# Table of Contents

- What Is Amazon FSx for Lustre? ................................................................. 1
- Multiple Deployment Options ................................................................. 1
- Amazon FSx for Lustre and Data Repositories ....................................... 1
  - Amazon S3 Integration .................................................................... 1
  - On-Premises Data Repositories ....................................................... 1
- Accessing File Systems .......................................................................... 2
- Integrations with AWS Services .............................................................. 2
- Security and Compliance ...................................................................... 2
- Assumptions ........................................................................................... 3
- Pricing for Amazon FSx for Lustre .......................................................... 3
- Amazon FSx for Lustre Forums ............................................................... 3
- Are You a First-Time User of Amazon FSx for Lustre? .......................... 3
- Setting Up .............................................................................................. 4
  - Sign Up for AWS .............................................................................. 4
  - Create an IAM User ......................................................................... 4
  - Adding Permissions to Use Data Repositories in Amazon S3 .............. 6
- Next Step ............................................................................................... 6
- Getting Started ..................................................................................... 7
  - Prerequisites .................................................................................... 7
  - Step 1: Create Your Amazon FSx for Lustre File System .................... 7
  - Step 2: Install the Lustre Client ..................................................... 10
  - Step 3: Run Your Analysis ............................................................. 11
  - (Optional) Step 4: Check Amazon FSx File System Status ................. 11
  - Step 5: Clean Up Resources ......................................................... 12
- Using File System Deployment Options ............................................... 13
  - File System Deployment Options .................................................... 13
    - Scratch File Systems ................................................................... 13
    - Persistent File Systems ............................................................... 14
- Using Data Repositories ........................................................................ 16
  - Working with Server-Side Encrypted Amazon S3 Buckets .................. 16
  - POSIX Metadata Support .................................................................. 18
- Data Repository Tasks .......................................................................... 19
  - Types of Data Repository Tasks ....................................................... 19
  - Task Status and Details ................................................................. 19
  - Using Data Repository Tasks ......................................................... 20
  - Working with Task Completion Reports ......................................... 27
  - Troubleshooting Failed Tasks ......................................................... 28
- Exporting Data to Your Amazon S3 Bucket ........................................... 29
  - Setting the Export Prefix ................................................................ 29
  - Viewing a File System's Export Path .............................................. 31
  - Exporting Changed Data and Metadata ............................................ 32
- Using HSM Commands ......................................................................... 34
  - Importing Data .............................................................................. 34
  - Exporting Files to Your Data Repository Using HSM Commands ...... 35
- Using Amazon FSx with Your On-Premises Data Repository ............... 36
- Performance ......................................................................................... 37
  - How Lustre File Systems Work ....................................................... 37
  - Aggregate File System Performance ............................................... 37
  - File System Storage Layout ........................................................... 38
  - Striping Data in Your File System .................................................. 39
  - Monitoring Performance and Usage ............................................... 39
  - Performance Tips ............................................................................ 40
- Accessing File Systems ......................................................................... 41
  - Installing the Lustre Client ............................................................. 41
What Is Amazon FSx for Lustre?

Amazon 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. Amazon 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.

Amazon FSx for Lustre is POSIX-compliant, so you can use your current Linux-based applications without having to make any changes. Amazon 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.

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 Amazon FSx for Lustre (p. 13).

Amazon FSx for Lustre and Data Repositories

You can use Amazon FSx for Lustre to process cloud datasets stored for the longer-term on persistent file systems, Amazon S3, or your on-premises data repository using in-cloud compute instances.

Amazon S3 Integration

Amazon FSx for Lustre integrates with Amazon S3, making it easy for you to process cloud datasets using the Lustre high-performance file system. You can use Amazon FSx for Lustre to process cloud datasets stored in Amazon S3 by linking an S3 bucket to the file system. When linked to a durable data repository on Amazon S3, an Amazon FSx for Lustre file system transparently presents S3 objects as files. 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 Amazon FSx for Lustre file system and its durable data repository on Amazon S3. For more information, see Using Data Repositories with Amazon FSx for Lustre (p. 16) and Transferring Data and Metadata with Data Repository Tasks (p. 19).

On-Premises Data Repositories

With Amazon FSx for Lustre, you can burst your data processing workloads from on-premises into the AWS Cloud by importing data using AWS Direct Connect or AWS VPN. For more information, see Using Amazon FSx with Your On-Premises Data Repository (p. 36).
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.

Amazon 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. 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.

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. 41).

Integrations with AWS Services

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

Amazon 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.

Amazon 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 Amazon FSx for Lustre file systems or use existing file systems during the cluster creation process.

Security and Compliance

Amazon 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 Amazon FSx for Lustre, including AWS Regions where encryption of data in transit is supported, see Data Encryption in Amazon FSx for Lustre (p. 59). Amazon FSx has been assessed to comply with ISO, PCI-DSS, and
SOC certifications, and is HIPAA eligible. For more more information, see Security in Amazon FSx for Lustre (p. 58).

### 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. 62).

### 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. 7).
2. For information on performance, see Amazon FSx for Lustre Performance (p. 37).
3. For information on linking your file system to an Amazon S3 bucket data repository, see Using Data Repositories with Amazon FSx for Lustre (p. 16).
4. For Amazon FSx for Lustre security details, see Security in Amazon FSx for Lustre (p. 58).
5. For information on the scalability limits of Amazon FSx for Lustre, including throughput and file system size, see Quotas (p. 75).
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. 4)
2. Create an IAM User (p. 4)

Sign Up for AWS

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

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 Practices in the IAM User Guide.

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

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

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
Note
We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

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

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

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

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

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

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

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

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

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

To verify the sign-in link for IAM users for your account, open the IAM console and check under AWS Account Alias on the dashboard.
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 Amazon FSx for Lustre (p. 16).

