

Service Name

Amazon Managed Service for Apache Flink

Service Overview

Amazon Managed Service for Apache Flink is a fully managed service that enables users to process and analyze streaming data using Apache Flink, an open-source framework for stream and batch processing. The service handles underlying infrastructure management including resource provisioning, availability zone failover resilience, parallel computation, automatic scaling, and application backups. Users can build applications using Java, Scala, Python, or SQL with support for Apache Flink's DataStream API, Table API, SQL API, and Process Functions. The service offers two deployment options: standard applications for long-running production workloads built with IDEs, and Studio notebooks for interactive real-time data exploration and dashboard creation using Apache Zeppelin notebooks.

Key Use Cases

- **Real-time analytics:** Processing and analyzing streaming data for immediate insights and time-series analytics
- **Event-driven applications:** Building responsive applications that react to streaming events with low latency
- **Streaming ETL:** Transforming and loading streaming data into data warehouses and data lakes in real-time
- **Real-time dashboards and metrics:** Creating interactive dashboards and monitoring systems with live data updates
- **Complex event processing:** Detecting patterns and anomalies in streaming data for fraud detection and monitoring
- **IoT data processing:** Processing high-volume sensor data and telemetry from connected devices
- **Clickstream analysis:** Real-time analysis of user behavior and website interactions for personalization

Features

- **Multiple Apache Flink APIs:** Support for Flink SQL, Table API, DataStream API, and Process Functions with Java, Scala, Python, and SQL
- **Automatic Scaling:** Auto-scales based on CPU utilization with configurable parallelism and Kinesis Processing Units (KPU)
- **Fault Tolerance:** Built-in checkpointing, snapshots, and savepoints for application state management and recovery
- **Studio Notebooks:** Interactive Apache Zeppelin notebooks for ad-hoc data exploration and real-time dashboard creation

- **High Availability:** Multi-AZ deployment with automatic failover and resilience across availability zones
- **Per-second Billing:** Pay-per-use pricing model with one-second increments and 10-minute minimum charge
- **Rich Connector Library:** Over 40 pre-built connectors for AWS services and external systems including Kafka, DynamoDB, S3
- **System Rollbacks:** Automatic rollback capabilities for failed updates and scaling operations

Value Proposition

Customers should choose Amazon Managed Service for Apache Flink for its serverless architecture that eliminates infrastructure management overhead while providing enterprise-grade reliability and performance. The service offers significant operational advantages including automatic scaling, built-in fault tolerance through checkpointing and snapshots, and multi-AZ high availability:

- Its per-second billing model with automatic resource optimization reduces costs compared to provisioning fixed infrastructure
- The service provides comprehensive integration with the AWS ecosystem through 40+ pre-built connectors, making it ideal for customers already using AWS services
- Additionally, the dual deployment options - production applications and interactive Studio notebooks - support both development and operational use cases, while the managed nature ensures consistent performance without the complexity of managing Apache Flink clusters

Service Integration

Amazon Managed Service for Apache Flink integrates deeply with core AWS analytics and data services. Primary integrations include Amazon Kinesis Data Streams for real-time data ingestion, Amazon MSK (Managed Streaming for Apache Kafka) for event streaming, Amazon S3 for data storage and batch processing, and Amazon DynamoDB for real-time database operations:

- The service connects to Amazon Data Firehose for data delivery, Amazon OpenSearch Service for search analytics, and Amazon Redshift for data warehousing
- Additional integrations encompass Amazon SQS for messaging, Amazon Managed Service for Prometheus for monitoring, CloudWatch for logging and metrics, AWS Glue for data catalog access, and IAM for security management
- The service also supports AWS KMS for encryption and VPC for network isolation, providing comprehensive integration across the AWS data and analytics ecosystem

Onboarding and Offboarding

Customers get started by first installing prerequisites including Java Development Kit 11, Apache Maven, Git client, and an IDE for development. The process begins with creating an AWS account and setting up the AWS CLI for programmatic access:

- Users then develop Apache Flink applications locally using DataStream API, Table API, or SQL, package them as JAR files, and upload to Amazon S3
- Applications are created through the AWS Console or CLI, configured with runtime properties, and started with specified parallelism settings
- For interactive development, customers can create Studio notebooks directly in the console using Apache Zeppelin for SQL, Python, or Scala development
- The service provides comprehensive getting started tutorials, workshops, and example applications to accelerate onboarding, with step-by-step guides available for different programming languages and use cases

Getting Started Guide: <https://docs.aws.amazon.com/managed-flink/latest/java/getting-started.html>

Data Removal

Data removal involves stopping applications using the StopApplication API, which automatically creates a final snapshot before termination. Customers must manually delete application snapshots using DeleteApplicationSnapshot API as the service never automatically deletes them. Application code stored in Amazon S3 buckets must be removed separately. VPC configurations, CloudWatch logs, and IAM roles created for the service should be cleaned up manually. The service provides cleanup guidance in documentation to ensure complete data removal and resource decommissioning during offboarding.

