

Service Name

Amazon Managed Workflows for Apache Airflow

Service Overview

Amazon Managed Workflows for Apache Airflow (MWAA) is a fully managed orchestration service for Apache Airflow that makes it easier to set up and operate end-to-end data pipelines in the cloud at scale. It simplifies managing Apache Airflow by automatically setting up and scaling infrastructure, providing built-in authentication and security, and integrating with various AWS services. Amazon MWAA uses Apache Airflow and Python to create workflows without managing underlying infrastructure, automatically scaling workers based on workload demands while maintaining the same open-source Apache Airflow user interface and functionality.

Key Use Cases

- **Batch data processing workflows:** Processing large datasets in scheduled batches with complex dependencies and orchestration requirements
- **ETL pipeline orchestration:** Coordinating extract, transform, and load operations across multiple data sources and destinations
- **Multi-service workflow coordination:** Orchestrating tasks across various AWS services like S3, EMR, Glue, Lambda, and SageMaker for comprehensive data processing
- **Cross-account and cross-region pipeline management:** Building unified pipelines spanning multiple AWS accounts and regions while maintaining compliance
- **Machine learning workflow orchestration:** Managing ML training pipelines, model deployment, and inference workflows with data dependencies
- **Legacy workflow migration:** Transitioning from AWS Data Pipeline or self-managed Airflow deployments to managed infrastructure

Features

- **Automatic Airflow Setup:** Quickly deploys Apache Airflow environments with chosen versions using the same open-source UI and code
- **Automatic Scaling:** Automatically scales workers and webserver based on workload demands with configurable minimum and maximum limits
- **Built-in Security:** Runs in Amazon VPC with automatic data encryption using AWS KMS, IAM integration, and role-based authentication
- **Public/Private Access Modes:** Supports both public internet access and private VPC-only access to Apache Airflow webserver
- **Streamlined Upgrades:** Provides automatic patches and managed upgrades to new Apache Airflow versions

- **Workflow Monitoring:** Integrates with CloudWatch for logs, metrics, and monitoring without third-party tools
- **AWS Service Integration:** Native integration with 20+ AWS services including S3, EMR, Glue, Lambda, and hundreds of operators
- **Serverless Option:** MWAA Serverless provides pay-per-use pricing with automatic infrastructure scaling

Value Proposition

Amazon MWAA eliminates the operational overhead of managing Apache Airflow infrastructure while providing enterprise-grade security, scalability, and reliability. Customers benefit from reduced operational costs by removing the need to manage open-source Apache Airflow at scale, automatic scaling that handles variable workloads efficiently, and seamless integration with AWS services:

- The managed service provides built-in security with VPC isolation and KMS encryption, streamlined upgrades without maintenance windows, and comprehensive monitoring through CloudWatch
- MWAA maintains full compatibility with Apache Airflow APIs and supports both traditional environments and serverless deployments, offering flexibility in pricing models and deployment options that self-managed solutions cannot match

Service Integration

Amazon MWAA integrates natively with over 20 core AWS services through built-in operators and connectors. Primary integrations include Amazon S3 for DAG storage and data operations, Amazon EMR for big data processing, AWS Glue for ETL workflows, AWS Lambda for serverless task execution, and Amazon Redshift for data warehousing:

- Additional integrations span Amazon Athena for analytics, AWS Batch for batch computing, Amazon DynamoDB for NoSQL operations, Amazon EKS and ECS for container orchestration, Amazon SageMaker for machine learning workflows, and messaging services like SQS and SNS
- The service also integrates with AWS security and monitoring services including IAM, KMS, CloudWatch, and CloudTrail, enabling comprehensive workflow orchestration across the entire AWS ecosystem

Onboarding and Offboarding

Getting started with Amazon MWAA requires creating an Amazon S3 bucket with versioning enabled for DAG files, configuring an Amazon VPC with appropriate subnets and security groups, and creating the MWAA environment through the AWS console, CLI, or CloudFormation. Customers specify the Apache Airflow version, S3 bucket location for DAGs, VPC networking configuration, and optional settings for plugins, requirements, and security:

- The environment takes approximately 20-30 minutes to create and automatically provisions schedulers, workers, and webserver as Fargate containers
- Prerequisites include AWS account permissions, VPC setup, and S3 bucket configuration
- Multiple deployment options are available including quick-start CloudFormation templates and incremental setup guides for different VPC networking scenarios

Getting Started Guide: <https://docs.aws.amazon.com/mwaa/latest/userguide/get-started.html>

Data Removal

When offboarding from Amazon MWAA, customers can delete their environment through the AWS console, CLI, or API, which automatically removes all managed infrastructure including schedulers, workers, and webserver. The metadata database and associated VPC endpoints are also cleaned up. However, customers retain full control over their DAG files, plugins, and requirements stored in S3, which must be manually deleted if desired. CloudWatch logs and metrics can be retained or deleted based on configured retention policies, ensuring customers maintain control over their workflow history and operational data.