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.

```json
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": ["iam:CreateServiceLinkedRole", "iam:AttachRolePolicy", "iam:PutRolePolicy"],
   }
}
```

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. 67).

Next Step

Getting Started with Amazon FSx for Lustre (p. 7)
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 (p. 7)
  • Step 1: Create Your Amazon FSx for Lustre File System (p. 7)
  • Step 2: Install and Configure the Lustre Client on Your Instance Before Mounting Your File System (p. 10)
  • Step 3: Run Your Analysis (p. 11)
  • (Optional) Step 4: Check Amazon FSx File System Status (p. 11)
  • Step 5: Clean Up Resources (p. 12)

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. 4).
• 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 Amazon 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, and newer 7.x versions, SUSE Linux Enterprise Server 12 SP3, and Ubuntu 16.04 and 18.04. For this getting started exercise, we recommend using CentOS 7.5, which is available in the AWS Marketplace.

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 Amazon 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 Amazon FSx for Lustre (p. 13).

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 information in the File system details section:

   • 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 type, choose Persistent or Scratch:
     - Choose Persistent 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 (AZ), 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. 60).
     - Choose 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.

   For more information, see File System Deployment Options for Amazon FSx for Lustre (p. 13).

   • For Storage capacity, provide a storage capacity for your file system, in TiB:
     - For a persistent or scratch 2 file system, this value can be 1.2 TiB or increments of 2.4 TiB.
     - For a scratch 1 file system, this value can be 1.2, 2.4, or increments of 3.6 TiB.
     - For a persistent file system, set a value for Throughput per unit of storage to either 50, 100, or 200 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 117 MB/s of file system throughput. You pay for the amount of throughput that you provision.
5. Provide networking and security group information in the **Network & security** section:
   - 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All TCP</td>
<td>TCP</td>
<td>0 - 65535</td>
<td>Custom the_ID_of_this_security_group</td>
<td>Inbound Lustre traffic rule</td>
</tr>
</tbody>
</table>

   The following screenshot shows an example of editing inbound rules.

   - 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](#).

7. (Optional) For **Data repository integration**, choose **Amazon S3** and specify the Amazon S3 bucket (with optional prefix) as the data repository source.

   Keep **Export prefix** at the default setting. For more information about the data repository integration, see [Using Data Repositories with Amazon FSx for Lustre](#).

   **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.

8. Choose **Review and create**.

9. 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 client.

**To download the Lustre client onto your Amazon EC2 instance**

1. Connect to your Amazon EC2 instance. For more information, see Connecting to Your Linux Instance from Windows Using PuTTY or Connecting to Your Linux Instance Using SSH in the Amazon EC2 User Guide for Linux Instances.
2. Open a terminal on your client.
3. Determine which kernel is currently running on the compute instance with the following command.

   ```
   uname -r
   ```
4. 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 2.10.6 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-2.10.6/el7/client/RPMS/x86_64/kmod-lustre-client-2.10.6-1.el7.x86_64.rpm
     sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.6/el7/client/RPMS/x86_64/lustre-client-2.10.6-1.el7.x86_64.rpm
     ```

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

**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 filename
```

For more information on how to run this command on a folder or large collection of files quickly, see Using Data Repositories with Amazon FSx for Lustre (p. 16).

### (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`.

<table>
<thead>
<tr>
<th>File System Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABLE</td>
<td>The file system is in a healthy state, and is reachable and available for use.</td>
</tr>
</tbody>
</table>
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.

   ```bash
   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, confirm that you want to delete your file system, and 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 Console User Guide.
Using Available Deployment Options for Amazon FSx for Lustre File Systems

Amazon 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 Amazon FSx for Lustre data striping, see Striping Data in Your File System (p. 39).

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

In this scenario, you store your datasets on the S3 data repository. When you create your Amazon 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 Amazon FSx for Lustre (p. 16) and Exporting Data to Your Amazon S3 Bucket (p. 29).

File System Deployment Options for Amazon FSx for Lustre

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

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. 37).

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.

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

**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 AZ.
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. 7) and CreateFileSystemLustreConfiguration in the Amazon FSx API Reference.
Using Data Repositories with Amazon FSx for Lustre

Amazon FSx for Lustre provides high-performance file systems optimized for fast workload processing. It can support workloads such as machine learning, high performance computing (HPC), video processing, financial modeling, and electronic design automation (EDA). These workloads commonly require data to be presented using a scalable, high-speed file system interface for data access. They typically have datasets stored on long-term durable data stores like Amazon S3, or on-premises storage.

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 durable data repository when you create the file system. For more information, see Step 1: Create Your Amazon FSx for Lustre File System (p. 7).

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. When you use Amazon FSx with a data repository, you can import your data into your Amazon FSx file system as needed. You can also run your compute-intensive workloads on Amazon EC2 instances in the AWS Cloud and export the results to your data repository after your workload is complete.

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 all linked file systems have been deleted.

**Topics**

- Working with Server-Side Encrypted Amazon S3 Buckets (p. 16)
- POSIX Metadata Support for Data Repositories (p. 18)
- Transferring Data and Metadata with Data Repository Tasks (p. 19)
- Exporting Data to Your Amazon S3 Bucket (p. 29)
- Using HSM Commands with Data Repositories (p. 34)
- Using Amazon FSx with Your On-Premises Data Repository (p. 36)

Working with Server-Side Encrypted Amazon S3 Buckets

Amazon FSx for Lustre supports Amazon S3 buckets that use server-side encryption with S3-managed keys (SSE-S3), and with Customer Master Keys (CMKs) 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 How Do I Enable Default Encryption for an Amazon S3 Bucket? in the Amazon Simple Storage Service Console User Guide.