Backup/Restore and Disaster Recovery

The service provides comprehensive backup through automatic checkpointing every 60 seconds by default and manual snapshots for long-term state preservation. Checkpoints enable automatic recovery from failures, while snapshots allow point-in-time restoration during updates and scaling operations. Multi-AZ deployment ensures high availability with automatic failover capabilities. System rollback features automatically restore previous application versions during failed operations. All backup data is encrypted and stored in service-managed S3 buckets with configurable retention policies.

Backup Capabilities

Amazon Managed Service for Apache Flink provides built-in backup through checkpointing and snapshots but does not directly integrate with AWS Backup service. The service manages its own backup strategy using Apache Flink's native checkpointing mechanism for automatic recovery and snapshot functionality for manual state preservation:

- Customers can create up to 1,000 snapshots per application for long-term backup retention
- All backup data is automatically encrypted and stored in service-owned S3 buckets, providing durable state preservation without requiring AWS Backup integration

Disaster Recovery

The service provides disaster recovery through multi-AZ high availability deployments and automatic failover mechanisms but does not integrate with AWS Elastic Disaster Recovery service. Managed Service for Apache Flink implements its own disaster recovery strategy using distributed JobManager deployment across availability zones, automatic checkpoint recovery, and snapshot-based restoration. The service's resilient architecture ensures applications can recover from infrastructure failures within the same region through built-in fault tolerance mechanisms rather than requiring external disaster recovery services.

Recovery Objectives

Amazon Managed Service for Apache Flink helps customers achieve low RTO through automatic failover across availability zones and rapid application restart from checkpoints, typically within minutes. RPO objectives are met through configurable checkpoint intervals (default 60 seconds) and manual snapshots for point-in-time recovery. The service's automatic scaling and system rollback features minimize downtime during updates and scaling operations. Multi-AZ deployment and stateful recovery mechanisms ensure minimal data loss and quick service restoration during failures.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/managed-flink/latest/java/limits.html>

FAQ: <https://aws.amazon.com/managed-service-apache-flink/faqs/>

Technical Requirements

Service Documentation: <https://docs.aws.amazon.com/managed-flink/latest/java/what-is.html>

User Guide: <https://docs.aws.amazon.com/managed-flink/latest/java/getting-started.html>

API Reference: <https://docs.aws.amazon.com/managed-flink/latest/apiv2/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/managed-service-apache-flink/pricing/>

Cost Optimization

Amazon Managed Service for Apache Flink offers several unique cost optimization features including per-second billing with 10-minute minimum charges, eliminating over-provisioning costs. Automatic scaling adjusts KPIs based on actual workload demand, scaling up when CPU utilization exceeds 75% and down when below 10%:

- The service charges only for consumed resources without upfront commitments or infrastructure provisioning costs
- Studio notebooks provide cost-effective development environments for experimentation without dedicated infrastructure
- Customers can optimize costs by right-sizing parallelism settings, using efficient checkpoint intervals, minimizing JAR file sizes, and leveraging automatic scaling
- The serverless model eliminates idle resource costs typical of self-managed Apache Flink clusters, making it particularly cost-effective for variable workloads and development environments

Savings Considerations

Main cost savings come from eliminating infrastructure management overhead and operational complexity of running Apache Flink clusters. The service reduces total cost of ownership by providing built-in high availability, automatic scaling, and managed updates without dedicated DevOps resources:

- Per-second billing ensures customers only pay for actual usage, particularly beneficial for batch workloads and intermittent streaming applications
- Automatic resource optimization prevents over-provisioning while maintaining performance requirements
- The managed service model eliminates costs associated with cluster monitoring, patch management, and capacity planning
- For organizations with variable streaming workloads, the pay-per-use model provides significant savings compared to maintaining always-on infrastructure
- Development and testing costs are reduced through Studio notebooks and the ability to start/stop applications on-demand without infrastructure penalties