throughput per unit of storage** to either 50, 100, or 200 MB/s per tebibyte (TiB). For HDD, set a value for **Throughput per unit of storage** to 12 or 40 MB/s per tebibyte (TiB). *Throughput per unit of storage* is the amount of read and write throughput for each 1 tebibyte (TiB) of storage provisioned, in MB/s/TiB. For a 2.4 TiB file system, provisioning 50 MB/s/TiB of per unit storage throughput yields 120 MB/s of file system throughput. You pay for the amount of throughput that you provision.
   - For **Storage capacity**, provide a storage capacity for your file system, in TiB:
     - For a persistent SSD or scratch 2 file system, this value can be 1.2 TiB or increments of 2.4 TiB. For a persistent HDD file system, this value can be increments of 6.0 TiB for 12 MB/s/TiB file systems and increments of 1.8 TiB for 40 MB/s/TiB file systems.
     - For a scratch 1 file system, this value can be 1.2, 2.4, or increments of 3.6 TiB.

     You can increase the amount of storage capacity as needed after you create the file system. For more information, see Managing storage and throughput capacity (p. 89).
   - 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 Lustre data compression (p. 94).
   - For **Lustre version**, choose **2.12** to create a Lustre 2.12 file system, which is the recommended version for all new file systems. Alternatively, you can choose **2.10** to create a Lustre 2.10 file system.



File system details

File system name - optional Info

FSx File System

Maximum of 256 Unicode letters, whitespace, and numbers, plus + - = . _ : /

Deployment and storage type Info

Select a deployment type and storage type to fit your workload requirements

● Persistent, SSD

○ Persistent, HDD

☑ with SSD cache

○ Scratch, SSD

Throughput per unit of storage Info

Throughput (MB/s) per unit of storage (TiB)

● 50 MB/s/TiB (up to 1.3 GB/s/TiB burst)

○ 100 MB/s/TiB (up to 1.3 GB/s/TiB burst)

○ 200 MB/s/TiB (up to 1.3 GB/s/TiB burst)

Storage capacity Info

TiB

Supported sizes: 1.2 TiB or increments of 2.4 TiB

Throughput capacity Info

Throughput capacity = Storage capacity (TiB) * Per unit storage throughput (MB/s)

0 MB/s

Data compression type Info

Data compression reduces the physical disk space needed to store file data. Select LZ4 to enable data compression

NONE

Lustre version Info

Lustre version 2.12 is recommended for all new file systems

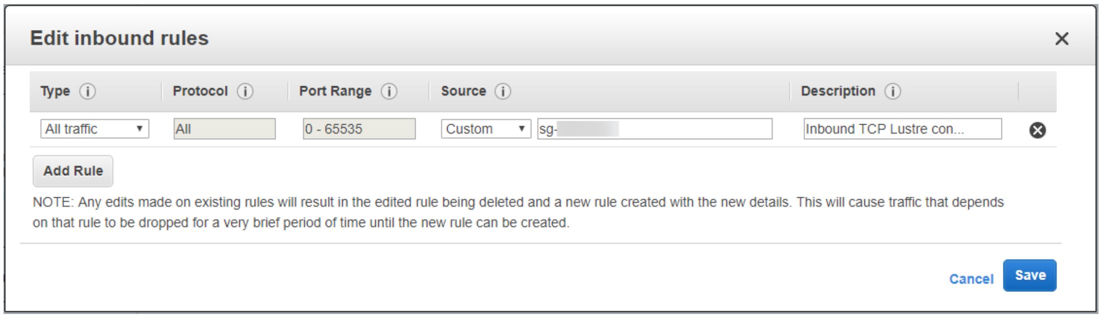
2.12

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

   - 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.

Type	Protocol	Port range	Source	Description
All TCP	TCP	0-65535	Custom *the_ID_of_this_security_group*	Inbound Lustre traffic rule

   The following screen capture shows an example of editing inbound rules.



Edit inbound rules

Type	Protocol	Port Range	Source		Description
All traffic	All	0 - 65535	Custom	sg-	Inbound TCP Lustre con...

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

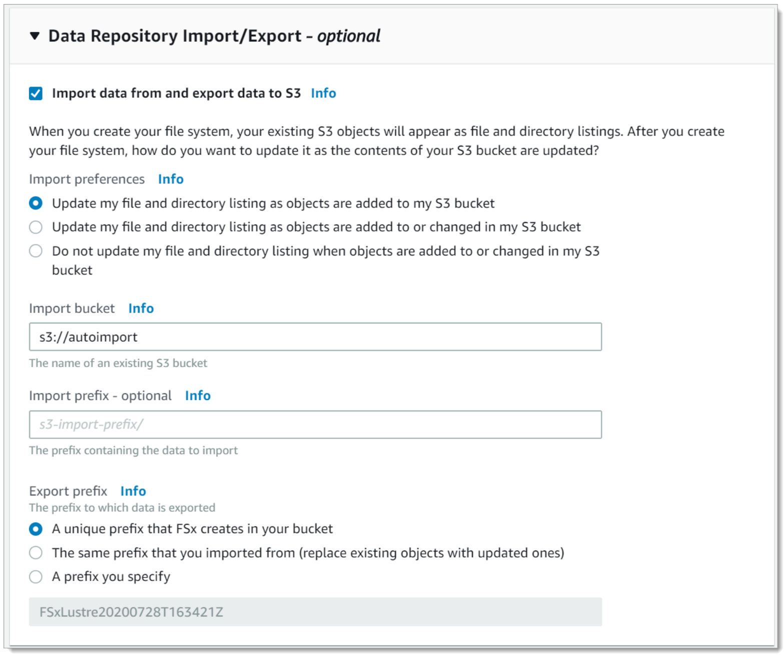
   - For **Subnet**, choose any value from the list of available subnets.

6. 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 the default Amazon FSx–managed key.
- 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 (p. 109).

7. (Optional) Use the **Data repository import/export** panel to configure a data repository linked to an Amazon S3 bucket.

   Select **Import data from and export data to S3** to expand the panel and configure the data repository settings.



▼ **Data Repository Import/Export -** *optional*

☑ **Import data from and export data to S3** Info

When you create your file system, your existing S3 objects will 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?

Import preferences Info

● Update my file and directory listing as objects are added to my S3 bucket
○ Update my file and directory listing as objects are added to or changed in my S3 bucket
○ Do not update my file and directory listing when objects are added to or changed in my S3 bucket

Import bucket Info

s3://autoimport

The name of an existing S3 bucket

Import prefix - optional Info

s3-import-prefix/

The prefix containing the data to import

Export prefix Info

The prefix to which data is exported

● A unique prefix that FSx creates in your bucket
○ The same prefix that you imported from (replace existing objects with updated ones)
○ A prefix you specify

FSxLustre20200728T163421Z

8. (Optional) Choose how Amazon FSx keeps your file and directory listings up to date automatically as you add or modify objects in your S3 bucket. By default, Amazon FSx imports file and directory listings at file system creation, and then whenever new files are added after this initial import. For more information, see Automatically import updates from your S3 bucket (p. 28).
9. Enter an optional **Import prefix** if you want to import only some of the file and directory listings in your S3 bucket into your file system. The import prefix defines where in your S3 bucket to import from. For more information, see Linking your file system to an S3 bucket (p. 22).
10. Keep **Export prefix** at the default setting. For more information about the data repository integration, see Linking your file system to an S3 bucket (p. 22).

    Important

    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.
11. In **Backup and maintenance -** ***optional***, you can do the following.

    For daily automatic backups:

    - Disable the **Daily automatic backup**, which is enabled by default.
    - Set the start time for **Daily automatic backup window**.
    - Set the **Automatic backup retention period**, from 1 - 35 days.

For more information, see Working with backups (p. 79).

12. Set the **Weekly maintenance window** start time, or keep it set to the default **No preference**.
13. Create any tags that you want to apply to your file system.
14. Choose **Next** to display the **Create file system summary** page.
15. 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 on your instance before mounting your file system

To mount your Amazon FSx for Lustre file system from your Amazon EC2 instance, first install the Lustre 2.10 client. The 2.10 versions of the Lustre client support Amazon FSx for Lustre version 2.12, in addition to version 2.10.

To download the Lustre client onto your Amazon EC2 instance

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. Do one of the following:

   - If the command returns `4.14.104-95.84.amzn2.x86_64` for x86-based EC2 instances, or `4.14.181-142.260.amzn2.aarch64` or higher for Graviton2-based EC2 instances, download and install the Lustre client with the following command.

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

   - If the command returns a result less than `4.14.104-95.84.amzn2.x86_64` for x86-based EC2 instances, or less than `4.14.181-142.260.amzn2.aarch64` for Graviton2-based EC2 instances, 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 above.

   For information about installing the Lustre client on other Linux distributions, see Installing the Lustre client (p. 55).

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 *mountname* with the file system's mount name, which you can get by running the **describe-file-systems** AWS CLI command or the DescribeFileSystems API operation.

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

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

   - `noatime` – Turns off updates to inode access times. To update inode access times, use the `mount` command without `noatime`.
   - `flock` – Enables file locking for your file system. If you don't want file locking enabled, use the `mount` command without `flock`.

3. 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 3: Run your analysis

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.

If you 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 Using data repositories with FSx for Lustre (p. 18).

(Optional) Step 4: Check Amazon FSx file system status

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

File system status	Description
AVAILABLE	The file system is in a healthy state, and is reachable and available for use.
CREATING	Amazon FSx is creating a new file system.
DELETING	Amazon FSx is deleting an existing file system.
UPDATING	The file system is undergoing a customer-initiated update.
MISCONFIGURED	The file system is in a failed but recoverable state.
FAILED	This status can mean either of the following: • The file system has failed and Amazon FSx can't recover it. • When creating a new file system, Amazon FSx couldn't create the file system.

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 Terminate Your Instance in the *Amazon EC2 User Guide for Linux Instances.*
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**.
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 How Do I Delete an S3 Bucket? in the *Amazon Simple Storage Service User Guide.*

Using available deployment options for Amazon FSx for Lustre file systems

FSx for Lustre provides a high performance, parallel file system that stores data across multiple network file servers to maximize performance and reduce bottlenecks. These servers have multiple disks. To spread load, Amazon FSx shards file system data into smaller chunks and spreads them across disks and servers using a process called striping. For more information about FSx for Lustre data striping, see Striping data in your file system (p. 50).

It's a best practice to link a highly durable long-term data repository residing on Amazon S3 with your FSx for Lustre high-performance file system.

In this scenario, you store your datasets on the S3 data repository. When you create your FSx for Lustre file system, you link it to your S3 data repository. At this point, the objects in your S3 bucket are listed as files and directories on your FSx file system. Amazon FSx then automatically copies the file contents from S3 to your Lustre file system when a file is accessed for the first time on the Amazon FSx file system. After your compute workload runs, or at any time, you can use a data repository task to export changes back to S3. For more information, see Using data repositories with FSx for Lustre (p. 18) and Using data repository tasks to export data and metadata changes (p. 42).

File system deployment options for FSx for Lustre

Amazon FSx for Lustre provides two file system deployment options: **scratch** and **persistent**.

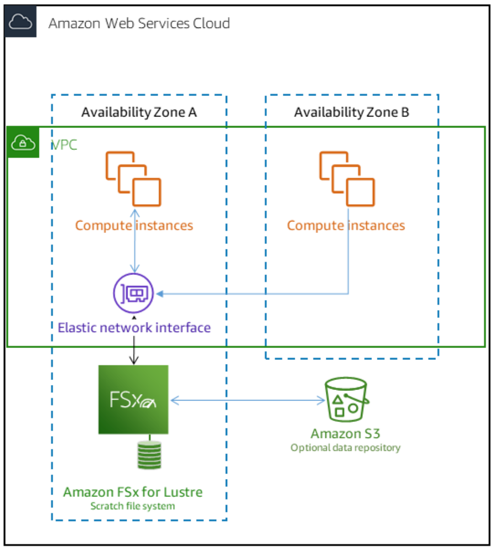
> **Note**
> Both deployment options support solid state drive (SSD) storage. However, hard disk drive (HDD) storage is only supported in the persistent deployment option.

Scratch file systems

Scratch file systems are designed for temporary storage and shorter-term processing of data. Data is not 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 Aggregate file system performance (p. 47).

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

The following diagram shows the architecture for an Amazon FSx for Lustre scratch file system.



On a scratch file system, file servers are not replaced if they fail and data is not 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. As larger file systems have more file servers and more disks, the probabilities of failure are increased.

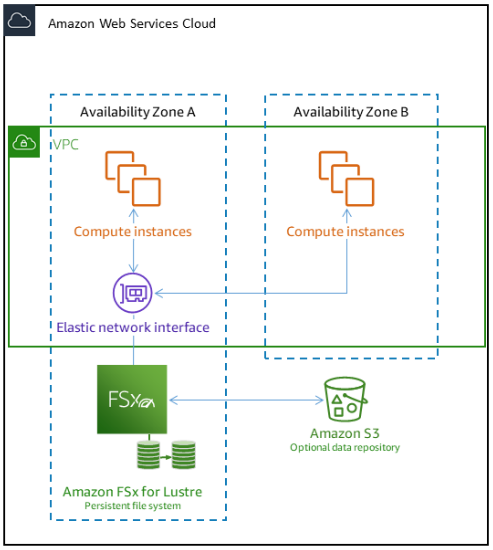
File System Size (TiB)	Number of File Servers	Availability/Durability Over One Day	Availability/Durability Over One Week
1.2	2	99.9%	99.4%
2.4	2	99.9%	99.4%
4.8	3	99.8%	99.2%
9.6	5	99.8%	98.6%
50.4	22	99.1%	93.9%

Persistent file systems

Persistent file systems are designed for longer-term storage and workloads. The file servers are highly available and data is automatically replicated within the same Availability Zone (AZ) that is associated with the file system. The data volumes attached to the file servers are replicated independently from the file servers to which they are attached.

Use persistent file systems for workloads that run for extended periods or indefinitely, and that might be sensitive to disruptions in availability.

The following diagram shows the architecture for an Amazon FSx for Lustre persistent file system, with replicated, highly available file servers and data volumes within a single Availability Zone.



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 is 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.

You choose the file system deployment type when you create a new file system, using the AWS Management Console, the AWS CLI, or the Amazon FSx for Lustre API. For more information, see Step 1: Create your Amazon FSx for Lustre file system (p. 8) and CreateFileSystemLustreConfiguration in the *Amazon FSx API Reference*.

Using data repositories with FSx for Lustre

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. They typically have datasets stored on long-term durable data repositories like Amazon S3 or on-premises storage. FSx for Lustre is natively integrated with data repositories such as Amazon S3, making it easier to process datasets with the Lustre file system.

> **Note**
> File system backups are not supported on file systems that are linked to a data repository. For more information, see Working with backups (p. 79).

Topics

Overview of data repositories

When you use Amazon FSx with a durable storage repository, you can ingest and process large volumes of file data in a high-performance file system. At the same time, you can periodically write intermediate results to your data repository. By using this approach, you can restart your workload at any time using the latest data stored in your data repository. When your workload is done, you can write final results from your file system to your data repository and delete your file system.

You can link your Amazon FSx file system to an Amazon S3 data repository when you create the file system. For more information, see Linking your file system to an S3 bucket (p. 22).

Amazon FSx 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 Amazon FSx file system. You can also run your compute-intensive workloads on Amazon EC2 instances in the Amazon Web Services Cloud and export the results to your data repository after your workload is complete.

In Amazon FSx for Lustre, you can export files and their associated metadata that you have written or modified in your file system to your durable data repository on Amazon S3 at any time. When you export a file or directory, your file system exports only data files and metadata that were created or modified since the last export or since file system creation. Such an export includes POSIX metadata.

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

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

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 durable data repository on Amazon S3. When you export changes in your file system to its linked data repository, Amazon FSx also exports POSIX metadata changes along with data changes. Because of this metadata export, you can implement and maintain access controls between your FSx for Lustre file system and its data repository on S3.