When writing files to your S3 bucket, Amazon FSx will follow 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 KMS key policy that allows Amazon FSx to encrypt and decrypt objects in your S3 bucket using your Customer Master Key (CMK).

The following statement allows a specific Amazon FSx file system to encrypt and decrypt objects for a specific S3 bucket, `bucket-name`.

```json
```

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

```json
{   "Sid": "Allow access through S3 for the FSx SLR to use the KMS CMK 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/*"       }   } }
```
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. Amazon FSx also exports POSIX metadata changes, along with data changes, to the linked data repository on S3. Because of this metadata export, you can implement and maintain access controls between your Amazon FSx for Lustre file system and its data repository on S3.

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

- test/mydir/
- test/

Amazon FSx doesn't import S3 object keys that are not POSIX-compliant, such as the following.

- .
- ..
- test/.
- 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, Amazon 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**
Amazon 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.

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

Transferring Data and Metadata with Data Repository Tasks

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

Data repository tasks optimize data and metadata transfers between your Amazon 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 GiB/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 Amazon 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 Exporting Data to Your Amazon S3 Bucket (p. 29).

Topics
• Understanding a Task's Status and Details (p. 19)
• Using Data Repository Tasks (p. 20)
• Working with Task Completion Reports (p. 27)
• Troubleshooting Failed Data Repository Tasks (p. 28)

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. 28).
- 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. 27).
- 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. 23).

**Using Data Repository Tasks**

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

**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.
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. 27).

- 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. 23).

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 page or for 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. 19).

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.

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.

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

  If the command is successful, Amazon FSx returns the response in JSON format.
Using Data Repository Tasks

```json
{
  "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": {
```
Viewing Tasks by File System

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

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

To Retrieve Tasks by File System (CLI)
- Use the following command to view all data repository tasks for file system `fs-0123456789abcdef0`.

```bash
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.

```json
{
    "DataRepositoryTasks": [
        {
            "Lifecycle": "FAILED",
            "Paths": [],
            "Report": {
                "Path": "s3://dataset-04/reports",
                "Format": "REPORT_CSV_20191124"
            }
        }
    ]
}
```
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.

```bash
aws fsx cancel-data-repository-task --task-id fs-0123456789abcdef0
```

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

```json
{
  "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. 28).

To learn how to enable a task completion report when creating a data repository task, see Creating a Data Repository Task (p. 20).

**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. 23).

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. 27).

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

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathSizeTooLong</td>
<td>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.</td>
</tr>
<tr>
<td>FileSizeTooLarge</td>
<td>The file size for the export is too large. The maximum file size that Amazon FSx can export is 5 TiB.</td>
</tr>
<tr>
<td>S3AccessDenied</td>
<td>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_service-linked role. For more information, see Using Service-Linked Roles for Amazon FSx for Lustre (p. 67).</td>
</tr>
<tr>
<td>S3Error</td>
<td>Amazon FSx encountered an S3-related error that wasn't S3AccessDenied.</td>
</tr>
<tr>
<td>ResourceBusy</td>
<td>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.</td>
</tr>
<tr>
<td>InternalError</td>
<td>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 repository on S3. For more information, see Exporting Files to Your Data Repository Using HSM Commands (p. 35).</td>
</tr>
</tbody>
</table>

28
Exporting Data to Your Amazon S3 Bucket

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.

Setting the Export Prefix

You set the export path, or prefix, when you create your file system using the Amazon FSx console, CLI, or API. The export prefix specifies the path to export changed data and metadata to on your data repository on S3. You can't change the export prefix after the file system is created.

To Set the Export Prefix (Console)

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/.
2. From the dashboard, choose Create file system.
3. Choose Amazon FSx for Lustre for the file system type, 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. 7).
5. Expand the Data repository integration section and choose Amazon S3 for Data repository type. Specify the Amazon S3 Import bucket and an optional Import prefix as the data repository source.
6. 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 Amazon FSx for Lustre. The prefix looks like the following: /FSxLustre/file-system-creation-timestamp. The timestamp is in UTC format, for example FSxLustre20181105T222312Z. This option is the default.
   - 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.
7. (Optional) Set Maintenance preferences or use the system defaults.
8. Choose Next and review the file system settings.
9. Choose Create file system.
Setting the Export Path (AWS CLI)

You can specify an export path, or prefix, when you create your file system using the AWS CLI or the Amazon FSx API.

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

```bash
$ aws fsx create-file-system \
   --client-request-token CRT1234 \
   --file-system-type LUSTRE \
   --lustre-configuration 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.

```json
{
   "FileSystems": [
      {
         "OwnerId": "owner-id-string",
         "CreationTime": 1549310341.483,
         "FileSystemId": "fs-0123456789abcdef0",
         "FileSystemType": "LUSTRE",
         "Lifecycle": "CREATING",
         "StorageCapacity": 3600,
         "VpcId": "vpc-123456",
         "SubnetIds": [
            "subnet-123456"
         ],
         "NetworkInterfaceIds": [
            "eni-039fcf55123456789"
         ],
         "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
         "Tags": [
            {
               "Key": "Name",
               "Value": "Lustre-TEST-1"
            }
         ],
         "LustreConfiguration": {
            "WeeklyMaintenanceStartTime": "2:04:30",
            "DeploymentType": "PERSISTENT_1",
            "DataRepositoryConfiguration": {
               "ImportPath": "s3://lustre-export-test-bucket/",
               "ExportPath": "s3://lustre-export-test-bucket/export",
               "ImportedFileChunkSize": 1024
            }
         }
      }
   ]
}
```
Viewing a File System's Export Path

You can view a file system's export path using the Amazon 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 Amazon 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.

To View a File System's Export Path (CLI)

• To determine the export path for your file system, use the `aws fsx describe-file-systems` command.

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

Look for the ExportPath property under LustreConfiguration in the response.

