

Service Name

AWS Step Functions

Service Overview

AWS Step Functions is a serverless workflow orchestration service that enables users to create and manage complex application workflows, also known as state machines. It coordinates multiple AWS services into serverless workflows by breaking down applications into multiple steps, adding flow logic, and tracking inputs and outputs between steps. Step Functions uses Amazon States Language (ASL) to define workflows as a series of event-driven steps, where each step is a state. The service offers two workflow types: Standard workflows for long-running, auditable workflows that can run up to one year with exactly-once execution, and Express workflows for high-event-rate workloads that can run up to five minutes with at-least-once execution. Step Functions maintains application state, provides built-in error handling, supports parallel processing, and integrates with over 220 AWS services and 10,000+ APIs.

Key Use Cases

- **Microservice orchestration:** Combining Lambda functions to build web-based applications with or without human approval steps
- **Security and IT automation:** Orchestrating security incident response workflows and automating responses to operational events in AWS accounts
- **Data processing and ETL orchestration:** Building data processing pipelines for streaming data and automating ETL processes with parallel job execution
- **Machine learning operations:** Running ETL jobs and building, training, and deploying machine learning models using Amazon SageMaker integration
- **Media processing:** Extracting data from PDFs or images and building serverless video transcoding pipelines with massive parallelization

Features

- **Visual Workflow Studio:** Drag-and-drop interface with Design, Code, and Config modes for creating and editing workflows visually or through Amazon States Language
- **Two Workflow Types:** Standard workflows for long-running auditable processes (up to 1 year) and Express workflows for high-volume short-running processes (up to 5 minutes)
- **Service Integration Patterns:** Three integration patterns: Request Response, Run a Job (.sync), and Wait for Callback (.waitForTaskToken) supporting 220+ AWS services and 10,000+ APIs

- **Built-in Error Handling:** Automatic retry mechanisms, dead-letter queues, error notifications, and exception handling with Catch and Retry states
- **State Management:** Maintains workflow state with execution history, input/output tracking, and ability to resume after interruptions
- **Parallel Processing:** Map state in Distributed mode for large-scale parallel processing and Parallel state for concurrent execution branches
- **Monitoring and Observability:** Integration with CloudWatch for logging, X-Ray for tracing, and visual execution history with detailed event tracking

Value Proposition

AWS Step Functions provides a serverless, visual approach to orchestrating complex workflows without managing infrastructure. Unlike traditional workflow engines or custom orchestration code, Step Functions eliminates the need to write connection logic between services, provides built-in error handling and retry mechanisms, and offers pay-per-use pricing:

- The service accelerates application development through its intuitive visual interface and pre-built integrations with 220+ AWS services
- Step Functions ensures high availability and fault tolerance with automatic scaling, maintains detailed execution history for auditing and debugging, and supports both synchronous and asynchronous execution patterns
- The service simplifies modernization of distributed applications by providing a centralized orchestration layer that reduces complexity and improves maintainability while offering enterprise-grade security and compliance features

Service Integration

Step Functions integrates seamlessly with core AWS compute services including Lambda for serverless functions, ECS and Fargate for containerized workloads, and AWS Batch for batch processing jobs. Data services integration includes DynamoDB for NoSQL operations, S3 for object storage, RDS for relational databases, and Amazon Redshift for data warehousing:

- Machine learning integrations cover Amazon SageMaker for model training and inference, Amazon Bedrock for generative AI, and Amazon Comprehend for natural language processing
- Messaging and event services include SNS for notifications, SQS for queuing, EventBridge for event routing, and API Gateway for REST APIs
- Additional integrations span monitoring services like CloudWatch, security services like IAM, and developer tools like CodeBuild and CodePipeline, providing comprehensive orchestration capabilities across the AWS ecosystem

Onboarding and Offboarding

Customers can get started with Step Functions through the AWS Console's built-in Hello World workflow tutorial, which takes 20-30 minutes to complete. The process begins by accessing the Step Functions console and selecting 'Get Started' followed by 'Run Hello World' to create a simple workflow using Workflow Studio's drag-and-drop interface:

- The tutorial covers creating state machines using visual design mode, understanding flow states and task states, viewing Amazon States Language (ASL) code, and integrating AWS services like Amazon Comprehend
- Customers learn to configure IAM roles and policies, execute workflows with input data, and monitor execution details
- The onboarding includes hands-on experience with the three Workflow Studio modes: Design for visual editing, Code for direct ASL editing, and Config for state machine configuration including logging, tracing, and security settings