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

```
test/mydir/
test/
```

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

- S3 object key = `test/mydir/` maps to the Amazon FSx directory `test/mydir`.
- S3 object key = `test/` maps to the Amazon FSx directory `test`.

For symlinks, FSx for Lustre uses the same Amazon S3 schema as AWS DataSync, shown following:

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

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

- `Content-Type` – HTTP entity header used to indicate the media type of resource for web browsers.
- `x-amz-meta-file-permissions` – File type and permissions in format `<octal file type><octal permission mask>`, consistent with `st_mode` in Linux stat(2).

  > **Note**
  > FSx for Lustre does not import or retain `setuid` and `setgid` information.
- `x-amz-meta-file-owner` – The owner UID expressed as an integer.
- `x-amz-meta-file-group` – The group UID expressed as an integer.
- `x-amz-meta-file-atime` – The last accessed time in nanoseconds. Terminate the time value with "ns"; otherwise Amazon FSx interprets the value as milliseconds.
- `x-amz-meta-file-mtime` – The last modified time in nanoseconds. Terminate the time value with "ns"; otherwise, Amazon FSx interprets the value as milliseconds.
- `x-amz-meta-user-agent` – The user agent, ignored during Amazon FSx import. During export, Amazon FSx sets this value to `aws-fsx-lustre`.

The default POSIX permissions that FSx for Lustre assigns to a file are 755, which allows read and execute access for all users and write access for the owner of the file.

Note

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

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 (p. 142).

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 **Lifecycle state** on the **Summary** panel of the file system details page. The `Lifecycle` property is located in the `DataRepositoryConfiguration` object in the response of a `describe-file-system` CLI command (the equivalent API action is `DescribeFileSystems`).

Walkthrough: Attaching POSIX permissions when uploading objects into an 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

1. From your local computer or machine, create a test directory and file 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 UID and GID of 500 and permissions as shown in the example.

2. Call the S3 API to create the directory `s3cptestdir` with metadata permissions. You must specify the directory name with a trailing slash ("/"). For information on supported POSIX metadata, see POSIX metadata support for data repositories (p. 19).

   Replace *`bucket_name`* with the actual name of your S3 bucket.

```
$ aws s3api put-object --bucket bucket_name --key s3cptestdir/ --metadata '{"user-
agent":"aws-fsx-lustre" , \
      "file-atime":"1595002920000000000ns" , "file-owner":"500" , "file-
permissions":"0100664","file-group":"500" , \
      "file-mtime":"1595002920000000000ns"}'
```

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": "bAlhCoWq7aIEjc3R6Myc6UOb8sHHtJkR",
    "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.

```
$ aws s3 cp s3cptestdir/s3cptest.txt s3://bucket_name/s3cptestdir/s3cptest.txt \
      --metadata '{"user-agent":"aws-fsx-lustre" , "file-
atime":"1595002920000000000ns" , \
      "file-owner":"500" , "file-permissions":"0100664","file-group":"500" , "file-
mtime":"1595002920000000000ns"}'
```

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_JTlTyIx7Uzql6",
    "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                       bytes        Used   Available Use% Mounted on
3rnxfbmv-MDT0000_UUID       34.4G        6.1M       34.4G   0% /fsx[MDT:0]
3rnxfbmv-OST0000_UUID        1.1T        4.5M        1.1T   0% /fsx[OST: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` director and the `s3cptest.txt` file have POSIX permissions imported.

Linking your file system to an 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 S3 bucket that you will link to. In the **Create file system** wizard, you set the following data repository configuration properties in the optional **Data repository integration** 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 (p. 28).
- **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 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.
- **Export prefix** – Defines where Amazon FSx will export 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, you will need to specify an export path to the S3 bucket without any prefixes when creating your file system.

- When creating a file system using the console, choose the **Export prefix** > **A prefix you specify** option and leave the prefix field blank.
- When creating 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://lustre-export-test-bucket/`.

Using this method, you can include an Import prefix when specifying the Import path, and it will not impact a 1:1 mapping for exports.

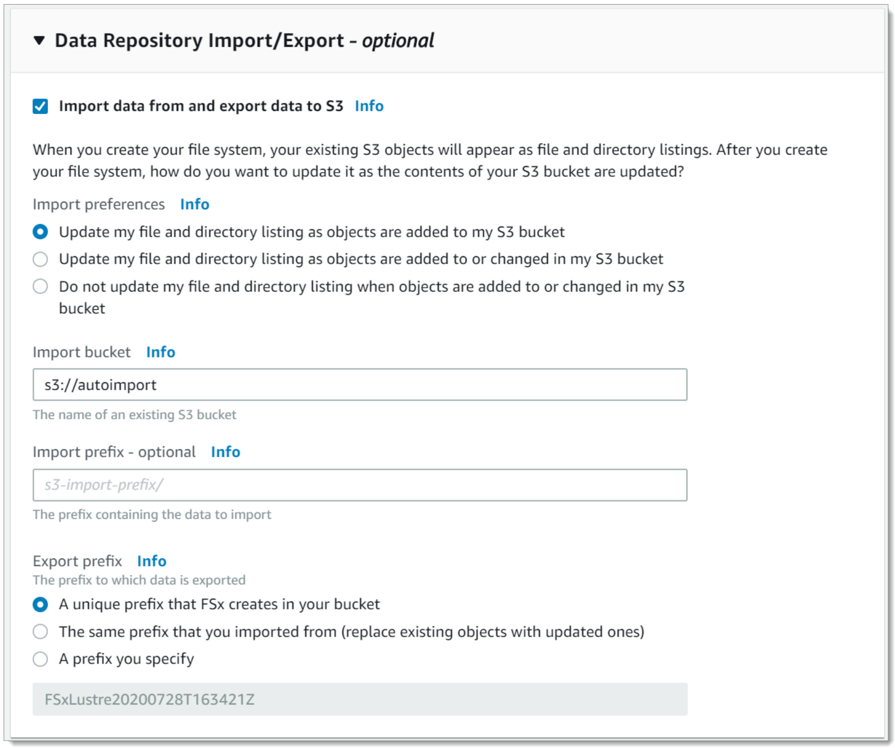
Creating file systems linked to an S3 bucket

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).

To create a file system linked to an S3 bucket (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/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 Amazon FSx for Lustre file system (p. 8).
5. 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 integration** section and configure the data repository settings.



▼ **Data Repository Import/Export** - *optional*

☑ **Import data from and export data to S3** Info

When you create your file system, your existing S3 objects will 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?

Import preferences Info

● Update my file and directory listing as objects are added to my S3 bucket
○ Update my file and directory listing as objects are added to or changed in my S3 bucket
○ Do not update my file and directory listing when objects are added to or changed in my S3 bucket

Import bucket Info

s3://autoimport

The name of an existing S3 bucket

Import prefix - optional Info

s3-import-prefix/

The prefix containing the data to import

Export prefix Info

The prefix to which data is exported

● A unique prefix that FSx creates in your bucket
○ The same prefix that you imported from (replace existing objects with updated ones)
○ A prefix you specify

FSxLustre20200728T163421Z

6. 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. After you create your file system, how do you want to update it as the contents of your S3 bucket are updated?

   - **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 import 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`, which is different from the default behavior when using the console.

   - **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.
   - **Do not update my file and directly listing when objects are added to or changed in my S3 bucket** - Amazon FSx only updates file and directory listing from the linked S3 bucket when the file system is created. FSx does not update file and directory listings for any new or changed objects after choosing this option.
7. 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 (p. 28).
8. Choose one of the three **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: /FSxLustre*file-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.

9. (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**.

To create a file system linked to an S3 bucket (AWS CLI)

The following example creates an Amazon FSx file system linked to the lustre-export-test-bucket, with an import preference that imports any new or changed 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-file-system, as shown following. The corresponding API operation is CreateFileSystem.

```
$ aws fsx create-file-system \
      --client-request-token CRT1234 \
      --file-system-type LUSTRE \
      --file-system-type-version 2.12 \
      --lustre-configuration
 AutoImportPolicy=NEW_CHANGED,DeploymentType=PERSISTENT_1,ImportPath=s3://lustre-
export-test-bucket/,ExportPath=s3://lustre-export-test-bucket/export \
      --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": "owner-id-string",
            "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",
                "DataRepositoryConfiguration": {
                    "AutoImportPolicy": "NEW_CHANGED",
                    "Lifecycle": "UPDATING",
                    "ImportPath": "s3://lustre-export-test-bucket/",
                    "ExportPath": "s3://lustre-export-test-bucket/export",
                    "ImportedFileChunkSize": 1024
                }
            }
        }
    ]
}
```

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 Enabling Amazon S3 default bucket encryption 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, *bucket_name*.

```
{
    "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/*"
        }
    }
}
```

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

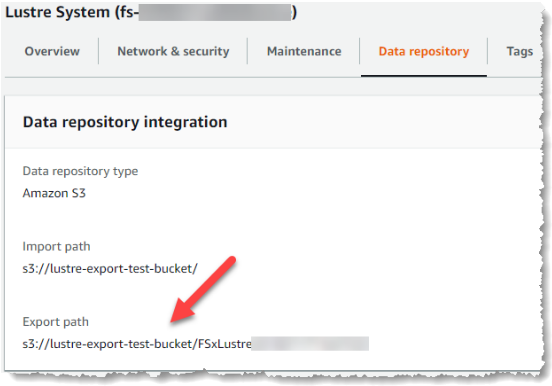
```
{
    "Sid": "Allow access through S3 for the FSx SLR to use the KMS key on the objects in
 the given S3 bucket",
    "Effect": "Allow",
    "Principal": {
        "AWS": "*"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",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

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.

To view a file system's export path (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Choose **File system name** or **File system ID** for the FSx for Lustre file system that you want to view the export path for. 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.



Lustre System (fs-[redacted])

Data repository integration

Data repository type
Amazon S3

Import path
s3://lustre-export-test-bucket/

Export path
s3://lustre-export-test-bucket/FSxLustre[redacted]

To view a file system's export path (CLI)

- To determine the export path for your file system, use the `describe-file-systems` 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": 3600,
    "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": {
        "WeeklyMaintenanceStartTime": "6:09:30",
        "DataRepositoryConfiguration": {
            "AutoImportPolicy": "NEW_CHANGED",
            "Lifecycle": "AVAILABLE",
            "ImportPath": "s3://lustre-export-test-bucket/",
    =======>"ExportPath": "s3://lustre-export-test-bucket/FSxLustre20190717T164753Z",
            "ImportedFileChunkSize": 1024
        }
    }
},
```

Importing files from your data repository

When you create a new file system linked to a data repository, Amazon FSx automatically imports the file metadata (the name, ownership, timestamp, and permissions) of the objects in your repository, such as an Amazon S3 bucket. Amazon FSx makes them visible as new files and directories in the FSx for Lustre file system. If the object does not include metadata, then Amazon FSx uses the default permissions of `root root 755`.

You can also configure your file system to automatically import file metadata for new or changed objects from your S3 bucket after creation. For more information, see Automatically import updates from your S3 bucket (p. 28).

Amazon FSx transparently copies the content of a file from your repository and loads it into the file system when your application first accesses the file in FSx. You can also preload your whole file system or an entire directory within your file system, for more information, see Preloading files into your file system (p. 31). If you request the preloading of multiple files simultaneously, Amazon FSx loads your files from your Amazon S3 data repository in parallel.

If you have a large number of files to import, it affects the amount of time it takes for Amazon FSx to create your file system.

Amazon FSx *only* imports S3 objects that have POSIX-compliant object keys, such as these:

```
test/mydir/
test/
```

Amazon FSx automatically copies file data for a given file from the linked durable data repository into your file system the first time you open that file. This data movement is managed by Amazon FSx and occurs transparently to your applications. Subsequent reads of these files are served directly out of the Amazon FSx file system with consistent sub millisecond latencies.

Automatically import updates from your S3 bucket

By default, when you create a new file system, Amazon FSx automatically 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 or changed in 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 only 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 preferences 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 will 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 setting when using the console to create a file system) 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 import 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.

- **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.
- **Do not update my file and directly listing when objects are added to or changed in my S3 bucket** – (Default setting when using the CLI or API to create a file system) Amazon FSx only updates file and directory listing from the linked S3 bucket when the file system is created. FSx does not update file and directory listings for any new or changed objects after choosing this option.

> **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.

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 will prevent 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.

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

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

Either of these conditions to will cause the data repository lifecycle state to become **Misconfigured**. For more information, see Data repository lifecycle state (p. 20).

It is a best practice to periodically sweep your linked S3 bucket and compare changes to the list of files on your file system, especially if your application requires a guarantee around importing changes.

Prerequisites

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

- The file system and its linked S3 bucket are located in the same AWS Region.
- The S3 bucket does not have a misconfigured **Lifecycle state**. For more information, see Data repository lifecycle state (p. 20).
- Your account has 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 S3 bucket (p. 22).

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 procedures.

To update import preferences on an existing file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/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 (p. 20).
5. Choose **Actions**, and then choose **Update import preferences** to display the **Update import preferences** dialog box.
6. Choose the new setting, and then choose **Update** to make the change.

To update import preferences on an existing file system (CLI)

- To update import preferences, use the `update-file-system` CLI command. The corresponding API operation is `UpdateFileSystem`.

```
$ aws fsx update-file-system \
    --file-system-id fs-0123456789abcdef0 \
    --lustre-configuration AutoImportPolicy=NEW_CHANGED
```

After successfully updating the file system's `AutoImportPolicy`, Amazon FSx returns the description of the updated file system as JSON, as shown in the following example.

```
{

    "FileSystems": [
        {
            "OwnerId": "111122223333",
            "CreationTime": 1549310341.483,
            "FileSystemId": "fs-0123456789abcdef0",
            "FileSystemType": "LUSTRE",
            "Lifecycle": "UPDATING",
            "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": {
                "WeeklyMaintenanceStartTime": "2:04:30",
                "DeploymentType": "PERSISTENT_1",
                "DataRepositoryConfiguration": {
                    "AutoImportPolicy": "NEW_CHANGED",
                    "Lifecycle": "UPDATING",
                    "ImportPath": "s3://lustre-export-test-bucket/",
                    "ExportPath": "s3://lustre-export-test-bucket/export",
                    "ImportedFileChunkSize": 1024
                }
            }
        }
    ]
}
```

Preloading files into your file system

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 `lfs` 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.

```
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.

```
nohup find local/directory -type f -print0 | xargs -0 -n 1 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 and throughput capacity (p. 89).

Data repository tasks

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

Data repository tasks optimize data and metadata transfers between your FSx for Lustre file system and its 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 GB/s. 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 data repository on S3. Files and associated permissions are stored in the same format that is used by other AWS services, including AWS DataSync, AWS Storage Gateway, and AWS Transfer for SFTP. Using the same format provides a consistent mechanism to control file access in AWS.

Types of data repository tasks

Currently, the **Export to repository** data repository task is the only task available. For more information about using this task to export from your Lustre file system to its linked data repository on S3, see Using data repository tasks to export data and metadata changes (p. 42).

Topics

Understanding a task's status and details

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 the failure. For more information about failed tasks, see Troubleshooting failed data repository tasks (p. 40).
- **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.

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` is the only type supported.
- 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 (p. 40).
- The time that the task started.
- The time that the task status was last updated. Task status is updated every 30 seconds.

For more information about accessing existing data repository tasks, see Accessing data repository tasks (p. 35).

Using data repository tasks

You can create, duplicate, view details, and cancel data repository tasks using the Amazon FSx console, CLI, or API.