```
{
    "OwnerId": "0123456789ab",
    "CreationTime": 1563382847.014,
    "FileSystemId": "",
    "FileSystemType": "LUSTRE",
    "Lifecycle": "AVAILABLE",
    "StorageCapacity": 3600,
    "VpcId": "vpc-6296a00a",
    "SubnetIds": [
        "subnet-1111111"
    ],
    "NetworkInterfaceIds": [
        "eni-0c288d5b88cc06c82d",
        "eni-0f38b702442c6918c"
    ],
    "DNSName": "fs-0123456789abcdef0.fsx.us-east-2.amazonaws.com",
    "Tags": [
        {
            "Key": "Name",
            "Value": "lfsx-0123456789abcdef0"
        }
    ]
}```
Exporting Changed Data and Metadata

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 Amazon FSx for Lustre file system and its linked durable data repository on Amazon S3. For more information, see Transferring Data and Metadata with Data Repository Tasks (p. 19).

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

**To Export Data and Metadata Changes (Console)**

1. Open the Amazon FSx console at [https://console.aws.amazon.com/fsx/](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.
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/efsx and /mnt/efsx/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 Export Data and Metadata Changes (CLI)

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

```bash
$ 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.

```json
{
  "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. 23).

Using HSM Commands with Data Repositories

You can use hierarchical storage management (HSM) commands to transfer data and metadata between your Amazon FSx for Lustre file system and a durable data repository. You can find a description of how to do this in the following sections.

Topics

- Importing Data from Your Amazon S3 Bucket (p. 34)
- Exporting Files to Your Data Repository Using HSM Commands (p. 35)

Importing Data from Your Amazon S3 Bucket

During file system creation, Amazon FSx automatically preloads a listing of file metadata into your file system. This file metadata includes POSIX metadata such as file name, size, modification time,
ownership, and permissions. Doing this allows clients to view the listing of files in your data repository as soon as your file system is available.

Importing data from your Amazon S3 data repository happens during file system creation. If you have a large number of files to import, this will impact the amount of time it takes for Amazon FSx to create your file system.

**Note**

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

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

Amazon FSx does not import S3 object keys that are not POSIX-compliant, such as these:

```
.
..
test/
test/..
```

Amazon FSx automatically copies file data for a given file from the linked Amazon S3 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 submillisecond latencies.

**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 latency penalty. If your application is sensitive to this latency and you know which file your application needs to access, you can optionally preload contents of an individual file. You do so using the `hsm_restore` command, as follows.

You can use the `hsm_action` command 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. 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 &
```

**Exporting Files to Your Data Repository Using HSM Commands**

**Note**

To export changes in your Amazon FSx for Lustre file system's data and metadata to its durable data repository on Amazon S3, use the approach described in Exporting Changed Data and Metadata (p. 32).

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.

```
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.

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

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

```bash
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, then the export is complete.

---

**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. 7) 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. 48).
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.

How Lustre File Systems Work

Each Amazon FSx for Lustre file system is comprised 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. When a client accesses data that's stored in the in-memory 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 cache:

When you read data that is stored on the file server's in-memory 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 Amazon 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).

Amazon 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 table shows the disk and network throughput, for each TiB of file system storage capacity, of each Amazon FSx for Lustre deployment option:

<table>
<thead>
<tr>
<th>Deployment Type</th>
<th>Network throughputs (MB/s per TiB of file system storage provisioned)</th>
<th>Network IOPS (IOPS per TiB of file system storage provisioned)</th>
<th>Memory for caching (GiB per TiB of file system storage provisioned)</th>
<th>Disk throughput (MB/s per TiB of file system storage provisioned)</th>
<th>Disk IOPS (IOPS per TiB of file system storage provisioned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Burst</td>
<td>Baseline</td>
<td>Burst</td>
<td>Thousand baseline</td>
<td></td>
</tr>
<tr>
<td>SCRATCH_1</td>
<td>200</td>
<td>350</td>
<td>Tens of thousands baseline</td>
<td>200 (read)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>200 (write)</td>
<td></td>
</tr>
<tr>
<td>SCRATCH_2</td>
<td>200</td>
<td>1300</td>
<td>Tens of thousands baseline</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>200 (read)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100 (write)</td>
<td></td>
</tr>
<tr>
<td>PERSISTENT-50</td>
<td>50</td>
<td>1,300*</td>
<td>Hundreds of thousands baseline</td>
<td>2.2</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240</td>
</tr>
<tr>
<td>PERSISTENT-100</td>
<td>100</td>
<td>1,300*</td>
<td></td>
<td>4.4</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240</td>
</tr>
<tr>
<td>PERSISTENT-200</td>
<td>200</td>
<td>1,300*</td>
<td></td>
<td>8.6</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240</td>
</tr>
</tbody>
</table>

**Note**

*Persistent file systems in the following AWS Regions provide network burst up to 530 MB/s per TiB of storage: Europe (Frankfurt), Europe (London), Europe (Stockholm), Asia Pacific (Singapore), and US West (Los Angeles).

For an example, a persistent file system with a storage capacity of 4.8 TiB and a per unit storage throughput of 200 MB/s per TiB provides an aggregate baseline throughput of 938 MB/s and a burst throughput of 6,094 MB/s per TiB of storage provisioned.

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 disks called object storage targets (OSTs). All file metadata (including file names, timestamps, permissions, and more) is stored on disks called metadata targets (MDTs). Amazon FSx for Lustre file systems are composed of a single MDT and multiple OSTs, each of which is built on SSD storage. Each OST is approximately 1.17 TiB in size. Amazon FSx for Lustre automatically 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 listing and 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. To view the storage capacity and consumption of each disk, run the following command.

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

The output of this command looks like the following.

**Example**

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

**Striping Data in Your File System**

Using Lustre, you can configure how files are striped across 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.

By default, each file created in Amazon FSx for Lustre using standard Linux tools is stored on a single disk. For files imported from Amazon S3, the file system's `ImportedFileChunkSize` parameter determines how many OSTs imported files will be striped across. Files larger than the `ImportedFileChunkSize` will be stored on multiple OSTs.

You can view the striping configuration of a file or directory using the following command:

```
lfs getstripe 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. For more information, see Configuring Lustre File Striping on wiki.lustre.org.

A file's striping parameters are set when the file is first created. Use the following command to create a new, empty file with a determined striping configuration.

```
lfs setstripe filename --stripe-count # of OSTs --stripe-size # of bytes
```

To modify the striping of an existing file, you can create a new file with the desired striping configuration using the `lfs setstripe` command. You then copy the original file into this new file.

**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. 53).

**Performance Tips**

When using Amazon FSx for Lustre, keep the following performance tips in mind. For service limits, see Quotas (p. 75).

- **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 \((2.4 \text{ GiB} / 6 \text{ OSTs}) = 400 \text{ MiB}\) 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 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.

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. 7). Next, you can install and configure the Lustre client on your compute instance.

Topics

• Installing the Lustre Client (p. 41)
• Mounting from an Amazon EC2 Instance (p. 47)
• Mounting Amazon FSx File Systems from On-Premises or a Peered Amazon VPC (p. 48)
• Mounting Your Amazon FSx File System Automatically (p. 49)
• Unmounting File Systems (p. 51)
• Working with Amazon EC2 Spot Instances (p. 52)

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 supports 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.

To Install the Lustre Client as an RPM Package (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. The Lustre client requires Amazon Linux kernel 4.14, version 104 or higher.

   ```bash
   uname -r
   ```

3. Do one of the following:

   • If the command returns 4.14.104-95.84.amzn2.x86_64 or a higher version of 4.14, download and install the Lustre client with the following command.

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

   • If the command returns a result less than 4.14.104, 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.

To Install the Lustre Client as an RPM Package (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 with the following command.

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

- If the command returns a result less than 4.14.104, 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.

To Install the Lustre Client as an RPM Package (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 2.10.6 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-2.10.6/el7/client/RPMS/x86_64/kmod-lustre-client-2.10.6-1.el7.x86_64.rpm
```
Installing the Lustre Client

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

To Install the Lustre Client as an RPM Package (CentOS and Red Hat 7.7 and Newer)

Note
The following procedure is not for CentOS and Red Hat 8.

Lustre client packages that are compatible with Red Hat Enterprise Linux (RHEL) and CentOS can be installed and updated from the Amazon FSx yum package repository. These packages are signed to ensure they have not been tampered with before or during download. Installation fails if the corresponding public key has not been installed on your system.

1. Open a terminal on your client.
2. Install the Amazon FSx rpm public key by using the following command.

```
```

3. Import the key by using the following command.

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

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

```
```

To stay on a specific version of RHEL and avoid moving to later versions as they are released, you can edit this repository configuration file.

To install a specific version of RHEL

1. Edit the repository configuration file to point to the specific version of RHEL that you want to stay on:
   a. Install a specific version by providing the version in the following command.

```
sudo sed -i 's@7@specific version@' /etc/yum.repos.d/aws-fsx.repo
```

   For example, if you want to stay on version 7.7 specifically, substitute `specific version` with 7.7 in the command.

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

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

```
sudo yum clean all
```

2. Install the packages from the repository. Use the following command.

```
sudo yum -y install https://downloads.whamcloud.com/public/lustre/lustre-2.10.6/el7/client/RPMS/x86_64/lustre-client-2.10.6-1.el7.x86_64.rpm
```
Installing the Lustre Client

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

Additional Information (CentOS and RedHat 7.7 and Newer)

The commands preceding install the two packages that are necessary for mounting and interacting with your Amazon FSx file system. Additional Lustre packages such as a package containing the source code and packages containing tests are included in the repository, 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 we have 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 as an RPM Package (SUSE Linux 12 SP3)**

1. Open a terminal on your client.
2. Download and install the Lustre client with the following commands. The client comes in two packages to download and install.

```
sudo rpm -ivh https://downloads.whamcloud.com/public/lustre/lustre-2.10.6/sles12sp3/client/RPMS/x86_64/lustre-client-kmp-default-2.10.6_k4.4.155_94.50-1.x86_64.rpm
sudo rpm -ivh https://downloads.whamcloud.com/public/lustre/lustre-2.10.6/sles12sp3/client/RPMS/x86_64/lustre-client-2.10.6-1.x86_64.rpm
```

**Note**

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

**To Install the Lustre Client as a .deb Package (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.

```bash
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. Use the following command.

```bash
sudo reboot
```

3. Install the Lustre client. Use the following command.

```bash
sudo apt-get install -y lustre-client-modules-$(uname -r)
```

To Install the Lustre Client as a .deb Package (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.

   ```bash
   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.

   ```bash
   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.
   a. Run the following command to determine which kernel is running.

   ```bash
   uname -r
   ```

   b. Run the following command to update to the latest Ubuntu kernel and Lustre version and then reboot.

