

# Service Name

Amazon File Cache

## Service Overview

Amazon File Cache is a fully managed, high-speed caching service on AWS that enables processing of file data from various data repositories, including on-premises file systems, AWS file systems, and Amazon S3 buckets. It serves as a temporary, high-performance storage location that presents data as a unified set of files and directories. The service provides sub-millisecond latency and high throughput access to dispersed datasets for file-based applications on AWS, supporting hundreds of GBps throughput and millions of IOPS. Amazon File Cache is POSIX-compliant and uses automatic data loading and release mechanisms for optimal performance and cost efficiency.

## Key Use Cases

- **High performance computing (HPC) workloads:** Processing compute-intensive tasks requiring fast access to large datasets distributed across multiple storage locations
- **Machine learning and data analytics:** Accelerating ML workflows by providing high-speed access to training data stored in S3 or on-premises systems
- **Media and content processing:** Enabling fast processing of media files and content workflows that require high throughput and low latency access

## Features

- **Multi-Repository Support:** Links up to 8 data repositories of the same type (Amazon S3 or NFS) per cache
- **High Performance:** Delivers sub-millisecond latencies, hundreds of GBps throughput, and millions of IOPS
- **POSIX Compliance:** Provides native file system interface with read-after-write consistency and file locking
- **Automatic Data Management:** Lazy-loads data on first access and supports preloading with HSM commands for data export
- **Cache Eviction:** Automatically releases less recently used files when cache fills up, with manual control options
- **Storage Quotas:** Supports user and group quotas for disk space and file count limits
- **Encryption:** Supports encryption at rest and in transit using AWS KMS keys
- **Lustre Integration:** Built on Lustre technology with support for open-source Lustre client and CSI driver

## Value Proposition

Amazon File Cache eliminates the complexity of managing high-performance file caching infrastructure while providing exceptional performance with sub-millisecond latencies and massive throughput capabilities. Unlike traditional caching solutions, it offers fully managed operations with automatic scaling, data replication within Availability Zones, and seamless integration with both cloud and on-premises data sources:

- The service provides cost optimization through pay-as-you-use pricing, automatic cache eviction, and eliminates upfront hardware investments
- Its POSIX compliance ensures existing Linux applications work without modifications, while native AWS integrations with services like Batch, EKS, and ECS accelerate deployment of compute-intensive workloads

## Service Integration

Amazon File Cache integrates natively with multiple core AWS services to enable comprehensive compute and storage workflows. It connects seamlessly with Amazon S3 buckets and NFS file systems as data repositories, supports access from Amazon EC2 instances across Availability Zones, and works with containerized workloads on Amazon ECS and Amazon EKS through the CSI driver:

- The service integrates with AWS Batch for high-performance computing workloads and AWS Thinkbox Deadline for render farm management
- It leverages AWS KMS for encryption, AWS CloudTrail for API logging, Amazon CloudWatch for performance monitoring, and AWS PrivateLink for secure private network access
- VPC integration provides network isolation and security group controls

## Onboarding and Offboarding

Getting started with Amazon File Cache requires an AWS account, Amazon EC2 instance, and appropriate IAM permissions. Customers create a cache through the AWS Console, CLI, or API by specifying storage capacity, subnet IDs, security groups, and data repository associations:

- The Lustre client must be installed on EC2 instances (included with Amazon Linux 2/Amazon Linux, available via AWS repositories for RHEL/CentOS/Ubuntu)
- After mounting the cache using standard Linux commands, applications can access cached data immediately
- The service supports both lazy loading and preloading strategies
- Getting started documentation provides step-by-step guidance including prerequisites, cache creation, client configuration, and running analysis workloads

Getting Started Guide: <https://docs.aws.amazon.com/fsx/latest/FileCacheGuide/getting-started.html>

## Data Removal

Offboarding from Amazon File Cache involves deleting cache resources through AWS Console, CLI, or API using the DeleteFileCache operation. When a cache is deleted, all cached data is permanently lost and cannot be recovered. Customers should export any modified data back to linked data repositories using HSM commands before deletion. The process includes terminating EC2 instances, deleting the cache (which transitions to DELETING status), and optionally removing associated S3 buckets if no longer needed for data preservation.

## Backup/Restore and Disaster Recovery

Amazon File Cache does not provide traditional backup capabilities as it serves as a temporary, high-performance cache rather than persistent storage. Data protection relies on the underlying data repositories (S3 or NFS) where the source data resides. The service provides automatic data replication within the same Availability Zone and automatic server replacement if failures occur. For disaster recovery, customers must ensure their linked data repositories have appropriate backup and recovery strategies implemented.

### Backup Capabilities

Amazon File Cache does not support traditional backup capabilities and is not integrated with AWS Backup service. As a caching service designed for temporary, high-performance storage, data protection relies on the durability and backup mechanisms of the linked data repositories (Amazon S3 or NFS). The service itself provides automatic replication within the Availability Zone and server replacement for fault tolerance.

### Disaster Recovery

Amazon File Cache does not integrate with AWS Elastic Disaster Recovery as it functions as a temporary cache rather than persistent storage. Disaster recovery strategies must focus on the underlying data repositories:

- The service provides automatic data replication within the same Availability Zone and server replacement capabilities
- For multi-region disaster recovery, customers must implement appropriate replication strategies for their S3 buckets or NFS data repositories

### Recovery Objectives

Amazon File Cache helps meet RPO and RTO objectives through its automatic replication within Availability Zones and rapid cache recreation capabilities. Since it's a temporary cache with data sourced from repositories, RPO depends on the backup strategy of underlying S3 or NFS data sources. RTO can be minimized through quick cache

provisioning and automatic data loading. However, customers must implement comprehensive disaster recovery planning for their data repositories to achieve specific RPO/RTO targets.

## Service Constraints

Service Quotas: <https://docs.aws.amazon.com/fsx/latest/FileCacheGuide/limits.html>

FAQ: <https://aws.amazon.com/filecache/faqs/>

## Technical Requirements

Service Documentation: <https://docs.aws.amazon.com/fsx/latest/FileCacheGuide/>

User Guide: <https://docs.aws.amazon.com/fsx/latest/FileCacheGuide/what-is.html>

API Reference: <https://docs.aws.amazon.com/fsx/latest/APIReference/Welcome.html>

## Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: <https://aws.amazon.com/filecache/pricing/>

## Cost Optimization

Amazon File Cache offers several unique cost optimization features including pay-only-for-provisioned-storage with no upfront costs, automatic cache eviction to free space for new data, and intelligent data loading that reduces unnecessary storage consumption. The service eliminates capital expenditure on caching infrastructure and reduces operational overhead through fully managed operations:

- Customers can optimize costs by right-sizing cache capacity, using lazy loading instead of preloading when appropriate, implementing storage quotas to control usage, and leveraging cache eviction policies
- The temporary nature of cached data reduces long-term storage costs compared to persistent file systems

## Savings Considerations

Primary cost savings come from eliminating the need to provision and manage high-performance caching infrastructure, reducing compute resource consumption through faster workload completion times, and optimizing data transfer costs by caching frequently accessed data closer to compute resources. The service reduces operational expenses through automated administration tasks, eliminates hardware refresh cycles, and provides elastic scaling without over-provisioning:

- Pay-as-you-use pricing ensures customers only pay for actual storage consumed, while automatic cache management reduces the need for dedicated storage administration staff
- Integration with spot instances and containerized workloads further enhances cost efficiency for batch processing scenarios