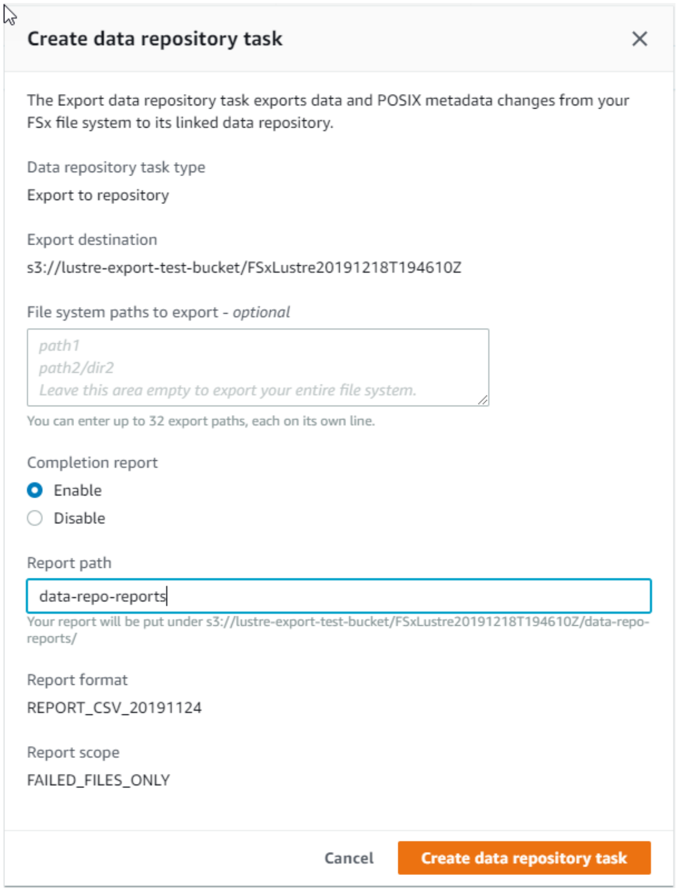
> **Note**
> We recommend that you download the latest AWS CLI so that you have access to all the required functionality. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*. Look for the **Upgrade** section for your operating system.

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.

To create a data repository task (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the navigation pane, choose **File systems**, then choose the Lustre file system that you want to create the task for.
3. For **Actions**, choose **Export to data repository**. This choice is not available if the file system isn't linked to a data repository on S3. The **Create data repository task** page appears.



Create data repository task

The Export data repository task exports data and POSIX metadata changes from your FSx file system to its linked data repository.

Data repository task type
Export to repository

Export destination
s3://lustre-export-test-bucket/FSxLustre20191218T194610Z

File system paths to export - *optional*

path1
path2/dir2
Leave this area empty to export your entire file system.

You can enter up to 32 export paths, each on its own line.

Completion report
● Enable
○ Disable

Report path
data-repo-reports

Your report will be put under s3://lustre-export-test-bucket/FSxLustre20191218T194610Z/data-repo-reports/

Report format
REPORT_CSV_20191124

Report scope
FAILED_FILES_ONLY

Data repository task type is set to **Export to repository**, which is the only task type currently supported. The **Export destination** value is the export prefix that you defined when you created the file system.

4. (Optional) Specify up to 32 directories or files to export from your Amazon FSx file system by providing the paths to those directories or files in **File system export paths**. 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.
5. (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 the file system's linked S3 data repository for **Report path**.
6. 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, choose **Data repository tasks (Lustre)** on the navigation pane. 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 create a data repository task (CLI)

The following procedure creates an export to repository task. Amazon FSx generates a task completion report after the task completes. If you don't want to generate a report, set `--report Enabled` to `false`. For more information about task completion reports, see Working with task completion reports (p. 40).

- To create a data repository task, use the `create-data-repository-task` CLI command. 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,Scope=FAILED_FILES_ONLY,Format=REPORT_CSV_20191124,Path=s3://
dataset-01/reports
```

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": "1545070680.240",
        "ClientRequestToken": "10192019-drt-12",
        "ResourceARN": "arn:aws:fsx:us-east-1:123456789012:task:task-123f8cd8e330c1321"
    }
}
```

After Amazon FSx begins processing the task, the task's status information becomes available. To view task details and status using the CLI, see To retrieve data repository tasks and task details (CLI) (p. 36).

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 data repository task** page. You can make changes to the paths to export, as needed, before creating and running the new task.

You can duplicate a task from the task details view or from the **Data repository tasks** page.

To duplicate an existing task

You can duplicate a task from the task details page or for the Data repository tasks page.

1. Choose a task on the **Data repository tasks (Lustre)** page.
2. Choose **Duplicate task**. The **Create data repository task** page appears. All settings for the new task are identical to those for the task that you're duplicating.
3. Change or add the paths that you want to export to. 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`.
4. Choose **Create data repository task** to create the task.

Accessing data repository tasks

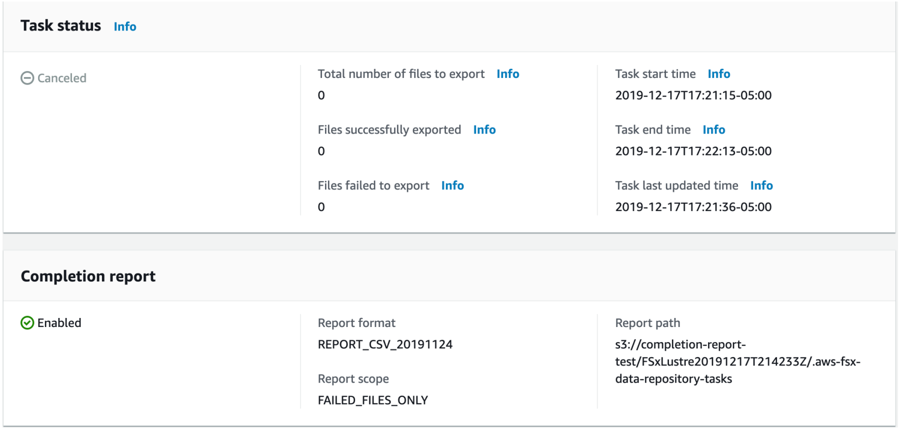
After you create a data repository task, you can access the task, and all existing tasks in your account, using the Amazon FSx console, CLI, and API. Amazon FSx provides the following detailed task information:

- All existing tasks.
- All tasks for a specific file system.
- All tasks with a specific lifecycle status. For more information about task lifecycle status values, see Understanding a task's status and details (p. 32).

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://console.aws.amazon.com/fsx/.
2. On the navigation pane, choose **Data repository tasks (Lustre)**. The **Data repository tasks** page appears, showing existing tasks.
3. To see a task's details, choose **Task ID** or **Task name** in the **Data repository tasks** page. The task detail page appears.



Task status Info

⊖ Canceled

Total number of files to export Info
0

Files successfully exported Info
0

Files failed to export Info
0

Task start time Info
2019-12-17T17:21:15-05:00

Task end time Info
2019-12-17T17:22:13-05:00

Task last updated time Info
2019-12-17T17:21:36-05:00

Completion report

✓ Enabled

Report format
REPORT_CSV_20191124

Report scope
FAILED_FILES_ONLY

Report path
s3://completion-report-test/FSxLustre20191217T214233Z/.aws-fsx-data-repository-tasks

To retrieve data repository tasks and task details (CLI)

Using the Amazon FSx `describe-data-repository-tasks` CLI command, you can view all the data repository tasks, and their details, in your account. `DescribeDataRepositoryTasks` 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.

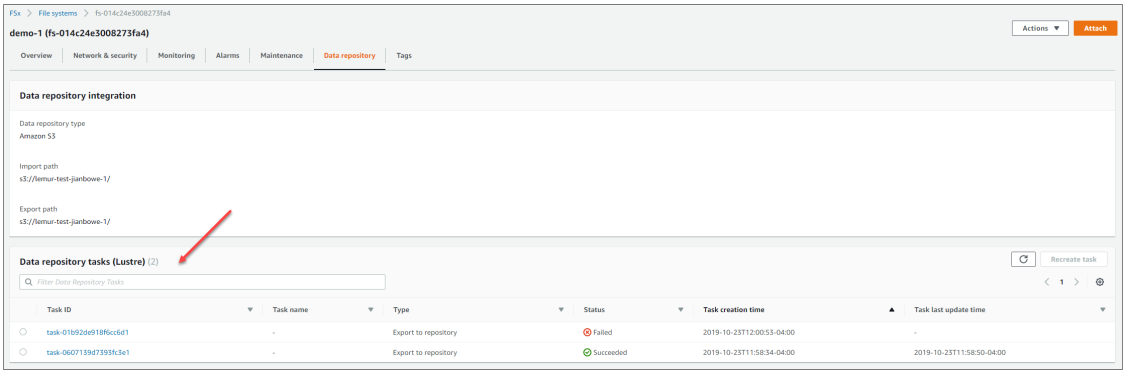
```
{
    "DataRepositoryTasks": [
        {
            "Lifecycle": "EXECUTING",
            "Paths": [],
            "Report": {
                "Path":"s3://dataset-01/reports",
                "Format":"REPORT_CSV_20191124",
                "Enabled":true,
                "Scope":"FAILED_FILES_ONLY"
            },
            "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.



demo-1 (fs-014c24e3008273fa4)

Data repository integration

Data repository type
Amazon S3

Import path
s3://lemur-test-jianbowe-1/

Export path
s3://lemur-test-jianbowe-1/

Data repository tasks (Lustre) (2)

	Task ID	Task name	Type	Status	Task creation time	Task last update time
○	task-01b92de918f6cc6d1	-	Export to repository	Failed	2019-10-23T12:00:53-04:00	-
○	task-0607139d7393fc3e1	-	Export to repository	Succeeded	2019-10-23T11:58:34-04:00	2019-10-23T11:58:50-04:00

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://console.aws.amazon.com/fsx/.
2. On the navigation pane, choose **Data repository tasks (Lustre)**. The **Data repository tasks** page appears, displaying existing tasks.
3. Choose **Task ID** or **Task name** for the task that you want to cancel.
4. Choose **Cancel task** to cancel the task.
5. Enter the task ID to confirm the cancellation request.

To cancel a data repository task (CLI)

Use the Amazon FSx `cancel-data-repository-task` CLI command, to cancel a task. `CancelDataRepositoryTask` is the equivalent API command.

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

  ```
  aws fsx cancel-data-repository-task \
      --task-id fs-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 a data repository task. The report includes results for the files processed by the task that match the scope of the report. Currently, the only available scope is `FAILED_FILES_ONLY`.

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 path must be located within the file system's export path, chosen when the file system was created. You can specify whether to generate a report for a task by using the `Enabled` parameter.

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 RFC-4180 - Common Format and MIME Type 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 Troubleshooting failed data repository tasks (p. 40).

To learn how to enable a task completion report when creating a data repository task, see Creating a data repository task (p. 33).

Troubleshooting failed data repository tasks

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 (p. 35).

Amazon FSx also provides information about the specific files and directories that failed in a task 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 (p. 40).

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

Error Code	Explanation
`PathSizeTooLong`	The task path is too long. The number of characters in the path for an exported file or directory can't exceed 1,024. This character number includes the export prefix set for the file system, plus the path to the exported file or directory.
`FileSizeTooLarge`	The file size for the export is too large. The maximum file size that Amazon FSx can export is 5 TiB.
`S3AccessDenied`	Access was denied to Amazon S3. The Amazon FSx file system must have permission to perform the `s3:PutObject` operation to export to a linked data repository on S3. This permission is granted in the `AWSServiceRoleForFSxS3Access_`*`fs-0123456789abcdef0`* service-linked role. For more information, see Using service-linked roles for Amazon FSx for Lustre (p. 123). If your S3 bucket uses server-side encryption with AWS KMS key stored in AWS Key Management Service (SSE-KMS), you must follow the policy configurations in Working with server-side encrypted Amazon S3 buckets (p. 25). 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 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 information, see Controlling ownership of uploaded objects using S3 Object Ownership in the *Amazon S3 User Guide*. Because the **Export to repository** task requires data to flow outside a file system's VPC, this error can occur if the target repository has a bucket policy that contains one of the `aws:SourceVpc` or `aws:SourceVpce` IAM global condition keys.
`S3Error`	Amazon FSx encountered an S3-related error that wasn't `S3AccessDenied`.
`ResourceBusy`	Amazon FSx was unable to export the file because it was being modified by another client on the file system. You can retry the DataRepositoryTask after your workflow has finished writing to the file.
`InternalError`	An error occurred within the Amazon FSx file system. 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 management (HSM) commands to export the files and directories to the data

Error Code	Explanation
	repository on S3. For more information, see Exporting files using HSM commands (p. 44).

Exporting changes to the data repository

You can export data and metadata changes, including POSIX metadata, from your Amazon FSx file system to its linked data repository. Associated POSIX metadata includes ownership, permissions, and timestamps. To perform this export, you use a data repository task.

Data repository tasks optimize data transfer by tracking changes between your Amazon FSx file system and its linked data repository. Only files or directories with new or modified data or metadata are exported. By exporting this data, you can implement and maintain access controls between your FSx for Lustre file system and its linked durable data repository on Amazon S3. For more information, see Data repository tasks (p. 31).

> **Important**
> To ensure that Amazon FSx can export your data to your S3 bucket, it must be stored in a UTF-8 compatible format.

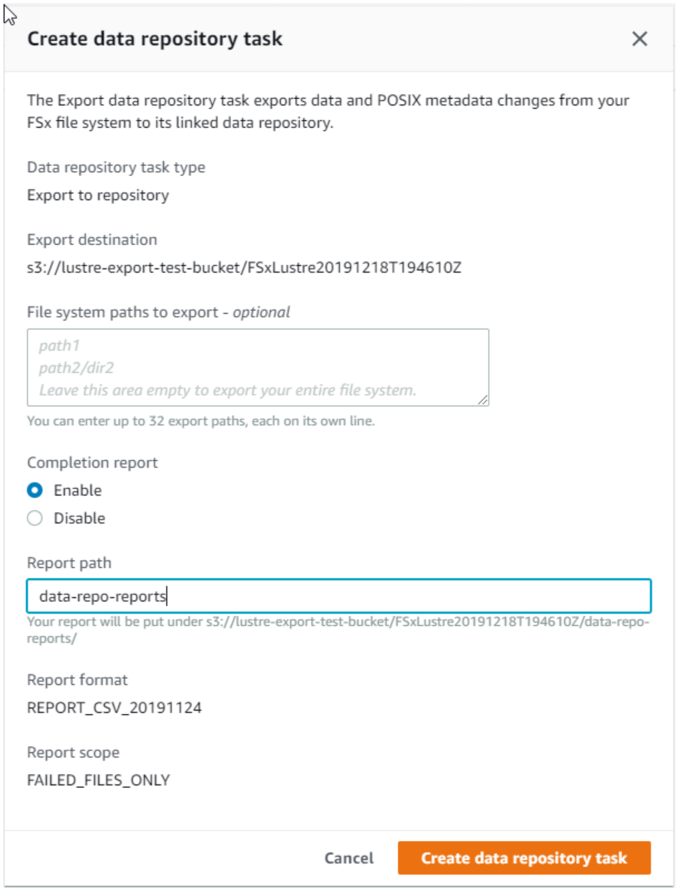
Topics

Using data repository tasks to export data and metadata changes

Use the following procedures to export data and metadata changes by using the Amazon FSx console and CLI.

To export changes (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. On the navigation pane, choose **File systems**, then choose the Lustre file system that you want to create the task for.
3. For **Actions**, choose **Export to data repository**. This choice is not available if the file system isn't linked to a data repository on S3. The **Create data repository task** page appears.



Create data repository task

The Export data repository task exports data and POSIX metadata changes from your FSx file system to its linked data repository.

Data repository task type
Export to repository

Export destination
s3://lustre-export-test-bucket/FSxLustre20191218T194610Z

File system paths to export - *optional*

path1
path2/dir2
Leave this area empty to export your entire file system.

You can enter up to 32 export paths, each on its own line.

Completion report
● Enable
○ Disable

Report path
data-repo-reports

Your report will be put under s3://lustre-export-test-bucket/FSxLustre20191218T194610Z/data-repo-reports/

Report format
REPORT_CSV_20191124

Report scope
FAILED_FILES_ONLY

Data repository task type is set to **Export to repository**, which is the only task type currently supported. The **Export destination** value is the export prefix that you defined when you created the file system.

4. (Optional) Specify up to 32 directories or files to export from your Amazon FSx file system by providing the paths to those directories or files in **File system export paths**. 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.