   ```bash
   sudo apt install -y linux-aws lustre-client-modules-aws && sudo reboot
   ```

   If your kernel version is greater than 4.15.0-1054-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.
Mounting from an Amazon EC2 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 Amazon FSx for Lustre file system for the mount point with the following command.

```bash
$ 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, the `describe-file-systems` CLI command, and the `DescribeFileSystems` API operation.

```bash
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 `-o 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.

```bash
$ ls /fsx
import-path  lustre
```

You can also use the `df` command, following.

```bash
$ df /fsx
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 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 AWS Management Console.

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

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

**To get 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 Amazon 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. 7) 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, the `describe-file-systems` 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. 77).

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>file_system_dns_name</code></td>
<td>The DNS name for your Amazon FSx file system, which identifies the file system. You can get this name from the console or programatically from the AWS CLI or an AWS SDK.</td>
</tr>
<tr>
<td><code>mountname</code></td>
<td>The mount name for the file system. You can get this name from the console or programatically from the AWS CLI using the <code>describe-file-systems</code> API.</td>
</tr>
</tbody>
</table>
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
```

Field | Description
---|---
`systems` command or an AWS 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.
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.
Working with Amazon EC2 Spot Instances

Amazon 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. 51).
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 Are Amazon CloudWatch, Amazon CloudWatch Events, and Amazon CloudWatch Logs? in the Amazon CloudWatch User Guide.

As with Amazon EFS, Amazon S3, and Amazon EBS, Amazon FSx for Lustre 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 AWS/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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataReadBytes</td>
<td>The number of bytes for file system read operations.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>To calculate the average throughput (bytes per second) for a period, divide the Sum statistic by the number of seconds in the period.</td>
</tr>
<tr>
<td></td>
<td>Units:</td>
</tr>
<tr>
<td></td>
<td>• Bytes for Sum, Minimum, Maximum, and Average.</td>
</tr>
<tr>
<td></td>
<td>• Count for SampleCount.</td>
</tr>
<tr>
<td></td>
<td>Valid statistics: Sum, Minimum, Maximum, Average, SampleCount</td>
</tr>
<tr>
<td>DataWriteBytes</td>
<td>The number of bytes for file system write operations.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
### Monitoring CloudWatch

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To calculate the average throughput (bytes per second) for a period, divide the Sum statistic by the number of seconds in the period.</td>
</tr>
<tr>
<td></td>
<td><strong>Units:</strong></td>
</tr>
<tr>
<td></td>
<td>• Bytes for Sum, Minimum, Maximum, and Average.</td>
</tr>
<tr>
<td></td>
<td>• Count for SampleCount.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid statistics:</strong> Sum, Minimum, Maximum, Average, SampleCount</td>
</tr>
<tr>
<td>DataReadOperations</td>
<td>The number of read operations.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Units:</strong></td>
</tr>
<tr>
<td></td>
<td>• Bytes for Sum, Minimum, Maximum, and Average.</td>
</tr>
<tr>
<td></td>
<td>• Count for SampleCount.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid statistics:</strong> Sum, Minimum, Maximum, Average, SampleCount</td>
</tr>
<tr>
<td>DataWriteOperations</td>
<td>The number of write operations.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Units:</strong></td>
</tr>
<tr>
<td></td>
<td>• Bytes for Sum, Minimum, Maximum, and Average.</td>
</tr>
<tr>
<td></td>
<td>• Count for SampleCount.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid statistics:</strong> Sum, Minimum, Maximum, Average, SampleCount</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MetadataOperations</td>
<td>The number of metadata operations.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Units:</td>
</tr>
<tr>
<td></td>
<td>• Count for Sum, Minimum, Maximum, Average, and SampleCount.</td>
</tr>
<tr>
<td></td>
<td>Valid statistics: Sum, Minimum, Maximum, Average, SampleCount.</td>
</tr>
<tr>
<td>FreeDataStorageCapacity</td>
<td>The amount of available storage capacity.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Units:</td>
</tr>
<tr>
<td></td>
<td>• Bytes for Sum, Minimum, Maximum.</td>
</tr>
<tr>
<td></td>
<td>• Count for SampleCount.</td>
</tr>
<tr>
<td></td>
<td>Valid statistics: Sum, Minimum, Maximum, Average, SampleCount.</td>
</tr>
</tbody>
</table>

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 aws fsx 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.

<table>
<thead>
<tr>
<th>How Do I Determine...</th>
<th>Relevant Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>My file system's throughput?</td>
<td>SUM(DataReadBytes + DataWriteBytes)/Period (in seconds)</td>
</tr>
</tbody>
</table>
### 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**

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*. 
Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon FSx for 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 AWS 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 (p. 58)
- File System Access Control with Amazon VPC (p. 62)
- Amazon VPC Network ACLs (p. 66)
- Administration Access Control with IAM for Amazon FSx for Lustre Resources (p. 66)
- Compliance Validation for Amazon FSx for Lustre (p. 70)

### Data Protection in Amazon FSx for Lustre

Amazon FSx for Lustre conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

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), so that 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.
• 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.

We strongly recommend that you never put sensitive identifying information, such as your customers’ account numbers, into 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 Amazon FSx or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don’t include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

Topics
• Data Encryption in Amazon FSx for Lustre (p. 59)
• Internetwork Traffic Privacy (p. 61)

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. 7)
• Amazon FSx for Lustre API Permissions: Actions, Resources, and Conditions Reference (p. 69)

Topics
• Encrypting Data at Rest (p. 59)
• Encrypting Data in Transit (p. 60)
• How Amazon FSx for Lustre Uses AWS KMS (p. 60)

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. 7).
Amazon FSx for Lustre Lustre User Guide

Data Encryption in Amazon FSx

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 Amazon FSx for Lustre uses AWS KMS, see How Amazon FSx for Lustre Uses AWS KMS (p. 60).

