Service Name

AWS CodePipeline

Service Overview

AWS CodePipeline is a fully managed continuous delivery service that automates and models the software release process. It enables users to visualize, configure, and automate the different stages of a software release workflow, from source code changes to production deployment. CodePipeline detects changes in source repositories, automatically builds code, runs tests, and deploys applications to production environments. The service orchestrates the entire release process through a series of configurable stages and actions, providing a consistent and reliable way to release software changes rapidly with high frequency and quality.

Key Use Cases

- Continuous Integration/Continuous Deployment (CI/CD): Automate build, test, and deployment workflows for applications across development, staging, and production environments
- Multi-stage application deployment: Deploy applications to Amazon EC2 instances, AWS Elastic Beanstalk, Amazon ECS containers, and AWS Lambda functions with automated progression through environments
- Infrastructure as Code automation: Automate deployment and management of AWS resources using AWS CloudFormation templates with version control and rollback capabilities

Features

- **Visual Pipeline Modeling**: Model release workflows using visual pipelines with multiple configurable stages and actions
- **AWS Service Integrations**: Native integration with AWS CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, ECS, Lambda, and CloudFormation
- **Stage-level Conditions**: Configure entry, success, and failure conditions with automatic rollback capabilities for enhanced release control
- **Pipeline Variables**: Use pipeline-level and action-level variables for dynamic configuration and parameterized deployments
- **Parallel and Sequential Execution**: Execute actions in parallel within stages or sequentially across stages with flexible execution modes
- **Cross-Region Support**: Deploy applications across multiple AWS regions using cross-region actions and artifact replication
- **Third-party Integrations**: Integration with GitHub, Jenkins, and other third-party tools through prebuilt and custom plugins

- **Manual Approval Gates**: Include manual approval actions to control progression between pipeline stages

Value Proposition

CodePipeline provides a fully managed, scalable CI/CD solution that eliminates the need to maintain custom deployment infrastructure. It offers deep integration with the AWS ecosystem, enabling seamless automation from source to production with minimal operational overhead:

- The service provides enterprise-grade features like stage-level conditions, automatic rollbacks, and cross-account deployments while maintaining cost-effectiveness through its pay-per-use pricing model
- Unlike self-managed CI/CD solutions, CodePipeline automatically scales, provides high availability, and integrates natively with AWS security and compliance features, reducing time-to-market and operational complexity for development teams

Service Integration

CodePipeline integrates extensively with core AWS developer services including AWS CodeCommit for source control, AWS CodeBuild for build and test automation, and AWS CodeDeploy for application deployment. It connects with compute services like Amazon EC2, AWS Lambda, Amazon ECS, and AWS Elastic Beanstalk for deployment targets:

- The service integrates with AWS CloudFormation for infrastructure-as-code deployments and Amazon S3 for artifact storage
- Additional integrations include Amazon SNS for notifications, AWS IAM for security and access control, Amazon CloudWatch for monitoring and logging, and AWS CloudTrail for API auditing and compliance tracking

Onboarding and Offboarding

Customers can get started by creating an AWS account and setting up an administrative user with CodePipeline permissions. The service offers a simplified getting started experience with pre-built pipeline templates for Build, Automation, and Deployment use cases:

- Users can create their first pipeline using the AWS Management Console, selecting a source repository (CodeCommit, GitHub, or S3), adding build and deployment stages, and configuring the target environment
- The new pipeline templates allow quick creation of fully configured pipelines ready to run
- Tutorials are available for common scenarios like deploying to EC2 instances, ECS containers, and Lambda functions

Getting Started Guide:
https://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-codepipeline.html

Data Removal

To offboard from CodePipeline, customers must delete all pipelines, which automatically removes pipeline execution history and associated metadata. Pipeline artifacts stored in Amazon S3 buckets must be manually deleted as they are retained independently. Custom actions, webhooks, and connections should be removed through the console or API. IAM roles and policies created for CodePipeline can be deleted once no longer needed. The service retains pipeline execution history for 12 months before automatic deletion, so customers should export any required historical data before offboarding.

Backup/Restore and Disaster Recovery

CodePipeline does not provide traditional backup capabilities as it is a workflow orchestration service that manages deployment processes rather than persistent data. Pipeline configurations can be exported and recreated using AWS CloudFormation templates or the AWS CLI for disaster recovery purposes. The service automatically maintains pipeline execution history for 12 months. For disaster recovery, customers should replicate pipeline configurations across regions and ensure source repositories and deployment targets have appropriate backup and recovery mechanisms in place.

Backup Capabilities

CodePipeline itself does not have backup capabilities and does not integrate with AWS Backup as it manages workflow orchestration rather than persistent data storage. Pipeline configurations serve as the primary ‘data’ and can be version-controlled through Infrastructure as Code practices using CloudFormation or CDK. Customers should backup pipeline definitions, source code repositories, and deployment artifacts separately using appropriate backup services for each component.

Disaster Recovery

CodePipeline does not directly integrate with AWS Elastic Disaster Recovery as it orchestrates deployment workflows rather than managing application infrastructure. However, pipelines can be configured to deploy across multiple regions and support disaster recovery strategies for the applications they deploy. Customers should implement cross-region pipeline configurations and ensure that source repositories, build artifacts, and deployment targets have appropriate disaster recovery mechanisms configured independently.

Recovery Objectives

CodePipeline helps achieve faster Recovery Time Objectives (RTO) by automating deployment processes and enabling rapid application updates across multiple

environments. The service supports automated rollback capabilities through stage-level conditions, reducing Recovery Point Objectives (RPO) by quickly reverting failed deployments. Cross-region deployment capabilities enable geographic distribution for improved disaster recovery posture. However, specific RTO/RPO targets depend on the underlying services and applications being deployed rather than CodePipeline itself.

Service Constraints

Service Quotas: https://docs.aws.amazon.com/general/latest/gr/codepipeline.html

FAQ: https://aws.amazon.com/codepipeline/faqs/

Technical Requirements

Service Documentation:
https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html

User Guide: https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html

API Reference:
https://docs.aws.amazon.com/codepipeline/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/codepipeline/pricing/

Cost Optimization

CodePipeline offers two pricing models: V1 pipelines at $1.00 per active pipeline per month (flat rate) and V2 pipelines at $0.002 per action execution minute (usage-based). Cost optimization strategies include choosing the appropriate pipeline type based on usage patterns, utilizing the AWS Free Tier (one free V1 pipeline monthly, 100 free V2 action execution minutes), and minimizing unnecessary pipeline executions through efficient triggering mechanisms. Organizations can reduce costs by consolidating similar workflows, optimizing build times, and using efficient artifact storage practices in S3.

Savings Considerations

Primary cost savings come from eliminating the need to maintain self-managed CI/CD infrastructure, reducing operational overhead and staffing requirements. The pay-per-use model for V2 pipelines provides cost efficiency for variable workloads compared to maintaining always-on build servers:

- Organizations save on licensing costs for third-party CI/CD tools and reduce time-to-market through automated deployments

- Additional savings result from improved deployment reliability, reduced rollback costs, and decreased manual intervention requirements
- The service's integration with other AWS services can provide economies of scale and volume discounts across the broader AWS usage portfolio