5. (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 the file system's linked S3 bucket for **Report path**.

6. 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, choose **Data repository tasks (Lustre)** on the navigation pane. 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 `create-data-repository-task` 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,Scope=FAILED_FILES_ONLY,Format=REPORT_CSV_20191124,Path=s3://
  dataset-01/reports
  ```

  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 on S3, you can check the status of the export. For more information about viewing data repository tasks, see Accessing data repository tasks (p. 35).

Exporting files using HSM commands

> **Note**
> To export changes in your FSx for Lustre file system's data and metadata to its durable data repository on Amazon S3, use the approach described in Using data repository tasks to export data and metadata changes (p. 42).

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 `NOOP` indicates that the file has successfully been exported.

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

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_action | grep
 "ARCHIVE" | wc -l
```

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

Releasing data from your file system

If you want to create storage space on your file system, you can release files from your file system. Releasing a file retains the file listing and metadata, but removes the local copy of that file's contents. You can release individual files from your file system using the following commands:

- To release one or more files from your file system if you are the file owner:

```
lfs hsm_release file1 file2 ...
```

- To release one or more files from your file system if you are not the file owner:

```
sudo lfs hsm_release file1 file2 ...
```

Using Amazon FSx with your on-premises data repository

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

To use Amazon FSx with your on-premises data

1. Create a file system. For more information, see Step 1: Create your Amazon FSx for Lustre file system (p. 8) in the getting started exercise.
2. Mount the file system from on-premises clients. For more information, see Mounting Amazon FSx file systems from on-premises or a peered Amazon VPC (p. 72).
3. Copy the data that you want to process into your Amazon FSx file system.
4. Run your compute-intensive workload on in-cloud Amazon EC2 instances mounting your file system. If you want to, you can periodically copy intermediate results to your data repository.
5. When you're finished, copy the final results from your file system back to your on-premises data repository, and delete your Amazon FSx file system.

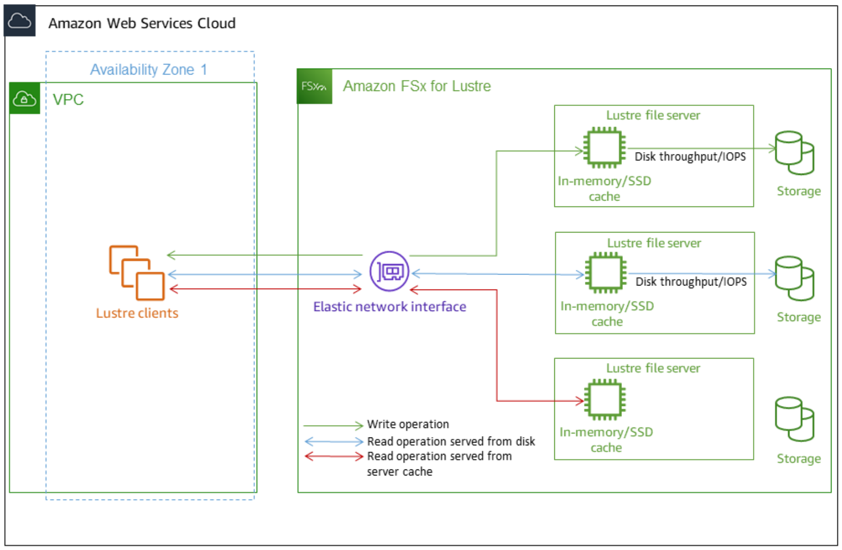
Amazon FSx for Lustre performance

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 Lustre's scalable architecture to support high levels of performance across large numbers of clients.

Topics

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, in-memory cache to enhance performance for the most frequently accessed data. HDD-based file systems can also be provisioned with an SSD-based read cache to further enhance performance for the most frequently accessed data. When a client accesses data that's stored in the in-memory or SSD cache, the file server doesn't need to read it 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 is not stored on the in-memory cache, file system performance is determined by the lower of the network throughput and disk throughput.

Aggregate file system performance

The throughput that an FSx for Lustre file system supports is proportional to its storage capacity. Amazon FSx for Lustre file systems scale to hundreds of GBps 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 Managing storage and throughput capacity (p. 89).

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 are designed for.

File system performance for HDD storage options

Deployment Type	Network throughput (MB/s per TiB of storage or SSD cache provisioned)		Network IOPS (IOPS per TiB of storage provisioned)	Cache storage (GiB per TiB of storage provisioned)	Disk latencies per file operation (milliseconds, P50)	Disk throughput (MB/s per TiB of storage or SSD cache provisioned)	
	Baseline	**Burst**				**Baseline**	**Burst**
PERSISTENT-12							
HDD storage	40	375*	Tens of thousands baseline Hundreds of thousands burst	0.4 memory	Metadata: sub-ms Data: single-digit ms	12	80 (read) 50 (write)
SSD read cache	200	1,900		200 SSD cache	Data: sub-ms	200	-
PERSISTENT-40							
HDD storage	150	1,300*	Tens of thousands baseline Hundreds of thousands burst	1.5 RAM	Metadata: sub-ms Data: single-digit ms	40	250 (read) 150 (write)
SSD read cache	750	6500		200 SSD cache	Data: sub-ms	200	-

File system performance for SSD storage options

Deployment Type	Network throughput (MB/s per TiB of storage provisioned)		Network IOPS (IOPS per TiB of storage provisioned)	Cache storage (GiB per TiB of storage provisioned)	Disk latencies per file operation (milliseconds, P50)	Disk throughput (MB/s per TiB of storage or SSD cache provisioned)	
	Baseline	Burst				Baseline	Burst
SCRATCH_2	200	1300	Tens of thousands baseline	6.7 RAM	Metadata: sub-ms Data: sub-ms	200 (read) 100 (write)	–
PERSISTENT-50	250	1,300*	Hundreds of thousands burst	2.2 RAM		50	240
PERSISTENT-100	500	1,300*		4.4 RAM		100	240
PERSISTENT-200	750	1,300*		8.8 RAM		200	240

Note

*Persistent file systems in the following AWS Regions provide network burst up to 530 MB/s 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).

Note

The FSx for Lustre SCRATCH_1 deployment option was designed to support approximately 200 MB/s/TiB.

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 MB/s per TiB of throughput per unit of storage provides an aggregate baseline disk throughput of 234 MB/s and a burst disk throughput of 1.125 GB/s.

Regardless of file system size, Amazon FSx for Lustre provides consistent, submillisecond latencies for file operations.

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 a single MDT and multiple OSTs. Each OST is approximately 1 to 2 TiB in size, depending on the file system's deployment type. 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                             bytes        Used   Available Use% Mounted on
mountname-MDT0000_UUID           68.7G        5.4M       68.7G   0% /fsx[MDT:0]
mountname-OST0000_UUID            1.1T        4.5M        1.1T   0% /fsx[OST:0]
mountname-OST0001_UUID            1.1T        4.5M        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.

- For Amazon FSx for Lustre file systems created before December 18, 2020, the default layout specifies a stripe count of one. 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 Amazon FSx for Lustre 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 five.
- For all Amazon FSx for Lustre 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 (p. 52).
- Stripe size by default is 1MiB. Setting a stripe offset may useful 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 -c8 -E -1 -c -1 /mountname/directory
```

This command sets three 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 -c8`) indicates a stripe count of 8 for files up to 10GiB in size.
- The third component (`-E -1 -c -1`) indicates that files larger than 10GiB will be striped across all OSTs.

> **Important**
> Appending data to a file created with a PFL layout will populate all of its layout components. For example, with the 3-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 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 December 18, 2020 is set with this command:

```
lfs setstripe -E 1G -c 1 -E -1 -c 5 /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 three-component example layout shown previously.

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 Monitoring Amazon FSx for Lustre (p. 101).

Performance tips

When using Amazon FSx for Lustre, keep the following performance tips in mind. For service limits, see Quotas (p. 138).

- **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.
- **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.

- **Workload balance across OSTs** – In some cases, your workload isn't driving the aggregate throughput that your file system can provide (200 MB/s 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, 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.

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 AWS 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 Getting started with Amazon FSx for Lustre (p. 8). Next, you can install and configure the Lustre client on your compute instance.

Topics

Installing the Lustre client

To mount your Amazon FSx for Lustre file system from a Linux instance, first install the open-source Lustre client. Amazon FSx for Lustre version 2.10 and version 2.12 both support access from the 2.10 versions of the Lustre client. Then, depending on your operating system version, use one of the following procedures.

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 Compiling Lustre on the Lustre Wiki.

Amazon Linux 2 and Amazon Linux

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. Do one of the following:

- If the command returns `4.14.104-95.84.amzn2.x86_64` for x86-based EC2 instances, or `4.14.181-142.260.amzn2.aarch64` or higher for AWS Graviton1- or Graviton2-powered EC2 instances, download and install the Lustre client with the following command.

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

- If the command returns a result less than `4.14.104-95.84.amzn2.x86_64` for x86-based EC2 instances, or less than `4.14.181-142.260.amzn2.aarch64` for AWS Graviton1- or Graviton2-powered EC2 instances, 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.

To install the Lustre client on Amazon Linux

1. Open a terminal on your client.
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 and Red Hat

To install the Lustre client on CentOS and Red Hat 7.5 or 7.6

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

   ```
   uname -r
   ```

3. Do one of the following:

   - If the instance is running kernel version `3.10.0-862.*`, download and install the Lustre 2.10.5 client with the following commands. The client comes in two packages to download and install.

```
sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.5/el7/
client/RPMS/x86_64/kmod-lustre-client-2.10.5-1.el7.x86_64.rpm
sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.5/el7/
client/RPMS/x86_64/lustre-client-2.10.5-1.el7.x86_64.rpm
```

- If the instance is running kernel version `3.10.0-957.*`, download and install the Lustre 2.10.8 client with the following commands. The client comes in two packages to download and install.

```
sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.8/el7/
client/RPMS/x86_64/kmod-lustre-client-2.10.8-1.el7.x86_64.rpm
sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.8/el7/
client/RPMS/x86_64/lustre-client-2.10.8-1.el7.x86_64.rpm
```

- If the instance is running kernel `3.10.0-1062.*` or greater, see To install the Lustre client on CentOS and Red Hat 7.7, 7.8, or 7.9 (x86_64 instances) (p. 57) for instructions on how to install the Lustre client from the Amazon FSx yum package repository.

Note
You might need to reboot your compute instance for the client to finish installing.

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.

```
curl https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-rpm-public-key.asc
 -o /tmp/fsx-rpm-public-key.asc
```

3. Import the key using the following command.

```
sudo rpm --import /tmp/fsx-rpm-public-key.asc
```

4. Add the repository and update the package manager using the following command.

```
sudo curl https://fsx-lustre-client-repo.s3.amazonaws.com/el/7/fsx-lustre-client.repo -
o /etc/yum.repos.d/aws-fsx.repo
```

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.

1. 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.

```
installonlypkgs=kernel, kernel-big#mem, kernel-enterprise, kernel-smp,
              kernel-debug, kernel-unsupported, kernel-source, kernel-devel, kernel-PAE,
              kernel-PAE-debug, kmod-lustre-client
```

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 Graviton1- and Graviton2-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.

```
curl https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-rpm-public-key.asc
 -o /tmp/fsx-rpm-public-key.asc
```

3. Import the key using the following command.

```
sudo rpm --import /tmp/fsx-rpm-public-key.asc
```

4. Add the repository and update the package manager using the following command.

```
sudo curl https://fsx-lustre-client-repo.s3.amazonaws.com/centos/7/fsx-lustre-
client.repo -o /etc/yum.repos.d/aws-fsx.repo
```

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.

1. 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.

```
installonlypkgs=kernel, kernel-big#mem, kernel-enterprise, kernel-smp,
              kernel-debug, kernel-unsupported, kernel-source, kernel-devel, kernel-PAE,
              kernel-PAE-debug, kmod-lustre-client
```

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 and newer

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 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.

   ```
   curl https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-rpm-public-key.asc
    -o /tmp/fsx-rpm-public-key.asc
   ```

3. Import the key by using the following command.

   ```
   sudo rpm --import /tmp/fsx-rpm-public-key.asc
   ```

4. Add the repository and update the package manager using the following command.

   ```
   sudo curl https://fsx-lustre-client-repo.s3.amazonaws.com/el/8/fsx-lustre-client.repo -
   o /etc/yum.repos.d/aws-fsx.repo
   ```

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 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.

1. 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 `4.18.0-305*`, you don't need to modify the repository configuration. Continue to the **To install the Lustre client** procedure.
   - If the command returns `4.18.0-240*`, 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 8.3, substitute *`specific_RHEL_version`* with `8.3` in the command.

   ```
   sudo sed -i 's#8#8.3#' /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 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.

```
installonlypkgs=kernel, kernel-PAE, installonlypkg(kernel), installonlypkg(kernel-module), 
              installonlypkg(vm), multiversion(kernel), kmod-lustre-client
```

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 16.04

You can get Lustre packages from the Ubuntu 16.04 Amazon FSx 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 public key. Use the following command.

    ```
    wget -O - https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-ubuntu-
    public-key.asc | sudo apt-key add -
    ```

    b. Add the Amazon FSx package repository to your local package manager. Use the following command.

    ```
    sudo bash -c 'echo "deb https://fsx-lustre-client-repo.s3.amazonaws.com/ubuntu
     xenial main" > /etc/apt/sources.list.d/fsxlustreclientrepo.list && apt-get update'
    ```

3. Determine which kernel is currently running on your client instance, and update as needed. The Lustre client on Ubuntu 16.04 requires kernel `4.4.0-1092-aws` or later.

    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 `4.4.0-1092-aws` 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 Amazon FSx for Lustre file system are installed. You can optionally install additional related packages, 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
```

Note

If you get a `Module Not Found error`, do the following:

Downgrade your kernel to the latest supported version. List all available versions of the lustre-client-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-4.4.0-1092-aws`, do the following:

1. Install the kernel this package was built for. Use the following commands.

```
sudo apt-get install -y linux-image-4.4.0-1092-aws
```

```
sudo sed -i 's/GRUB_DEFAULT=.\+/GRUB\_DEFAULT="Advanced options for
 Ubuntu>Ubuntu, with Linux 4.4.0-1099-aws"/' /etc/default/grub
```

```
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)
   ```

To install the Lustre client on Ubuntu 18.04

You can get Lustre packages from the Ubuntu 18.04 Amazon FSx 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.

   ```
   wget -O - https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-ubuntu-
   public-key.asc | sudo apt-key add -
   ```

   b. Add the Amazon FSx package repository to your local package manager using the following command.

   ```
   sudo bash -c 'echo "deb https://fsx-lustre-client-repo.s3.amazonaws.com/ubuntu
    bionic main" > /etc/apt/sources.list.d/fsxlustreclientrepo.list && apt-get update'
   ```

3. Determine which kernel is currently running on your client instance, and update as needed. The Lustre client on Ubuntu 18.04 requires kernel `4.15.0-1054-aws` or later for x86-based EC2 instances and kernel `5.3.0-1023-aws` or later for Arm-based EC2 instances powered by AWS Graviton2 processors.

   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 `4.15.0-1054-aws` for x86-based EC2 instances, or greater than `5.3.0-1023-aws` for Graviton2-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, 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
```

Note
If you get a `Module Not Found error`, do the following:

Downgrade your kernel to the latest supported version. List all available versions of the `lustre-client-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-4.15.0-1054-aws`, do the following:

1. Install the kernel this package was built for using the following commands.

```
sudo apt-get install -y linux-image-4.15.0-1054-aws
```

```
sudo sed -i 's/GRUB_DEFAULT=.\+/GRUB\_DEFAULT="Advanced options for
 Ubuntu>Ubuntu, with Linux 4.15.0-1054-aws"/' /etc/default/grub
```

```
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)
```

To install the Lustre client on Ubuntu 20.04

You can get Lustre packages from the Ubuntu 20.04 Amazon FSx 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.