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.

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>Scratch 2</th>
<th>Persistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (Ohio)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>US West (Oregon), excluding US West (Los Angeles)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

How Amazon FSx for Lustre Uses AWS KMS

AA-SERVICENAMEElong; integrates with AWS Key Management Service (AWS KMS) for key management for encrypting data at rest. Amazon FSx uses customer master keys (CMKs) to encrypt your file system in the following way:

- Encrypting data at rest – Amazon FSx for Lustre uses a single customer master key (CMK), either the AWS managed CMK for Amazon FSx or a custom CMK, 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 CMK used to encrypt and decrypt data, either the AWS managed CMK for
Amazon FSx or a custom CMK. You specify which key to use when you create a persistent file system. You can enable, disable, or revoke grants on this CMK. This CMK can be one of the two following types:

- **AWS managed CMK for Amazon FSx** – This is the default CMK. You’re not charged to create and store a CMK, but there are usage charges. For more information, see [AWS Key Management Service pricing](https://aws.amazon.com/kms/pricing/).
- **Customer-managed CMK** – This is the most flexible master key to use, because you can configure its key policies and grants for multiple users or services. For more information on creating CMKs, see [Creating Keys](https://docs.aws.amazon.com/kms/latest/developerguide/creating-keys.html) in the *AWS Key Management Service Developer Guide*.

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

**Important**

Amazon FSx accepts only symmetric CMKs. You can't use asymmetric CMKs with Amazon FSx.

### Amazon FSx Key Policies for AWS KMS

Key policies are the primary way to control access to CMKs. For more information on key policies, see [Using Key Policies in AWS KMS](https://docs.aws.amazon.com/kms/latest/developerguide/using-key-policies.html) in the *AWS Key Management Service Developer Guide*. The following list describes all the AWS KMS–related permissions supported by Amazon FSx for encrypted at rest file systems:

- **kms:Encrypt** – (Optional) Encrypts plaintext into ciphertext. This permission is included in the default key policy.
- **kms:Decrypt** – (Required) Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted. This permission is included in the default key policy.
- **kms:ReEncrypt** – (Optional) Encrypts data on the server side with a new customer master key (CMK), without exposing the plaintext of the data on the client side. The data is first decrypted and then re-encrypted. This permission is included in the default key policy.
- **kms:GenerateDataKeyWithoutPlaintext** – (Required) Returns a data encryption key encrypted under a CMK. 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](https://docs.aws.amazon.com/kms/latest/developerguide/using-grants.html) 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 customer master 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 **Select KMS master key** list. 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 Service and On-Premises Clients and Amazon FSx**

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?

Access to Amazon FSx via the network is through 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, make sure that you sign requests using an access key ID and a secret access key that are associated with an IAM principal. Or you can use the AWS Security Token Service (STS) to generate temporary security credentials to sign requests.

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.
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 Amazon 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group ID of the security group that you just created</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Lustre clients</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group ID of the security group that you just created</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Lustre clients</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
</tbody>
</table>

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 Amazon 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group ID of the security group that you just created</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group IDs of the security group associated with your Lustre clients</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group ID of the security group that you just created</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Lustre clients</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
</tbody>
</table>

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/](https://console.aws.amazon.com/fsx/).
2. On the console dashboard, choose 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** under **Actions**.
5. In the **Change Security Groups** dialog box, choose the security groups to use, and choose **Save**.
### Lustre Client VPC Security Group Rules

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Amazon FSx for Lustre file systems</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups that are applied to your Lustre clients</td>
<td>Allows Lustre traffic between Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Amazon FSx for Lustre file systems</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group IDs of the security groups that are applied to your Lustre clients</td>
<td>Allows Lustre traffic between Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>988</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Amazon FSx for Lustre file systems</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups that are applied to your Lustre clients</td>
<td>Allows Lustre traffic between Lustre clients</td>
</tr>
</tbody>
</table>
### 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. 62).

### Administration Access Control with IAM for Amazon FSx for Lustre Resources

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**

- Resources and Operations for Amazon FSx for Lustre (p. 66)
- Using Service-Linked Roles for Amazon FSx for Lustre (p. 67)
- Understanding Resource Ownership (p. 68)
- Managing Access to Resources (p. 69)
- Amazon FSx for Lustre API Permissions: Actions, Resources, and Conditions Reference (p. 69)

### Resources and Operations for Amazon FSx for Lustre

In Amazon FSx for Lustre, the primary resource is a file system. Amazon FSx for Lustre also supports additional resource types, the data repository task. However, for Amazon FSx, you can create data repository tasks only in the context of an existing file system. Data repository tasks are referred to as

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Amazon FSx for Lustre file systems</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom TCP rule</td>
<td>TCP</td>
<td>1021-1023</td>
<td>Choose Custom and enter the security group IDs of the security groups associated with your Amazon FSx for Lustre file systems</td>
<td>Allows Lustre traffic between Amazon FSx for Lustre file servers and Lustre clients</td>
</tr>
</tbody>
</table>
subresources. 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.

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:PutObject
- s3:Get*
- s3[List*

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.

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.

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.

### Managing Access to Resources

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

**Note**
This section discusses using IAM in the context of Amazon FSx for 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.

### Amazon FSx for Lustre API and Required Permissions for Actions

<table>
<thead>
<tr>
<th>Amazon FSx for Lustre API Operations</th>
<th>Required Permissions (API Actions)</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescribeFileSystems</td>
<td>fsx:DescribeFileSystems</td>
<td>N/A</td>
</tr>
</tbody>
</table>
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 Amazon 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.

```json
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-11-14T22:36:07Z"
            }
        },
    },
    "eventTime": "2018-11-14T22:36:07Z",
    "eventSource": "fsx.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "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.