Getting Started Guide: <https://docs.aws.amazon.com/step-functions/latest/dg/getting-started.html>

Data Removal

Step Functions data removal involves deleting state machine definitions, execution history, and associated resources. Customers can delete state machines through the console, CLI, or API, which removes the workflow definition and stops new executions. Existing execution history is retained according to service quotas but can be managed through CloudWatch Logs retention policies if logging is enabled. For complete offboarding, customers must also remove associated IAM roles, CloudWatch log groups, and any Lambda functions or other resources created specifically for the workflows to ensure no residual charges or data retention.

Backup/Restore and Disaster Recovery

Step Functions provides inherent resilience through its serverless architecture with automatic multi-AZ deployment and built-in fault tolerance. State machine definitions are automatically backed up as part of the AWS control plane. Execution state is maintained during service interruptions, allowing workflows to resume from the last successful state. For disaster recovery, customers can export state machine definitions as JSON/YAML and redeploy in alternate regions. Cross-region replication of dependent resources like Lambda functions and DynamoDB tables ensures complete DR coverage, while execution history can be preserved through CloudWatch Logs cross-region replication.

Backup Capabilities

Step Functions automatically maintains state machine definitions within the AWS control plane infrastructure. The service does not directly integrate with AWS Backup as it's a serverless orchestration service rather than a data storage service:

- However, customers can programmatically export state machine definitions using APIs for backup purposes
- The service maintains execution history up to quota limits, and when CloudWatch logging is enabled, execution logs are preserved according to CloudWatch retention policies, providing audit trails and debugging information for recovery scenarios

Disaster Recovery

Step Functions supports disaster recovery through its multi-region availability and serverless architecture. The service does not directly integrate with AWS Elastic Disaster Recovery as it's an orchestration service rather than compute infrastructure:

- However, customers can implement DR by maintaining state machine definitions in multiple regions and replicating dependent resources
- The service's stateless nature and JSON-based definitions enable quick recovery by redeploying workflows in alternate regions, while execution state resilience ensures workflows can continue after regional failures

Recovery Objectives

Step Functions helps customers achieve aggressive RPO and RTO objectives through its serverless architecture and built-in resilience. For RPO, the service maintains real-time state information and can resume workflows from the last successful state, minimizing data loss. For RTO, the multi-AZ deployment provides automatic failover capabilities within regions, while cross-region deployment of state machines enables rapid recovery during regional outages. The service's integration with other AWS services allows customers to design workflows that incorporate backup and recovery patterns, such as automated failover procedures and data replication workflows.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/general/latest/gr/step-functions.html>

FAQ: <https://aws.amazon.com/step-functions/faqs/>

Technical Requirements

Service Documentation: <https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html>

User Guide: <https://docs.aws.amazon.com/step-functions/latest/dg/getting-started.html>

API Reference: <https://docs.aws.amazon.com/step-functions/latest/apireference/index.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/step-functions/pricing/>

Cost Optimization

Step Functions offers unique cost optimization through workflow type selection, where Express workflows cost significantly less than Standard workflows for high-volume, short-duration use cases. Customers can nest Express workflows within Standard workflows to optimize costs for mixed workloads:

- The service supports cost optimization through efficient state design, minimizing state transitions which are the primary billing unit for Standard workflows
- Using S3 ARNs instead of passing large payloads reduces state transition costs and avoids payload size limits
- Map state in Distributed mode provides cost-effective parallel processing for large-scale operations
- Proper timeout configuration prevents runaway executions and unnecessary charges, while workflow consolidation reduces overall state transitions and associated costs

Savings Considerations

Primary cost savings come from choosing appropriate workflow types, with Express workflows offering up to 90% cost reduction for high-volume, short-duration scenarios compared to Standard workflows. Optimizing state machine design by reducing unnecessary state transitions, combining multiple operations into single states, and using efficient data passing patterns significantly reduces costs:

- Leveraging the free tier of 4,000 state transitions per month for Standard workflows provides cost savings for small to medium workloads
- Using Step Functions instead of managing custom orchestration infrastructure eliminates server costs, reduces operational overhead, and provides pay-per-use billing
- The service's integration capabilities reduce development time and maintenance costs compared to building custom workflow solutions, while automatic scaling eliminates over-provisioning costs typical in traditional workflow engines