```
wget -O - https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-ubuntu-
public-key.asc | sudo apt-key add -
```

   b. Add the Amazon FSx package repository to your local package manager using the following command.

      ```
      sudo bash -c 'echo "deb https://fsx-lustre-client-repo.s3.amazonaws.com/ubuntu
       focal main" > /etc/apt/sources.list.d/fsxlustreclientrepo.list && apt-get update'
      ```

3. Determine which kernel is currently running on your client instance, and update as needed. The Lustre client on Ubuntu 20.04 requires kernel `5.4.0-1011-aws` or later for x86-based EC2 instances and kernel `5.4.0-1015-aws` or later for Arm-based EC2 instances powered by AWS Graviton2 processors.

   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 `5.4.0-1011-aws` for x86-based EC2 instances, or greater than `5.4.0-1015-aws` for Graviton2-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 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
      ```

Note

If you get a `Module Not Found error`, do the following:

Downgrade your kernel to the latest supported version. List all available versions of the lustre-client-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
```

```
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.

```
sudo wget https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-sles-public-
key.asc
```

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.

```
sudo wget https://fsx-lustre-client-repo.s3.amazonaws.com/suse/sles-12/SLES-12/fsx-
lustre-client.repo
```

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.

```
sudo wget https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-sles-public-
key.asc
```

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.

```
sudo wget https://fsx-lustre-client-repo.s3.amazonaws.com/suse/sles-12/SLES-12/fsx-lustre-client.repo
```

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.

```
sudo wget https://fsx-lustre-client-repo-public-keys.s3.amazonaws.com/fsx-sles-public-key.asc
```

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.

```
sudo wget https://fsx-lustre-client-repo.s3.amazonaws.com/suse/sles-12/SLES-12/fsx-lustre-client.repo
```

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
   ```

   - 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.
2. Make a directory on your FSx for Lustre file system for the mount point with the following command.

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

3. 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 *`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.

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

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

   - `noatime` – Turns off updates to inode access times. To update inode access times, use the `mount` command without `noatime`.
   - `flock` – Enables file locking for your file system. If you don't want file locking enabled, use the `mount` command without `flock`.

4. 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    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% /fsx
   ```

```
tmpfs                              203956       0     203956    0% /run/user/1000
```

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

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:

1. 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 What is Amazon Elastic Container Service? in the *Amazon Elastic Container Service Developer Guide*.

We recommend using option 1 (Mounting from an Amazon EC2 instance hosting Amazon ECS tasks (p. 70)) 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 (Mounting from a Docker container (p. 71)), if you're unable to configure the EC2 instance, or if your application requires the container's flexibility.

> **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

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 (p. 55).

To mount your file system from Amazon ECS on an EC2 instance

1. 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 *`mountname`* 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 lustre2.10
mkdir -p "$fsx_mountpoint"
mount -t lustre ${fsx_dnsname}@tcp:/${fsx_mountname} ${fsx_mountpoint} -o noatime,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. 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 `mountname` with the file system's mount name.
   - Replace `mountpoint` with the file system's mount point.

```
"command": [
  "/bin/sh -c \"amazon-linux-extras install -y lustre2.10; mount -t
 lustre file_system_dns_name@tcp:/mountname mountpoint -o noatime,flock;\""
],
```

2. Add `SYS_ADMIN` capability to your container to authorize it to mount your FSx for Lustre file system, using the `linuxParameters` property.

```
"linuxParameters": {
  "capabilities": {
      "add": [
        "SYS_ADMIN"
      ]
   }
}
```

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 AWS 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 on-premises networks. For more information about using VPC transit gateways, see Getting Started with Transit Gateways in the *Amazon VPC Transit Gateways Guide*.

A *VPC peering connection* is a networking connection between two VPCs. This 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 What is VPC Peering? 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 created before 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 RFC 1918 private IP address range: • 10.0.0.0/8 • 172.16.0.0/12 • 192.168.0.0/16	✓
Peered networks using AWS Direct Connect or AWS VPN		✓

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 Working with backups (p. 79).

To retrieve the IP address of the primary network interface for a file system

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the navigation pane, choose **File systems**.
3. Choose your file system from the dashboard.
4. From the file system details page, choose **Network & security**.
5. For **Network interface**, choose the ID for your primary elastic network interface. Doing this takes you to the Amazon EC2 console.
6. 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 Amazon FSx for Lustre file system (p. 8) in the Getting Started exercise.

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.

```
file_system_dns_name@tcp:/mountname /fsx lustre defaults,noatime,flock,_netdev 0 0
```

Warning

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 Automatic mounting fails and the instance is unresponsive (p. 141).

3. Save the changes to the file.

Your EC2 instance is now configured to mount the Amazon FSx file system whenever it restarts.

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
`file_system_dns_name``@tcp:/`	The DNS name for your Amazon FSx file system, which identifies the file system. You can get this name from the console or programmatically 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 **describe-file-systems** command or the AWS API or SDK using the `DescribeFileSystems` 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: • `defaults` – This value tells the operating system to use the default mount options. You can list the default mount options after the file system has been mounted by viewing the output of the `mount` command. • `noatime` – This option turns off inode access time updates. If you want to update inode access times, remove this mount option. • `flock` – mounts your file system with file locking enabled. If you don't want file locking enabled, remove this mount option. • `_netdev` – The value tells the operating system that the file system resides on a device that requires network access. This option prevents the instance from mounting the file system until the network has been enabled on the client.
0	A value that indicates whether the file system should be backed up by `dump`. For Amazon FSx, this value should be `0`.

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/
```

2. 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.
- 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 Lustre documentation website.

Unmounting file systems

Before you delete a file system, we recommend that you unmount it from every Amazon EC2 instance that it's connected to. 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
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/sda1 ext4 8123812 1138920 6884644 15% /
```

```
$ umount /fsx
```

```
$ df -T
```

```
Filesystem Type 1K-blocks Used Available Use% Mounted on
/dev/sda1 ext4 8123812 1138920 6884644 15% /
```

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 Spot Instances in the *Amazon EC2 User Guide for Linux Instances*.

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 (p. 75).

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.

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 Best practices 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")
    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 "${FSXPATH}"; then
        echo "Error unmouting '$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 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 99.999999999% (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 block-level 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

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 File system deployment options (p. 15) and Using data repositories (p. 18).

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 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 `update-file-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 (p. 84).

Creating user-initiated backups

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

To create a user-initiated file system backup

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

You have now created your file system backup. You can find a table of all your backups in the Amazon FSx 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 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. Backups created by AWS Backup have backup type `AWS_BACKUP`. 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 Creating backup copies across AWS accounts in the *AWS Backup Developer Guide*.

Backup copy limitations

The following are some limitations when you copy backups:

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

Permissions for cross-Region backup copies

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

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

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

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

Full and incremental copies

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

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

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. In the navigation pane, choose **Backups**.
3. In the **Backups** table, choose the backup that you want to copy, and then choose **Copy backup**.
4. In the **Settings** section, do the following:

   - In the **Destination Region** list, choose a destination AWS Region to copy the backup to. The destination can be in another AWS Region (cross-Region copy) or within the same AWS Region (in-Region copy).
   - (Optional) Select **Copy Tags** to copy tags from the source backup to the destination backup. If you select **Copy Tags** and also add tags at step 6, all the tags are merged.
5. For **Encryption**, choose the AWS KMS encryption key to encrypt the copied backup.
6. For **Tags - optional**, enter a key and value to add tags for your copied backup. If you add tags here and also selected **Copy Tags** at step 4, all the tags are merged.
7. Choose **Copy backup**.

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

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

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

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

  ```
  aws fsx copy-backup \
  ```

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

The response shows the description of the copied backup.

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

Restoring backups

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

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

> **Note**
> You can only restore your backup to a file system of the same Lustre version type, deployment type, throughput per unit of storage, 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. For more information, see Managing storage and throughput capacity (p. 89).

To restore a file system from a backup

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

   Doing so opens the file system creation wizard. This wizard is identical to the standard file system creation wizard, except the file system configuration (e.g., Deployment Type, throughput per unit of storage). However, you can change the associated VPC, and backup settings.
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 that 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 https://console.aws.amazon.com/fsx/.
2. From the console dashboard, choose **Backups** from the left side navigation.

3. Choose the backup that you want to delete from the **Backups** table, and then choose **Delete backup**.
4. In the **Delete backups** dialog box that opens, confirm that the ID of the backup identifies the backup that you want to delete.
5. Confirm that the check box is checked for the backup that you want to delete.
6. Choose **Delete backups**.

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

Storage quotas

You can create storage quotas for users and groups 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 or a group can consume. Storage quotas automatically track user- and group-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

User and group 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 *block quota* limits the amount of disk space that a user or group can consume. You configure the storage size in kilobytes.
- An *inode quota* limits the number of files or directories that a user or group can create. You configure the maximum number of inodes as an integer.

> **Note**
> Default quotas and project quotas are not supported.

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 and group 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 quotas. You can also set different grace periods for a user quota and a group quota. When user and group quotas have different grace periods, the soft limit transforms to a hard limit after the grace period of either user or group quota 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 `lfs` commands in your Linux terminal. The `lfs setquota` command sets quota limits, and the `lfs quota` command displays quota information.

For more information about Lustre quota commands, see the *Lustre Operations Manual* on the Lustre documentation website.

Setting user and group quotas

The syntax of the `setquota` command for setting user or group quotas is as follows.

```
lfs setquota {-u|--user|-g|--group} username|groupname
             [-b block_softlimit] [-B block_hardlimit]
             [-i inode_softlimit] [-I inode_hardlimit]
             /mount_point
```

Where:

- `-u` or `--user` specifies a user to set a quota for.
- `-g` or `--group` specifies a group 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.

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
```

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 grace periods

The default grace period is one week. You can adjust the default grace period for users and groups, using the following syntax.

```
lfs setquota -t {-u|-g}
             [-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.
- `-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.

```
sudo lfs setquota -t -u -b 1000 -i 1w4d /mnt/fsx
```

Viewing quotas

The `quota` command displays information about user quotas, group 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.

View quota command	Quota information displayed
`lfs quota -u username /mount_point`	General quota information 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.
`lfs quota -g groupname /mount_point`	General quota information for a specific group.
`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.

Quotas and Amazon S3 linked buckets

You can link your FSx for Lustre file system to an Amazon S3 data repository when you create the file system. For more information, see Linking your file system to an S3 bucket (p. 22).

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 `hsm_release` command on 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 and throughput capacity

You can increase the 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.

You can track the storage optimization progress at any time using the Amazon FSx console, CLI, and API. For more information, see Monitoring storage capacity increases (p. 92).

Topics

Important points to know when increasing storage capacity

Here are a few important items to consider when increasing storage capacity:

- **Increase only** – You can only *increase* the amount of storage capacity for a file system; you cannot decrease storage capacity.
- **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, or until the storage optimization process has completed, whichever time is longer.
- **Throughput capacity** – The Amazon FSx console, the AWS CLI, and the Amazon FSx API don't allow you to specify a desired throughput level. However, 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.
- **Deployment type** – You can increase the storage capacity of all deployment types except for scratch 1 file systems. If you have a scratch 1 file system, you can create a new one with a larger storage capacity.

When to increase storage and throughput 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 (p. 101).

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 How to use Amazon FSx for Lustre metrics (p. 104). For information about how storage capacity affects throughput capacity, see Amazon FSx for Lustre performance (p. 46).

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`).

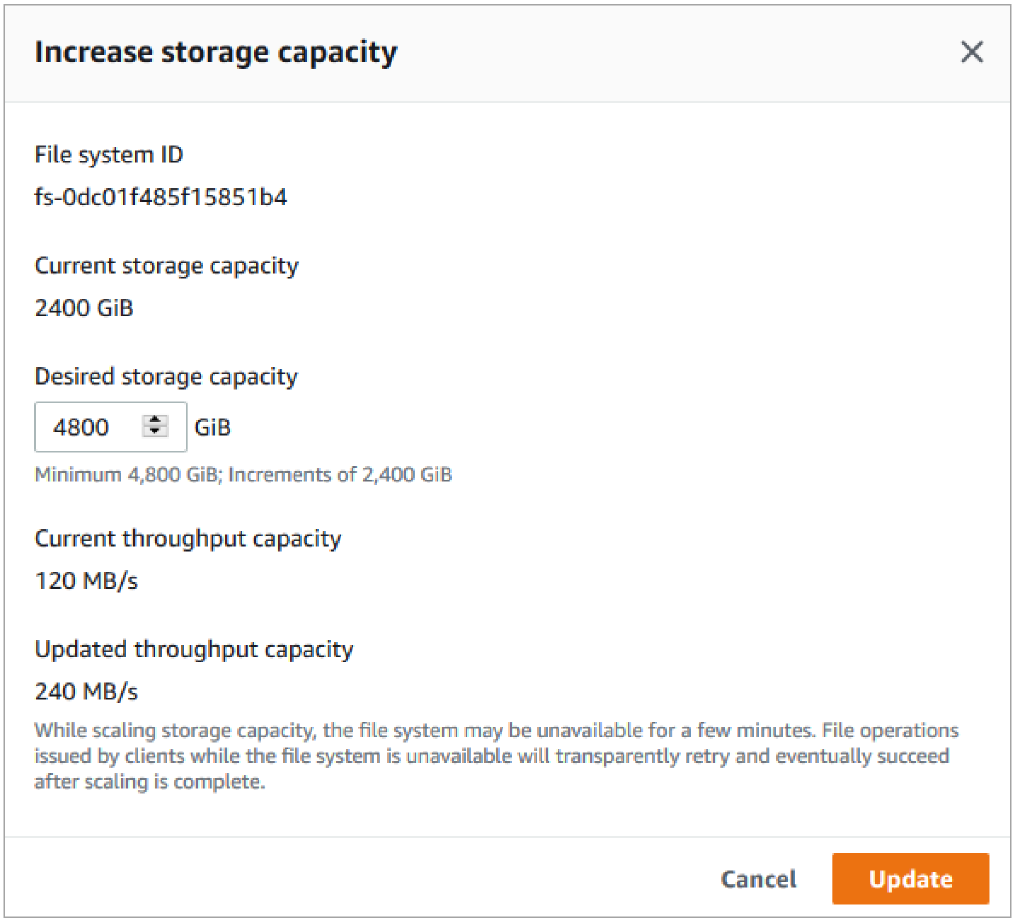
If a storage scaling request is pending and a file system backup request is also pending, the backup task has higher precedence. The storage scaling task does not start until the backup task is finished.

How to increase storage capacity

You can increase a file system's storage capacity using the Amazon FSx console, the AWS CLI, or the Amazon FSx API.

To increase storage capacity for a file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the Lustre file system that you want to increase storage capacity for.
3. 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.



Increase storage capacity

File system ID

fs-0dc01f485f15851b4

Current storage capacity

2400 GiB

Desired storage capacity

4800 GiB

Minimum 4,800 GiB; Increments of 2,400 GiB

Current throughput capacity

120 MB/s

Updated throughput capacity

240 MB/s

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 scaling is complete.

4. 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 MB/s/TiB file systems and multiples of 1800 GiB for 40 MB/s/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 update-file-system. Set the following parameters:

- Set `--file-system-id` to the ID of the file system you are updating.
- 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 MB/s/TiB file systems and multiples of 1800 for 40 MB/s/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
```

You can monitor the progress of the update by using the AWS CLI command describe-file-systems. Look for the `administrative-actions` in the output.

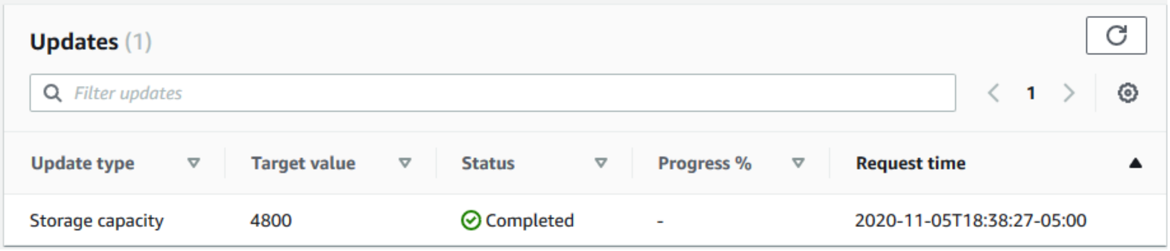
For more information, see AdministrativeAction.