```json
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "111122223333",
        "arn": "arn:aws:sts::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
```
"mfaAuthenticated": "false",
"creationDate": "2018-11-14T23:40:54Z"
}

"eventTime": "2018-11-14T23:40:54Z",
"eventSource": "fsx.amazonaws.com",
"eventName": "UntagResource",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "console.amazonaws.com",
"requestParameters": {
  "resourceARN": "arn:aws:fsx:us-east-1:111122223333:file-system/fs-ab12cd34ef56gh789"
},
"responseElements": null,
"requestID": "aEXAMPLE-abcd-1234-56ef-b4cEXAMPLE51",
"eventID": "bEXAMPLE-gl12-3f5h-3sh4-ab6EXAMPLE9p",
"eventType": "AwsApiCall",
"apiVersion": "2018-03-01",
"recipientAccountId": "111122223333"}
Amazon FSx for 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.

Changing the Maintenance Window of an Existing File System

You can use the AWS CLI or one of the AWS SDKs to change the maintenance window for your file systems.

To update the weekly maintenance window of your file system

1. Open a command prompt or terminal on your computer.
2. Run the following command, replacing the file system ID with the ID for your file system, and the date and time with when you want to begin the window.

    `aws fsx update-file-system --file-system-id fs-01234567890123456 --lustre-configuration WeeklyMaintenanceStartTime=1:01:30`
# Quotas

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

## Quotas That You Can Increase

Following are the quotas for Amazon FSx for Lustre per AWS account, per AWS Region, which you can increase for specific deployment types (p. 13) by contacting AWS Support.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default limit</th>
<th>Can be increased up to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of file systems</td>
<td>100</td>
<td>Thousands</td>
</tr>
<tr>
<td>Total storage for all file systems</td>
<td>US East (Ohio) Region – 100,800 GiB</td>
<td>Petabytes</td>
</tr>
<tr>
<td></td>
<td>US East (N. Virginia) Region – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (N. California) Region – 25,200 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (Oregon) Region – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Hong Kong) Region – 25,200 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Singapore) – 25,200 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Sydney) – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Tokyo) – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Frankfurt) Region – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Ireland) Region – 100,800 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (London) Region – 25,200 GiB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Stockholm) Region – 25,200 GiB</td>
<td></td>
</tr>
</tbody>
</table>

**To request a limit increase**

1. Open the AWS Support Center page, sign in, if necessary, and then choose Create Case.
2. Under **Regarding**, choose **Service Limit Increase**.

3. Under **Limit Type**, choose the type of limit to increase, fill in the necessary fields in the form, and then choose your preferred method of contact.

## Resource Quotas for Each File System

Following are the limits on Amazon FSx for Lustre resources for each file system in an AWS Region.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Limit per file system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tags</td>
<td>50</td>
</tr>
</tbody>
</table>

For information on throughput capacity, see *Amazon FSx for Lustre Performance (p. 37)*.
Troubleshooting

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. 62).

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 Using DNS Name Fails**

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.

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. 62).

---

**Creating a File System with Data Repository Fails**

You can't create a file system linked to a data repository in Amazon S3 bucket, and encounter an error like 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 a data repository in 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. 6).

For more information about service-linked roles, see Using Service-Linked Roles for Amazon FSx for Lustre (p. 67).
### Document History

- **API version:**
- **Latest documentation update:** February 12, 2020

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.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two new file system deployment types released (p. 80)</td>
<td>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 <em>FSx for Lustre Deployment Options</em>.</td>
<td>February 12, 2020</td>
</tr>
<tr>
<td>Support for POSIX metadata added (p. 80)</td>
<td>Amazon 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 <em>POSIX Metadata Support for Data Repositories</em>.</td>
<td>December 23, 2019</td>
</tr>
<tr>
<td>New data repository tasks feature released (p. 80)</td>
<td>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 <em>Transferring Data &amp; Metadata Using Data Repository Tasks</em>.</td>
<td>December 23, 2019</td>
</tr>
<tr>
<td>Additional AWS Region support added (p. 80)</td>
<td>Amazon FSx for Lustre is now available in the Europe (London) Region AWS Region. For Amazon FSx for Lustre region-specific limits, see <em>Limits</em>.</td>
<td>July 9, 2019</td>
</tr>
<tr>
<td>Additional AWS Region support added (p. 80)</td>
<td>Amazon FSx for Lustre is now available in the Asia Pacific (Singapore) AWS Region. For Amazon FSx for Lustre region-specific limits, see <em>Limits</em>.</td>
<td>June 26, 2019</td>
</tr>
<tr>
<td>Lustre client support for Amazon Linux and Amazon Linux 2 added (p. 80)</td>
<td>The Amazon FSx for Lustre client now supports Amazon EC2 instances running Amazon Linux and Amazon Linux 2. For more information see <em>Installing the Lustre Client</em>.</td>
<td>March 11, 2019</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>User-defined data export path support added (p. 80)</td>
<td>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 Amazon FSx for Lustre into your data processing workflows. For more information, see Exporting Data to Your Amazon S3 Bucket.</td>
<td>February 6, 2019</td>
</tr>
<tr>
<td>Total storage default limit increased (p. 80)</td>
<td>The default total storage for all Amazon FSx for Lustre file systems increased to 100,800 GiB. For more information, see Limits.</td>
<td>January 11, 2019</td>
</tr>
<tr>
<td>Amazon FSx for Lustre is now generally available (p. 80)</td>
<td>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.</td>
<td>November 28, 2018</td>
</tr>
</tbody>
</table>