Backup/Restore and Disaster Recovery

Amazon MWAA provides built-in backup capabilities through its managed metadata database with automatic snapshots and point-in-time recovery. The service stores DAGs and supporting files in Amazon S3, which provides 11 9's of durability and cross-region replication capabilities. While MWAA doesn't directly integrate with AWS Backup service, customers can implement disaster recovery by replicating S3 buckets across regions and recreating environments. The managed nature ensures automatic patching and infrastructure redundancy within regions, while customers can design cross-region DR strategies using S3 replication and Infrastructure as Code approaches.

Backup Capabilities

Amazon MWAA includes automatic backup capabilities through its managed metadata database with built-in snapshots and point-in-time recovery. DAG files and supporting artifacts are stored in Amazon S3, providing inherent durability and versioning capabilities:

- However, MWAA does not directly integrate with AWS Backup service for centralized backup management
- Customers can implement backup strategies using S3 versioning, cross-region replication, and Infrastructure as Code templates to recreate environments
- The service automatically maintains environment configuration and can be restored through CloudFormation or Terraform templates

Disaster Recovery

Amazon MWAA does not directly integrate with AWS Elastic Disaster Recovery service. However, the service supports disaster recovery through its architecture using S3 for DAG storage with cross-region replication capabilities and managed metadata database with automatic backups:

- Customers can implement disaster recovery by replicating S3 buckets to secondary regions and using Infrastructure as Code to quickly recreate MWAA environments
- The containerized architecture on Fargate provides built-in fault tolerance within regions, while multi-region strategies require customer-implemented cross-region replication and environment recreation processes

Recovery Objectives

Amazon MWAA helps customers meet RPO and RTO objectives through its managed architecture and S3-based storage model. RPO can be minimized to near-zero using S3 cross-region replication for DAG files and continuous metadata database backups. RTO depends on environment recreation time (20-30 minutes) plus S3 replication lag. The service supports backup and restore DR strategies with RTOs of 2-4 hours, suitable for most workflow orchestration requirements. Customers requiring faster recovery can implement warm standby architectures using multiple regions with pre-created environments and automated failover mechanisms.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/mwaa/latest/userguide/mwaa-quotas.html>

FAQ: <https://aws.amazon.com/managed-workflows-for-apache-airflow/faqs/>

Technical Requirements

Service Documentation: <https://docs.aws.amazon.com/mwaa/latest/userguide/what-is-mwaa.html>

User Guide: <https://docs.aws.amazon.com/mwaa/latest/userguide/get-started.html>

API Reference: <https://docs.aws.amazon.com/mwaa/latest/API/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-workflows-for-apache-airflow/pricing/>

Cost Optimization

Amazon MWAA offers unique cost optimization through MWAA Serverless with pay-per-use pricing based on actual task execution time, eliminating costs for idle infrastructure. The traditional MWAA model supports automatic scaling of workers and webserver to match workload demands, preventing over-provisioning:

- Customers can optimize costs by choosing appropriate environment classes (micro, small, medium, large), configuring minimum/maximum worker counts, and utilizing the micro environment for development and testing
- The service eliminates operational overhead costs of managing self-hosted Airflow infrastructure including patching, monitoring, and scaling
- Cost control is enhanced through precise resource allocation and the ability to schedule maintenance windows to minimize disruption

Savings Considerations

Key cost savings opportunities include using MWAA Serverless for sporadic workloads to avoid paying for idle time, selecting the smallest appropriate environment class like mw1.micro for development and light production workloads, and implementing proper worker autoscaling configurations to minimize over-provisioning. Customers save significantly on operational costs by eliminating the need for dedicated infrastructure management teams, automated patching, and 24/7 monitoring:

- Right-sizing worker counts based on actual workload patterns, using efficient DAG designs to reduce task execution time, and leveraging AWS service integrations instead of external tools contribute to cost optimization
- The managed service model eliminates capital expenses for infrastructure while providing predictable operational expenses based on actual usage patterns