Monitoring storage capacity increases

You can monitor the progress of a storage capacity increase using the Amazon FSx console, the API, or the AWS CLI.

Monitoring increases in the console

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



Updates (1)

Update type	Target value	Status	Progress %	Request time
Storage capacity	4800	✓ Completed	-	2020-11-05T18:38:27-05:00

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 describe-file-systems AWS CLI command and the DescribeFileSystems API action. The AdministrativeActions array lists the 10 most recent update actions for each administrative action type. When you increase a file system's storage capacity, two AdministrativeActions are generated: a FILE_SYSTEM_UPDATE and a STORAGE_OPTIMIZATION action.

The following example shows an excerpt of the response of a **describe-file-systems** CLI command. The file system has a storage capacity of 4800 GB, and there is a pending administrative action to increase the storage capacity to 9600 GB.

```
{
    "FileSystems": [
        {
            "OwnerId": "111122223333",
            .
            .
            .
            "StorageCapacity": 4800,
            "AdministrativeActions": [
                {
                     "AdministrativeActionType": "FILE_SYSTEM_UPDATE",
                     "RequestTime": 1581694764.757,
                     "Status": "PENDING",
                     "TargetFileSystemValues": {
                         "StorageCapacity": 9600
                     }
                },
                {
                    "AdministrativeActionType": "STORAGE_OPTIMIZATION",
                    "RequestTime": 1581694764.757,
                    "Status": "PENDING",
                }
            ]
```

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
                }
            ]
```

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.

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 MB/s per TiB of storage. Your disk throughput has a baseline of 50 MB/s per TiB. With data compression, your disk throughput could increase from 50 MB/s per TiB to a maximum of 250 MB/s 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 Aggregate file system performance (p. 47).

Topics

- Managing data compression (p. 95)
- Compressing previously written files (p. 97)
- Viewing file sizes (p. 97)
- Using CloudWatch metrics (p. 97)

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://console.aws.amazon.com/fsx/.
2. Follow the procedure for creating a new file system described in Step 1: Create your Amazon FSx for Lustre file system (p. 8) 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**.
6. 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 `create-file-system` with the `DataCompressionType` parameter, as shown following. The corresponding API operation is CreateFileSystem.

```
$ 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 \
      --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
            }
        }
    ]
}
```

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 (p. 97).

To update data compression on an existing file system (console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. Navigate to **File systems**, and choose the 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 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 `describe-backups` AWS CLI command (the equivalent API action is `DescribeBackups`).

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.
- `ls -l` 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 information about using metric math to calculate this ratio, see Metric math: Data compression ratio (p. 104).

For more information about monitoring your file system’s performance, see Monitoring Amazon FSx for Lustre (p. 101).

Tag your Amazon FSx 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

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. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper Tagging Best Practices.

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

If you're using the Amazon FSx 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 (p. 116).

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 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 Using cost allocation tags in the *AWS Billing and Cost Management 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.

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

Permissions and tag

For more information about the permissions required to tag Amazon FSx resources at creation, see Grant permission to tag resources during creation (p. 116). For more information about using tags to restrict access to Amazon FSx resources in IAM policies, see Using tags to control access to your Amazon FSx resources (p. 120).

Amazon FSx for Lustre maintenance windows

Amazon FSx for Lustre performs routine software patching for the Lustre software it manages. The maintenance window is your opportunity to control what day and time of the week this software patching occurs.

Patching occurs infrequently, typically once every several weeks. Patching should require only a fraction of your 30-minute maintenance window. During these few minutes of time, your file system will be temporarily unavailable.

You choose the maintenance window during file system creation. If you have no time preference, then a 30-minute default window is assigned.

You can use the Amazon FSx Management Console, AWS CLI 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://console.aws.amazon.com/fsx/.
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 use the AWS CLI or one of the AWS SDKs to change the maintenance window for your file systems using the UpdateFileSystem operation.

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
```

Monitoring Amazon FSx for Lustre

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

Monitoring with Amazon CloudWatch

You can monitor file systems using Amazon 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 web application or service is performing. By default, Amazon FSx for Lustre metric data is automatically sent to CloudWatch at 1-minute periods. For more information about CloudWatch, see What Is Amazon CloudWatch? in the *Amazon CloudWatch User Guide*.

CloudWatch metrics are reported as raw *Bytes*. Bytes are not rounded to either a decimal or binary multiple of the unit.

Amazon FSx for Lustre publishes the following metrics into the `FSx` namespace in CloudWatch. For each metric, Amazon FSx for Lustre emits a data point per disk per minute. To view aggregate file system details, you can use the `Sum` statistic. Note that the file servers behind your Amazon FSx for Lustre file systems are spread across multiple disks.

Metric	Description
`DataReadBytes`	The number of bytes for file system read operations. The `Sum` statistic is the total number of bytes associated with read operations during the period. The `Minimum` statistic is the minimum number of bytes associated with read operations on a single disk. The `Maximum` statistic is the maximum number of bytes associated with read operations on the disk. The `Average` statistic is the average number of bytes associated with read operations per disk. The `SampleCount` statistic is the number of disks. 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`, and `Average`. • Count for `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`
`DataWriteBytes`	The number of bytes for file system write operations. 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 disk. The `Maximum` statistic is the maximum number of bytes associated with write operations on the disk. The `Average` statistic is the average number of bytes associated with write operations per disk. The `SampleCount` statistic is the number of disks.

Metric	Description
	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`, and `Average`. - Count for `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`
`DataReadOperations`	The number of read operations. The `Sum` statistic is the total number of read operations. The `Minimum` statistic is the minimum number of read operations on a single disk. The `Maximum` statistic is the maximum number of read operations on the disk. The `Average` statistic is the average number of read operations per disk. The `SampleCount` statistic is the number of disks. 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: - Bytes for `Sum`, `Minimum`, `Maximum`, and `Average`. - Count for `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`
`DataWriteOperations`	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 disk. The `Maximum` statistic is the maximum number write operations on the disk. The `Average` statistic is the average number of write operations per disk. The `SampleCount` statistic is the number of disks. 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: - Bytes for `Sum`, `Minimum`, `Maximum`, and `Average`. - Count for `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`

Metric	Description
`MetadataOperations`	The number of metadata operations. The `Sum` statistic is the count of metadata operations. The `Minimum` statistic is the minimum number of metadata operations per disk. The `Maximum` statistic is the maximum number of metadata operations per disk. The `Average` statistic is the average number of metadata operations per disk. The `SampleCount` statistic is the number of disks. 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`, and `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`
`FreeDataStorageCapacity`	The amount of available storage capacity. The `Sum` statistic is the total number of bytes available in the file system. The `Minimum` statistic is the total number bytes available in the fullest disk. The `Maximum` statistic is the total number of bytes available in the disk with the most remaining available storage. The `Average` statistic is the average number of bytes available per disk. The `SampleCount` statistic is the number of disks. Units: • Bytes for `Sum`, `Minimum`, `Maximum`. • Count for `SampleCount`. Valid statistics: `Sum`, `Minimum`, `Maximum`, `Average`, `SampleCount`
`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 a disk in the file system. The `Maximum` statistic is the largest number of logical bytes stored in a disk in the file system. The `Average` statistic is the average number of logical bytes stored per disk. The `SampleCount` statistic is the number of disks. 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 disks in the file system. The Minimum statistic is the total number of bytes occupied in the emptiest disk. The Maximum statistic is the total number of bytes occupied in the fullest disk. The Average statistic is the average number of bytes occupied per disk. The SampleCount statistic is the number of disks. Units: • Bytes for Sum, Minimum, Maximum. • Count for SampleCount. Valid statistics: Sum, Minimum, Maximum, Average, SampleCount

Amazon FSx for Lustre dimensions

Amazon FSx for Lustre metrics use the FSx namespace and provide metrics for a single dimension, FileSystemId. A file system's ID can be found using the describe-file-systems AWS CLI command, and it takes the form of *fs-01234567890123456*.

How to use Amazon FSx for Lustre metrics

The metrics reported by Amazon FSx for Lustre provide information that you can analyze in different ways. The list following shows some common uses for the metrics. These are suggestions to get you started, not a comprehensive list.

How Do I Determine...	Relevant Metrics
My file system's throughput?	SUM(DataReadBytes + DataWriteBytes)/Period (in seconds)
My file system's IOPS?	Total IOPS = SUM(DataReadOperations + DataWriteOperations + MetadataOperations)/Period (in seconds)
My file system's data compression ratio?	SUM(LogicalDiskUsage) / SUM(PhysicalDiskUsage)

Metric math: Data compression ratio

Using metric math, you can query multiple CloudWatch metrics and use math expressions to create new time series based on these metrics. You can visualize the resulting time series in the CloudWatch console and add them to dashboards. For more information on metric math, see Use Metric Math in the *Amazon CloudWatch User Guide.*

This metric math expression calculates the data compression ratio of your Amazon FSx for Lustre file system. To calculate this ratio, first get the sum statistic of the total logical disk usage (without compression), which is provided by the `LogicalDiskUsage` metric. Then divide that by the sum statistic of the total physical disk usage (with compression), provided by the `PhysicalDiskUsage` metric.

So if your logic is this: sum of `LogicalDiskUsage` ÷ sum of `PhysicalDiskUsage`

Then your CloudWatch metric information is the following.

ID	Usable metric	Statistic	Period
m1	`LogicalDiskUsage`	Sum	1 minute
m2	`PhysicalDiskUsage`	Sum	1 minute

Your metric math ID and expression are the following.

ID	Expression
e1	`m1/m2`

`e1` is the data compression ratio.

Accessing CloudWatch metrics

You can see Amazon FSx for Lustre metrics for CloudWatch in many ways. You can view them through the CloudWatch console, or you can access them using the CloudWatch CLI or the CloudWatch API. The following procedures show you how to access the metrics using these various tools.

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, type its name in the search field.
5. (Optional) To filter by dimension, select **FileSystemId**.

To access metrics from the AWS CLI

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

To access metrics from the CloudWatch API

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

Creating CloudWatch alarms to monitor Amazon FSx for Lustre

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy.

Alarms invoke actions for sustained state changes only. CloudWatch alarms don't invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods.

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

To set alarms using the CloudWatch console

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Choose **Create Alarm**. Doing this launches the Create Alarm Wizard.
3. Choose **FSx Metrics** and scroll through the Amazon FSx for Lustre metrics to locate the metric that you want to place an alarm on. To display just the Amazon FSx for Lustre metrics in this dialog box, search on the file system ID of your file system. Choose the metric to create an alarm on, and choose **Next**.
4. Enter the **Name**, **Description**, **Whenever** values for the metric.
5. If you want CloudWatch to send you an email when the alarm state is reached, for **Whenever this alarm**, choose **State is ALARM**. For **Send notification to**, choose an existing SNS topic. If you choose **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in this box for future alarms.

   > **Note**
   > If you use **Create topic** to create a new Amazon SNS topic, verify the email addresses before sending them 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 don't receive a notification.
6. Preview the alarm you're about to create in the **Alarm Preview** area. If it appears as expected, choose **Create Alarm**.

To set an alarm using the AWS CLI

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

To set an alarm using the CloudWatch API

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

Security in 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 shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the Amazon Web Services Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon FSx for 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

The AWS shared responsibility model 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. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with 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 recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.

- 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 personal data that is stored in Amazon S3.
- If you require FIPS 140-2 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) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form 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 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

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 on encryption with Amazon FSx for Lustre, see these related topics:

- Create Your Amazon FSx for Lustre File System (p. 8)
- Amazon FSx for Lustre API permissions: actions, resources, and conditions reference (p. 117)

Topics

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 (p. 8).

> **Note**
> The AWS key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms. The infrastructure is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

For more information on how FSx for Lustre uses AWS KMS, see How Amazon FSx for Lustre uses AWS KMS (p. 109).

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 Cryptography Basics in the *AWS Key Management Service Developer Guide*.

Encrypting data in transit

Scratch 2 and persistent file systems automatically encrypt data in transit when they are accessed from Amazon EC2 instances that support encryption in transit. To learn which EC2 instances support encryption in transit, see Encryption in Transit in the *Amazon EC2 User Guide for Linux Instances*.

In-transit encryption of data for scratch 2 and persistent file systems is available in the following AWS Regions.

AWS Region	Scratch 2	Persistent
US East (Ohio)	✓	✓
US East (N. Virginia)	✓	✓
US West (Oregon), excluding US West (Los Angeles)	✓	✓
AWS GovCloud (US-West)	✓	✓
Europe (Ireland)	✓	✓
Europe (Frankfurt)	✓	
Asia Pacific (Singapore)	✓	

How Amazon FSx for Lustre uses AWS KMS

Amazon FSx for Lustre integrates with AWS Key Management Service (AWS KMS) for key management for encrypting data at rest. Amazon FSx uses AWS KMS keys to encrypt your file system in the following way:

- **Encrypting data at rest** – Amazon FSx for Lustre uses a KMS key, either the AWS managed key for Amazon FSx or a custom KMS key, to encrypt and decrypt file system data. All scratch FSx for Lustre file systems are encrypted at rest with keys managed by the service. Data is encrypted using an XTS-AES-256 block cipher. Data is automatically encrypted before being written to the file system, and is automatically decrypted as it is read. 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, either the AWS managed key for Amazon FSx or a custom KMS key. 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 AWS Key Management Service pricing.
- **Customer managed key** – This is the most flexible KMS key to use, because you can configure its key policies and grants for multiple users or services. For more information on creating customer managed keys, see Creating keys in the *AWS Key Management Service Developer Guide.*

  If you use a customer managed key as your KMS key for file data encryption and decryption, you can enable key rotation. When you enable key rotation, AWS KMS automatically rotates your key once per year. Additionally, with a customer managed key, you can choose when to disable, re-enable, delete, or revoke access to your customer managed key at any time.

> **Important**
> Amazon FSx accepts only symmetric KMS keys. You can't use asymmetric KMS keys with Amazon FSx.

Amazon FSx Key Policies for AWS KMS

Key policies are the primary way to control access to KMS keys. For more information on key policies, see Using key policies in AWS KMS in the *AWS Key Management Service Developer Guide.* The following list describes all the AWS KMS–related permissions supported by Amazon FSx for encrypted at rest file systems:

- **kms:Encrypt** – (Optional) Encrypts plaintext into ciphertext. This permission is included in the default key policy.
- **kms:Decrypt** – (Required) Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted. This permission is included in the default key policy.
- **kms:ReEncrypt** – (Optional) Encrypts data on the server side with a new KMS key, without exposing the plaintext of the data on the client side. The data is first decrypted and then re-encrypted. This permission is included in the default key policy.
- **kms:GenerateDataKeyWithoutPlaintext** – (Required) Returns a data encryption key encrypted under a KMS key. This permission is included in the default key policy under **kms:GenerateDataKey***.
- **kms:CreateGrant** – (Required) Adds a grant to a key to specify who can use the key and under what conditions. Grants are alternate permission mechanisms to key policies. For more information on grants, see Using grants in the *AWS Key Management Service Developer Guide.* This permission is included in the default key policy.
- **kms:DescribeKey** – (Required) Provides detailed information about the specified KMS key. This permission is included in the default key policy.
- **kms:ListAliases** – (Optional) Lists all of the key aliases in the account. When you use the console to create an encrypted file system, this permission populates the list 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.

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 APIs for performing administrative tasks and Lustre ports to interact with the file system.

Encrypting API traffic

To access the AWS published APIs, clients must support Transport Layer Security (TLS) 1.0. We recommend TLS 1.2 or above. 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 AWS Security Token Service (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 Encrypting data in transit (p. 109). FSx for Lustre does not natively offer encryption in transit between on-premise clients and file systems.

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.*

> **Warning**
> 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.

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 Security Group Rules in the *Amazon EC2 User Guide for Linux Instances.*

To create a security group for your Amazon FSx File System

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.

4. Specify a name and description for the security group.
5. For **VPC**, choose the 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.

Type	Protocol	Port Range	Source	Description
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group ID of the security group that you just created	Allows Lustre traffic between FSx for Lustre file servers
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security groups associated with your Lustre clients	Allows Lustre traffic between FSx for Lustre file servers and Lustre clients
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group ID of the security group that you just created	Allows Lustre traffic between FSx for Lustre file servers
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups associated with your Lustre clients	Allows Lustre traffic between FSx for Lustre file servers and 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.

Type	Protocol	Port Range	Source	Description
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group ID of the security group that you just created	Allow Lustre traffic between FSx for Lustre file servers
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security group associated with your Lustre clients	Allow Lustre traffic between FSx for Lustre file servers and Lustre clients
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group ID of the security group that you just created	Allows Lustre traffic between FSx for Lustre file servers
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups associated with your Lustre clients	Allows Lustre traffic between FSx for Lustre file servers and 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://console.aws.amazon.com/fsx/.
2. On the console dashboard, chose your file system to view its details.
3. On the **Network & Security** tab, choose your file system's network interface IDs (for example, `ENI-01234567890123456`). Doing this redirects you to the Amazon EC2 console.
4. Choose each network interface ID. Each action opens a new instance of the Amazon EC2 console in your browser. For each security group, choose **Change Security Groups** for **Actions**.
5. In the **Change Security Groups** dialog box, choose the security groups to use, and 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.

Type	Protocol	Port Range	Source	Description
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security groups that are applied to your Lustre clients	Allows Lustre traffic between Lustre clients
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security groups associated with your FSx for Lustre file systems	Allows Lustre traffic between FSx for Lustre file servers and Lustre clients
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups that are applied to your Lustre clients	Allows Lustre traffic between Lustre clients
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups associated with your FSx for Lustre file systems	Allows Lustre traffic between FSx for Lustre file servers and Lustre clients

Add the following outbound rules to the security groups applied to your Lustre clients.

Type	Protocol	Port Range	Source	Description
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security groups that are applied to your Lustre clients	Allows Lustre traffic between Lustre clients
Custom TCP rule	TCP	988	Choose **Custom** and enter the security group IDs of the security groups associated with your FSx for Lustre file systems	Allow Lustre traffic between FSx for Lustre file servers and Lustre clients

Type	Protocol	Port Range	Source	Description
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups that are applied to your Lustre clients	Allows Lustre traffic between Lustre clients
Custom TCP rule	TCP	1021-1023	Choose **Custom** and enter the security group IDs of the security groups associated with your FSx for Lustre file systems	Allows Lustre traffic between FSx for Lustre file servers and 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 File System Access Control with Amazon VPC (p. 111).

Resource administration access control with IAM for Amazon FSx for Lustre

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to AWS Identity and Access Management (IAM) identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM best practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

Amazon FSx for Lustre resources and operations

In Amazon FSx for Lustre, the primary resource is a *file system*. FSx for Lustre also supports the backup and data repository task subresources. You can create data repository tasks only in the context of an existing file system. You can create backups only in the context of an existing file system, or by copying an existing backup. These resources and subresources have unique Amazon Resource Names (ARNs) associated with them as shown in the following table.

Amazon FSx for Lustre provides a set of operations to work with Amazon FSx for Lustre resources. For a list of available operations, see the Amazon FSx for Lustre API Reference.

Understanding resource ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the root account, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a file system, your AWS account is the owner of the resource. In Amazon FSx for Lustre, the resource is the file system.
- If you create an IAM user in your AWS account and grant permissions to create a file system to that user, the user can create a file system. However, your AWS account, to which the user belongs, owns the file system resource.
- If you create an IAM role in your AWS account with permissions to create a file system, anyone who can assume the role can create a file system. Your AWS account, to which the role belongs, owns the file system resource.

Grant permission to tag resources during creation

Some resource-creating Amazon FSx for Lustre API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based access control (ABAC). For more information, What is ABAC for AWS in the *IAM User Guide*.

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `fsx:CreateFileSystem` or `fsx:CreateBackup`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `fsx:TagResource` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `fsx:TagResource` action.

The following example demonstrates a policy that allows users to create file systems and apply any tags to file systems during creation in a specific AWS account.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateFileSystem",
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/*"
    }
  ]
}
```

Similarly, the following policy allows users to create backups for a specific file system resource and apply any tags to the backup during backup creation.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "fsx:CreateBackup"
      ],
      "Resource": "arn:aws:fsx:region:account-id:file-system/file-system-id"
    },
    {
      "Effect": "Allow",
      "Action": [
         "fsx:TagResource"
      ],
      "Resource": "arn:aws:fsx:region:account-id:backup/*"
    }
  ]
}
```

The `fsx:TagResource` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `fsx:TagResource` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `fsx:TagResource` action.

Users require the `fsx:CreateBackup` permission when deleting an FSx for Lustre file system using the `DeleteFileSystem` API action or CLI command, and `SkipFinalBackup` is set to false. Otherwise, the action will fail because the user was not allowed to create a final backup before the file system is deleted.

For more information about tagging Amazon FSx resources, see Tag your Amazon FSx resources (p. 98). For more information about using tags to control access to FSx resources, see Using tags to control access to your Amazon FSx resources (p. 120).

Managing access to Amazon FSx resources

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

> **Note**
> This section discusses using IAM in the context of Amazon FSx for Lustre. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM policies) and policies attached to a resource are referred to as *resource-based* policies. Amazon FSx for Lustre supports only identity-based policies (IAM policies).

Amazon FSx for Lustre API permissions: actions, resources, and conditions reference

When you are setting up access control and writing a permissions policy that you can attach to an IAM identity (identity-based policies), you can use the following as a reference. The each Amazon FSx for Lustre API operation, the corresponding actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

You can use AWS-wide condition keys in your Amazon FSx for Lustre policies to express conditions. For a complete list of AWS-wide keys, see Available Keys in the *IAM User Guide*.

To specify an action, use the `fsx:` prefix followed by the API operation name (for example, `fsx:CreateFileSystem`). Each action applies to either a single Amazon FSx for Lustre file system, to all Amazon FSx for Lustre file systems owned by an AWS account, to a single backup, or to all backups owned by an AWS account.

This section only includes the Amazon FSx permissions required for these actions. Additional permissions from othe AWS services are required for some of these actions.

Amazon FSx for Lustre API and required permissions for actions

Amazon FSx for Lustre API operation	Required permissions (API actions)	Resource
CancelDataRepositoryTask	`fsx:CancelDataRepositoryTask`	`arn:aws:fsx:region:account-id:file-system/file-system-id`

CopyBackup	`fsx:CopyBackup`
	`fsx:CopyBackup`
	`fsx:TagResource`
		`arn:aws:fsx:region:account-id:backup/source-backup-id` – the source backup
		`arn:aws:fsx:region:account-id:backup/*` – the destination region
		`arn:aws:fsx:region:account-id:backup/*` – required to copy or create tags on the backup copy

CreateBackup	fsx:CreateBackup
	fsx:CreateBackup
	`fsx:TagResource`
		`arn:aws:fsx:region:account-id:backup/*`
		`arn:aws:fsx:region:account-id:file-system/file-system-id`
		`arn:aws:fsx:region:account-id:backup/*` – required to create tags on the new backup

CreateDataRepositoryTask	`fsx:CreateDataRepositoryTask`
	`fsx:CreateDataRepositoryTask`
	`fsx:TagResource`
		`arn:aws:fsx:region:account-id:file-system/file-system-id`
		`arn:aws:fsx:region:account-id:task/*`
		`arn:aws:fsx:region:account-id:task/*` – required to create tags on the task

CreateFileSystem	`fsx:CreateFileSystem` `fsx:TagResource`	`arn:aws:fsx:region:account-id:file-system/*`

Amazon FSx for Lustre API operation	Required permissions (API actions)	Resource
		`arn:aws:fsx:region:account-id:file-system/*` – to create tags on the file system

CreateFileSystemFromBackup	`fsx:CreateFileSystemFromBackup`
	`fsx:CreateFileSystemFromBackup`
	`fsx:TagResource`
		`arn:aws:fsx:region:account-id:file-system/*`
		`arn:aws:fsx:region:account-id:backup/*`
		`arn:aws:fsx:region:account-id:file-system/*` – to create tags on the file system

DeleteBackup	`fsx:DeleteBackup`	`arn:aws:fsx:region:account-id:backup/backup-id`

DeleteFileSystem	`fsx:DeleteFileSystem`
	`fsx:TagResource`
	`fsx:CreateBackup`
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		`arn:aws:fsx:region:account-id:backup/*` – required to create tags on a final backup if created
		`arn:aws:fsx:region:account-id:backup/*` – For Lustre file systems, required to create a final backup.

DescribeBackups	`fsx:DescribeBackups`	`arn:aws:fsx:region:account-id:backup/*`
DescribeDataRepositoryTasks	`fsx:DescribeDataRepositoryTasks`	`arn:aws:fsx:region:account-id:task/*`
DescribeFileSystems	`fsx:DescribeFileSystems`	`arn:aws:fsx:region:account-id:file-system/*`

ListTagsForResource	`fsx:ListTagsForResource`	`arn:aws:fsx:region:account-id:backup/backup-id`
		`arn:aws:fsx:region:account-id:file-system/filesystem-id`
		`arn:aws:fsx:region:account-id:task/task-id`

Amazon FSx for Lustre API operation	Required permissions (API actions)	Resource
TagResource	fsx:TagResource	arn:aws:fsx:*region*:*account-id*:backup/*backup-id* arn:aws:fsx:*region*:*account-id*:file-system/*filesystem-id* arn:aws:fsx:*region*:*account-id*:task/*task-id*
UntagResource	fsx:UntagResource	arn:aws:fsx:*region*:*account-id*:backup/*backup-id* arn:aws:fsx:*region*:*account-id*:file-system/*filesystem-id* arn:aws:fsx:*region*:*account-id*:task/*task-id*
UpdateFileSystem	fsx:UpdateFileSystem	arn:aws:fsx:*region*:*account-id*:file-system/*filesystem-id*

Using tags to control access to your Amazon FSx resources

To control access to Amazon FSx resources and actions, you can use AWS Identity and Access Management (IAM) policies based on tags. You can provide the control in two ways:

1. Control access to Amazon FSx resources based on the tags on those resources.
2. Control what tags can be passed in an IAM request condition.

For information about how to use tags to control access to AWS resources, see Controlling access using tags in the *IAM User Guide*. For more information about tagging Amazon FSx resources at creation, see Grant permission to tag resources during creation (p. 116). For more information about using tags, see Tag your Amazon FSx resources (p. 98).

Controlling access based on tags on a resource

To control what actions a user or role can perform on an Amazon FSx resource, you can use tags on the resource. For example, you might want to allow or deny specific API operations on a file system resource based on the key-value pair of the tag on the resource.

Example Example policy – Create a file system on when providing a specific tag

This policy allows the user to create a file system only when they tag it with a specific tag key value pair, in this example, `key=Department, value=Finance`.

```
{
    "Effect": "Allow",
    "Action": [
        "fsx:CreateFileSystem",
        "fsx:TagResource"
    ],
    "Resource": "arn:aws:fsx:region:account-id:file-system/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Department": "Finance"
        }
    }
}
```

Example Example policy – Create backups only on file systems with a specific tag

This policy allows users to create backups only on file systems that are tagged with the key value pair `key=Department, value=Finance`, and the backup will be created with the tag `Deparment=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:TagResource",
                "fsx:CreateBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }

    ]
}
```

Example Example policy – Create a file system with a specific tag from backups with a specific tag

This policy allows users to create file systems that are tagged with `Department=Finance` only from backups that are tagged with `Department=Finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateFileSystemFromBackup"
            ],
            "Resource": "arn:aws:fsx:region:account-id:backup/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Example Example policy – Delete file systems with specific tags

This policy allows a user to delete only file systems that are tagged with `Department=Finance`. If they create a final backup, then it must be tagged with `Department=Finance`. For Lustre file systems, users need the `fsx:CreateBackup privilege to create the final backup.`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:DeleteFileSystem"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateBackup",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id: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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateDataRepositoryTask"
            ],
            "Resource": "arn:aws:fsx:region:account-id:file-system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "fsx:CreateDataRepositoryTask",
                "fsx:TagResource"
            ],
            "Resource": "arn:aws:fsx:region:account-id:task/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Using service-linked roles for Amazon FSx for Lustre

Amazon FSx for Lustre uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon FSx for Lustre. Service-linked roles are predefined by Amazon FSx for Lustre and include all the permissions that the service requires to call other AWS services on your behalf.

Amazon FSx for Lustre defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon FSx for Lustre 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 for Lustre resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Amazon FSx for Lustre

Amazon FSx for Lustre 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.

For `AWSServiceRoleForAmazonFSx`, the role permissions policy allows Amazon FSx for Lustre to complete the following actions on the all applicable AWS resources:

- ec2:CreateNetworkInterface

For AWSServiceRoleForFSxS3Access_*fs-01234567890*, the role permissions policy allows Amazon FSx for Lustre to complete the following actions on your Amazon S3 bucket hosting your data repository.

- 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 for Lustre

You don't need to manually create a service-linked role. When you create a file system in the AWS Management Console, the IAM CLI, or the IAM API, Amazon FSx for Lustre creates the service-linked roles for you.

> **Important**
> The service-linked roles 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 these service-linked roles, and then need to create them again, you can use the same process to recreate the role in your account. When you create a file system, Amazon FSx for Lustre creates the service-linked role for you again.

> **Note**
> In order for Amazon FSx to create the AWSServiceRoleForFSxS3Access_fs-01234567890 service-linked role, your IAM entity will need to be allowed access to the following IAM actions, in addition to the AmazonFSxFullAccess managed policy:
>
> - iam:AttachRolePolicy
> - iam:PutRolePolicy

Editing a service-linked role for Amazon FSx for Lustre

Amazon FSx for Lustre 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 for Lustre

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 for Lustre 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 a service-linked role using IAM

Use the IAM console, the AWS 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 for Lustre service-linked roles

Amazon FSx for Lustre supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS Regions and Endpoints.

AWS managed policies for Amazon FSx

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ViewOnlyAccess** AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AmazonFSxServiceRolePolicy

You can't attach AmazonFSxServiceRolePolicy to your IAM entities. This policy is attached to a service-linked role that allows Amazon FSx to manage AWS resources on your behalf. For more information, see Using service-linked roles for Amazon FSx for Lustre (p. 123).

This policy grants administrative permissions that allows Amazon FSx to manage AWS reources on the user's behalf.

Permissions details

This policy includes the following permissions.

- `cloudwatch` – Allows Amazon FSx to publish metric data points to CloudWatch.
- `ds` – Allows Amazon FSx to view, authorize, and unauthorize applications in your AWS Directory Service directory.

- `ec2` – Allows Amazon FSx to do the following:
  - View, create, and disassociate network interfaces associated with an Amazon FSx file system.
  - View one or more Elastic IP addresses associated with an Amazon FSx file system.
  - View Amazon VPCs, security groups, and subnets associated with an Amazon FSx file system.
  - Create a permission for an AWS-authorized user to perform certain operations on a network interface.
- `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 Kinesis 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 Kinesis Data Firehose delivery stream.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ds:AuthorizeApplication",
                "ds:GetAuthorizedApplicationDetails",
                "ds:UnauthorizeApplication",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeAddresses",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVPCs",
                "ec2:DisassociateAddress",
                "route53:AssociateVPCWithHostedZone"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "AmazonFSx.FileSystemId"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssignPrivateIpAddresses",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:UnassignPrivateIpAddresses"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonFSx.FileSystemId": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:ReplaceRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/AmazonFSx": "ManagedByAmazonFSx"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/fsx/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:DescribeDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/aws-fsx-*"
        }
    ]
}
```

AWS managed policy: AmazonFSxFullAccess

You can attach AmazonFSxFullAccess to your IAM entities. Amazon FSx also attaches this policy to a service role that allows Amazon FSx to perform actions on your behalf.

Provides full access to Amazon FSx and access to related AWS services.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals full access to perform all Amazon FSx actions.
- `ds` – Allows principals to view information about the AWS Directory Service directories.

- `iam` – Allows principles to create an Amazon FSx service linked role on the user's behalf. This is required so that Amazon FSx can manage AWSresources on the user's behalf.
- `logs` – Allows principals to create log groups, log streams, and write events to log streams. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to CloudWatch Logs.
- `firehose` – Allows principals to write records to a Amazon Kinesis Data Firehose. This is required so that users can monitor FSx for Windows File Server file system access by sending audit access logs to Kinesis Data Firehose.
- `ec2` – Allows principals to create tags under the specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ds:DescribeDirectories",
                "fsx:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "s3.data-source.lustre.fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/fsx/*:log-group:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord"
            ],
            "Resource": [
                "arn:aws:firehose:*:*:deliverystream/aws-fsx-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonFSx": "ManagedByAmazonFSx"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["fsx.amazonaws.com"]
                }
            }
        }
    ]
}
```

AWS managed policy: AmazonFSxConsoleFullAccess

You can attach the `AmazonFSxConsoleFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow full access to Amazon FSx and access to related AWS services via the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to perform all actions in the Amazon FSx management console.
- `cloudwatch` – Allows principals to view CloudWatch Alarms in the Amazon FSx management console.
- `ds` – Allows principals to list information about an AWS Directory Service directory.
- `ec2` – Allows principals to create tags on route tables, list network interfaces, route tables, security groups, subnets and the VPC associated with an Amazon FSx file system.
- `kms` – Allows principals to list aliases for AWS Key Management Service keys.
- `s3` – Allows principals to list some or all of the objects in an Amazon S3 bucket (up to 1000).
- `iam` – Grants permission to create a service linked role that allows Amazon FSx to perform actions on the user's behalf.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "ds:DescribeDirectories",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "firehose:ListDeliveryStreams",
                "fsx:*",
                "kms:ListAliases",
                "logs:DescribeLogGroups",
                "s3:ListBucket"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "s3.data-source.lustre.fsx.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonFSx": "ManagedByAmazonFSx"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["fsx.amazonaws.com"]
                }
            }
        }
    ]
}
```

AWS managed policy: AmazonFSxConsoleReadOnlyAccess

You can attach the `AmazonFSxConsoleReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions to Amazon FSx and related AWS services so that users can view information about these services in the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `fsx` – Allows principals to view information about Amazon FSx file systems, including all tags, in the Amazon FSx Management Console.
- `cloudwatch` – Allows principals to view CloudWatch Alarms in the Amazon FSx Management Console.
- `ds` – Allows principals to view information about an AWS Directory Service directory in the Amazon FSx Management Console.
- `ec2` – Allows principals to view network interfaces, security groups, subnets and the VPC associated with an Amazon FSx file system in the Amazon FSx Management Console.
- `kms` – Allows principals to view aliases for AWS Key Management Service keys in the Amazon FSx Management Console.
- `log` – Allows principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.
- `firehose` – Allows principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "ds:DescribeDirectories",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "firehose:ListDeliveryStreams",
                "fsx:Describe*",
                "fsx:ListTagsForResource",
                "kms:DescribeKey",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon FSx updates to AWS managed policies

View details about updates to AWS managed policies for Amazon FSx since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon FSx Document history (p. 149) page.

Change	Description	Date
AmazonFSxServiceRolePolicy (p. 125) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to manage network configurations for Amazon FSx for NetApp ONTAP file systems.	September 2, 2021

Change	Description	Date
AmazonFSxFullAccess (p. 127) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021
AmazonFSxConsoleFullAccess (p. 129) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to create Amazon FSx for NetApp ONTAP Multi-AZ file systems.	September 2, 2021
AmazonFSxConsoleFullAccess (p. 129) – Update to an existing policy	Amazon FSx added new permisions to allow Amazon FSx to create tags on EC2 route tables for scoped down calls.	September 2, 2021
AmazonFSxServiceRolePolicy (p. 125) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to CloudWatch Logs log streams. This is required so that users can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021
AmazonFSxServiceRolePolicy (p. 125) – Update to an existing policy	Amazon FSx added new permissions to allow Amazon FSx to describe and write to Amazon Kinesis Data Firehose delivery streams. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Kinesis Data Firehose.	June 8, 2021
AmazonFSxFullAccess (p. 127) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe and create CloudWatch Logs log groups, log streams, and write events to log streams. This is required so that principals can view file access audit logs for FSx for Windows File Server file systems using CloudWatch Logs.	June 8, 2021

Change	Description	Date
AmazonFSxFullAccess (p. 127) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe and write records to a Amazon Kinesis Data Firehose. This is required so that users can view file access audit logs for an FSx for Windows File Server file system using Amazon Kinesis Data Firehose.	June 8, 2021
AmazonFSxConsoleFullAccess (p. 129) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can choose an existing CloudWatch Logs log group when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleFullAccess (p. 129) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can choose an existing Kinesis Data Firehose delivery stream when configuring file access auditing for an FSx for Windows File Server file system.	June 8, 2021
AmazonFSxConsoleReadOnlyAccess (p. 130) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon CloudWatch Logs log groups associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021

Change	Description	Date
AmazonFSxConsoleReadOnlyAccess (p. 130) – Update to an existing policy	Amazon FSx added new permissions to allow principals to describe the Amazon Kinesis Data Firehose delivery streams associated with the account making the request. This is required so that principals can view the existing file access auditing configuration for an FSx for Windows File Server file system.	June 8, 2021
Amazon FSx started tracking changes	Amazon FSx started tracking changes for its AWS managed policies.	June 8, 2021

Compliance Validation for Amazon FSx for Lustre

Third-party auditors assess the security and compliance of Amazon FSx for Lustre as part of multiple AWS compliance programs. These include SOC, PCI, ISO, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using Amazon FSx is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

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 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"
}
```

Quotas

Following, you can find out about quotas when working with Amazon FSx for Lustre.

Topics

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 file systems	100	The maximum number of Amazon FSx for Lustre persistent 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 storage capacity	100800	The maximum amount of storage capacity (in GiB) that you can configure for all Amazon FSx for Lustre persistent file systems in this account.
Lustre Scratch file systems	100	The maximum number of Amazon FSx for Lustre scratch file systems that you can 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 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 **Amazon FSx**.
4. Choose a quota.
5. Choose **Request quota increase**, and follow the directions to request a quota increase.
6. To view the status of the quota request, choose **Quota request history** in the console navigation pane.

For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Resource quotas for each file system

Following are the 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
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	200 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 Amazon FSx Endpoints and Quotas in the *AWS General Reference*.

Troubleshooting

Use the following information to help you resolve issues that you might encounter when working with Amazon FSx for Lustre.

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.

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 Amazon VPC Security Groups (p. 111).

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 Using DNS with Your VPC 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 Updating DNS Support for Your VPC 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 (p. 111).

You can't access your file system

There are a number of potential causes for being unable to access your file system, each with their own resolution, as follows.

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 (p. 111).

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. For more information, see Data repository lifecycle state (p. 20). A linked data repository can have a misconfigured lifecycle state under the following conditions:

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 Adding permissions to use data repositories in Amazon S3 (p. 7).
2. Using the Amazon FSx CLI or API, refresh the file system's `AutoImportPolicy` 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
```

For more information about service-linked roles, see Using service-linked roles for Amazon FSx for Lustre (p. 123).

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

1. 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:ObjectCreated:*` and `s3:ObjectRemoved:*`.
2. Please ensure that there is an S3 Event Notification Configuration in you linked S3 bucket with the name `FSx`, event types `s3:ObjectCreated:*` and `s3:ObjectRemoved:*`, 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
```

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 Adding permissions to use data repositories in Amazon S3 (p. 7).

For more information about service-linked roles, see Using service-linked roles for Amazon FSx for Lustre (p. 123).

Troubleshooting storage issues

In some cases, you may experience storage issues with your file system. You can troubleshoot these issues by using `lfs` commands, such as the `lfs 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 File system storage layout (p. 50). 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 Managing storage and throughput capacity (p. 89).

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 Unbalanced storage on OSTs (p. 144) section.

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.

Action to take

1. Use the `lfs df -h` command to determine the available capacity remaining on each storage target.
2. Use the `lfs find` command to identify files that can be migrated to other storage targets. For example, the following command returns every file that is larger than 1TB and has at least 1 object on `OST 2`. Any file that is striped across multiple OSTs and using `OST 2` will be returned.

   ```
   lfs find /mnt/lfs -size +1T --ost 2
   ```

3. Use the `lfs migrate` command to move files from one storage target to another storage target or to multiple storage targets. The following command shows an example of moving a file from the current storage target.

   ```
   lfs migrate /mnt/lfs/file1
   ```

   If you want to change the storage layout of the file, for example stripe a file from a single storage target to multiple storage targets (such as 4 storage targets), you can use a command similar to the following:

   ```
   lfs migrate -c 4 /mnt/lfs/file1
   ```

   If the file is very large, stripe the file across multiple storage targets. For more information, see Handling Full OSTs on the Lustre website.

   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 Striping data in your file system (p. 50).

Additional information

This section provides a reference of supported, but deprecated Amazon FSx features.

Topics

Setting up a custom backup schedule

We recommend using AWS Backup to set up a custom backup schedule for your file system. The information provided here is for reference purposes if you need to schedule backups more frequently than you can when using AWS Backup.

When enabled, Amazon FSx 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.

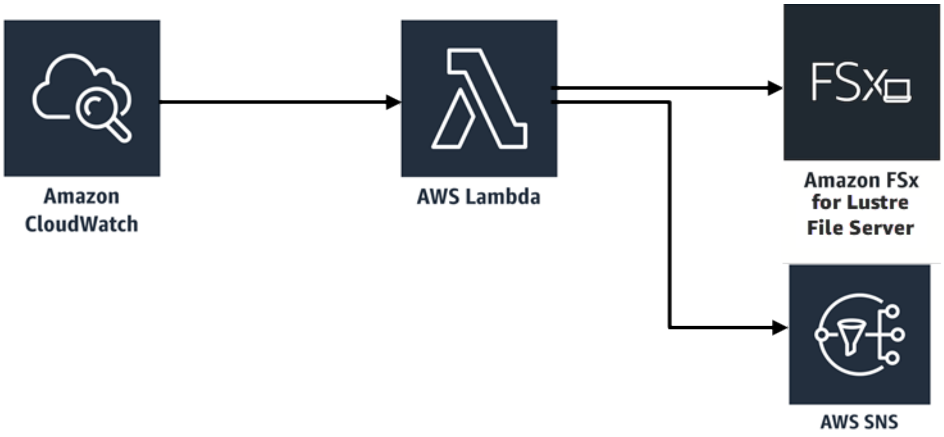
The solution automatically deploys all the components needed, and takes in the following parameters:

- 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 Schedule Expressions for Rules in the Amazon CloudWatch User Guide.

Architecture overview

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



This solution does the following:

1. The AWS CloudFormation template deploys an CloudWatch Event, a Lambda function, an Amazon SNS queue, and an IAM role. The IAM role gives the Lambda function permission to invoke the Amazon FSx 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.

AWS CloudFormation template

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

Automated deployment

The following procedure configures and deploys this custom backup scheduling solution. It takes about five minutes to deploy. Before you start, you must have the ID of an Amazon FSx 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 (p. 8).

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

To launch the custom backup solution stack

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

   > **Note**
   > By default, this template launches in the US East (N. Virginia) AWS Region. Amazon FSx 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*.

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

Parameter	Default	Description
Amazon FSx 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, triggering a new backup and deleting old backups outside of the retention period.

Parameter	Default	Description
Backup retention (days)	7	The number of days to keep user-initiated backups. The Lambda function deletes user-initiated backups older than this number of days.
Name for backups	user-scheduled backup	The name for these backups, which appears in the **Backup Name** column of the Amazon FSx for Lustre Management Console.
Backup notifications	Yes	Choose whether to be notified when backups are successfully initiated. A notification is always sent if there's an error.
Email address	No default value	The email address to subscribe to the SNS notifications.

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

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

Additional options

You can use the Lambda function created by this solution to perform custom scheduled backups of more than one Amazon FSx 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 AWS 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 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 AWS CloudFormation stack. Thus, they aren't removed if you delete the stack.

Document history

- **API version:** 2018-03-01
- **Latest documentation update:** October 5, 2021

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.

update-history-change	update-history-description	update-history-date
Support added for Lustre version 2.12 (p. 149)	You can now choose Lustre version 2.12 when you create an FSx for Lustre file system. For more information, see Step 1: Create your Amazon FSx for Lustre file system.	October 5, 2021
Lustre client support for Centos and Red Hat Enterprise Linux (RHEL) 8.4 added (p. 149)	The FSx for Lustre client now supports Amazon EC2 instances running Centos and Red Hat Enterprise Linux (RHEL) 8.4. For more information, see Installing the Lustre client.	June 9, 2021
Support added for data compression (p. 149)	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 (p. 149)	You can now use Amazon FSx to copy backups within the same AWS account to another AWS Region (cross-Region copies) or within the same AWS Region (in-Region copies). For more information, see Copying backups.	April 12, 2021
Lustre client support for Lustre filesets (p. 149)	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 (p. 149)	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	December 17, 2020

	Systems from On-Premises or a Peered Amazon VPC.	
Lustre client support for Arm-based Centos 7.9 added (p. 149)	The FSx for Lustre client now supports Amazon EC2 instances running Arm-based Centos 7.9. For more information, see Installing the Lustre client.	December 17, 2020
Lustre client support for Centos and Red Hat Enterprise Linux (RHEL) 8.3 added (p. 149)	The FSx for Lustre client now supports Amazon EC2 instances running Centos and Red Hat Enterprise Linux (RHEL) 8.3. For more information, see Installing the Lustre client.	December 16, 2020
Support added for storage and throughput capacity scaling (p. 149)	You can now increase the storage and throughput capacity for existing FSx for Lustre file systems as your storage and throughput requirements evolve. For more information, see Managing storage and throughput capacity.	November 24, 2020
Support added for storage quotas (p. 149)	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 (p. 149)	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 information, see Using AWS Backup with Amazon FSx.	November 9, 2020
Support added for the HDD (hard disk drive) storage options (p. 149)	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 throughput-intensive workloads that typically have large, sequential file operations. For more information, see Multiple Storage Options.	August 12, 2020

Support for importing linked data repository changes into FSx for Lustre (p. 149)	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 Automatically import updates from the data repository.	July 23, 2020
Lustre client support for SUSE Linux SP4 and SP5 added (p. 149)	The FSx for Lustre client now supports Amazon EC2 instances running SUSE Linux SP4 and SP5. For more information, see Installing the Lustre client.	July 20, 2020
Lustre client support for Centos and Red Hat Enterprise Linux (RHEL) 8.2 added (p. 149)	The FSx for Lustre client now supports Amazon EC2 instances running Centos and Red Hat Enterprise Linux (RHEL) 8.2. For more information, see Installing the Lustre client.	July 20, 2020
Support for automatic and manual file system backups added (p. 149)	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 (p. 149)	Scratch file systems are designed for temporary storage and shorter-term processing of data. Persistent file systems are designed for longer-term storage and workloads. For more information, see FSx for Lustre Deployment Options.	February 12, 2020
Support for POSIX metadata added (p. 149)	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 (p. 149)	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 Transferring Data & Metadata Using Data Repository Tasks.	December 23, 2019

Additional AWS Region support added (p. 149)	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 (p. 149)	FSx for Lustre is now available in the Asia Pacific (Singapore) 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 (p. 149)	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
User-defined data export path support added (p. 149)	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 additional flexibility to incorporate 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 (p. 149)	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 (p. 149)	Amazon FSx for Lustre is a fully managed file system that is optimized for compute-intensive workloads, such as high-performance computing, machine learning, and media processing workflows.	November 28, 2018