

Service Name

AWS CodeCommit

Service Overview

AWS CodeCommit is a fully managed source control service hosted by Amazon Web Services that enables organizations to privately store and manage assets such as documents, source code, and binary files in the cloud. CodeCommit eliminates the need to manage your own source control system or worry about scaling infrastructure by providing secure, highly scalable private Git repositories. The service supports standard Git functionality and works seamlessly with existing Git-based tools, enabling teams to collaborate through features like pull requests, branching, merging, and code reviews while ensuring high availability, durability, and enterprise-grade security.

Key Use Cases

- **Source code management:** Store and version control application source code with full Git functionality for software development teams
- **CI/CD pipeline integration:** Serve as the source repository for automated build, test, and deployment workflows using AWS CodePipeline, CodeBuild, and CodeDeploy
- **Collaborative development:** Enable team collaboration through pull requests, code reviews, branching, and merging with fine-grained access control
- **Asset management:** Store and manage documents, binaries, and other development assets alongside source code in secure private repositories
- **Enterprise compliance:** Meet regulatory and compliance requirements with encryption, audit trails, and IAM-based access controls for secure code storage

Features

- **Private Git Repositories:** Unlimited private Git repositories with full Git functionality including branching, merging, and tagging
- **Pull Requests and Code Reviews:** Built-in pull request workflows with approval rules, code review capabilities, and team collaboration features
- **Encryption and Security:** Data encrypted in transit via HTTPS/SSH and at rest using AWS KMS with fine-grained IAM access controls
- **High Availability and Durability:** Repositories stored across multiple Availability Zones using Amazon S3 and DynamoDB for enterprise-grade reliability
- **Event Notifications and Triggers:** Repository event notifications via Amazon SNS and configurable triggers to invoke AWS Lambda functions
- **Cross-Account Access:** Support for cross-account repository sharing and federated access using IAM roles and policies

- **Multiple Connection Methods:** Access via HTTPS Git credentials, SSH keys, git-remote-codecommit, and AWS CLI credential helper

Value Proposition

Organizations choose AWS CodeCommit for its deep integration with AWS services and unique enterprise capabilities that differentiate it from other Git hosting solutions. CodeCommit provides seamless IAM integration for fine-grained access control, VPC endpoint support for private network access, and comprehensive AWS CloudTrail logging for audit and compliance requirements:

- The service eliminates infrastructure management overhead while offering enterprise-grade security with AWS KMS encryption and FIPS 140-2 compliance
- CodeCommit's native integration with AWS CodePipeline, CodeBuild, and CodeDeploy enables frictionless CI/CD workflows within the AWS ecosystem, making it ideal for organizations already invested in AWS services or those requiring strict compliance and security controls

Service Integration

CodeCommit integrates natively with core AWS DevOps and development services to enable complete CI/CD workflows. Primary integrations include AWS CodePipeline for automated build and deployment pipelines, AWS CodeBuild for continuous integration and testing, and AWS CodeDeploy for application deployments:

- The service leverages AWS IAM for access control and authentication, AWS KMS for encryption key management, and Amazon CloudWatch Events for monitoring and notifications
- CodeCommit also integrates with Amazon SNS for event notifications, AWS Lambda for custom triggers, AWS CloudTrail for audit logging, and Amazon CodeGuru Reviewer for automated code reviews
- Additional integrations include AWS Amplify for frontend deployments, AWS Cloud9 for cloud-based development environments, and AWS Elastic Beanstalk for web application deployments

Onboarding and Offboarding

Getting started with CodeCommit requires setting up AWS credentials and configuring local Git access through multiple supported methods. Customers begin by creating an AWS account and configuring IAM permissions, then choose from connection methods including HTTPS Git credentials (simplest), SSH keys, git-remote-codecommit (recommended for federated access), or AWS CLI credential helper:

- The setup process involves creating repositories through the AWS Management Console, AWS CLI, or APIs, then cloning them locally using standard Git commands

- CodeCommit provides comprehensive tutorials covering repository creation, local setup, making commits, creating branches, and configuring team access permissions
- The service offers multiple getting started guides depending on Git familiarity and connection preferences, ensuring smooth onboarding for both new and experienced Git users

Getting Started Guide:

<https://docs.aws.amazon.com/codecommit/latest/userguide/getting-started.html>

Data Removal

CodeCommit offboarding involves deleting repositories through the AWS Management Console, AWS CLI, or API operations. Before deletion, customers should backup repository data using standard Git clone commands or AWS Backup service snapshots. Once repositories are deleted, all associated data including commit history, branches, tags, and metadata are permanently removed from CodeCommit storage. Customers can export repository data by cloning to local systems or pushing to alternative Git hosting services before deletion. The service provides confirmation prompts and supports programmatic deletion for bulk operations during migration scenarios.

Backup/Restore and Disaster Recovery

CodeCommit provides built-in high availability and disaster recovery through automatic replication across multiple Availability Zones within a region. The service stores repository data redundantly in Amazon S3 and Amazon DynamoDB for durability and availability. While CodeCommit itself is highly durable, customers can create additional backups using standard Git operations like cloning repositories or by integrating with AWS Backup service for automated snapshot management. Repository data can be restored from local clones or backup snapshots, and the distributed nature of Git provides inherent backup capabilities through multiple repository copies.

Backup Capabilities

CodeCommit offers native backup capabilities through AWS Backup service integration, enabling automated repository snapshots and centralized backup management. Customers can create manual backups using standard Git clone operations to preserve complete repository history, branches, and tags:

- The service's architecture provides inherent backup through automatic replication across multiple Availability Zones
- Repository data can be exported and backed up to external systems using Git operations, and deleted repositories can be restored from backups if created before deletion

Disaster Recovery

CodeCommit supports disaster recovery through its multi-Availability Zone architecture and integration with AWS disaster recovery services. The service automatically replicates data across multiple AZs for high availability and fault tolerance:

- While CodeCommit doesn't directly integrate with AWS Elastic Disaster Recovery, customers can implement DR strategies using Git's distributed nature by maintaining repository mirrors across regions or external systems
- Repository data can be quickly restored from local clones or backup snapshots in disaster scenarios

Recovery Objectives

CodeCommit helps customers achieve low Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO) through its highly available architecture and Git's distributed nature. The service's automatic multi-AZ replication provides near-zero RPO for committed changes, while the ability to quickly clone repositories from backup sources enables rapid recovery. Customers can maintain multiple repository copies across regions or external systems to further reduce RTO. The service's integration with AWS Backup enables point-in-time recovery capabilities for more granular RPO requirements.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/codecommit/latest/userguide/limits.html>

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

Technical Requirements

Service Documentation:

<https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html>

User Guide: <https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html>

API Reference:

<https://docs.aws.amazon.com/codecommit/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/codecommit/pricing/>

Cost Optimization

CodeCommit offers unique cost optimization through its usage-based pricing model charging only for active users (those performing Git operations monthly) rather than total users or repositories. The service includes a permanent free tier for up to 5 active users per month, unlimited private repositories, and no charges for data transfer or Git operations beyond storage and request quotas:

- Cost optimization strategies include managing active user counts by consolidating access through service accounts, using the service's unlimited repository feature to avoid per-repository fees, and leveraging the free storage and request allowances
- Organizations can optimize costs by timing development activities and using batch operations to minimize active user months

Savings Considerations

CodeCommit provides significant cost savings compared to self-managed Git hosting solutions by eliminating infrastructure provisioning, maintenance, and scaling costs. The service's operational model reduces total cost of ownership through automatic backup, security patching, and high availability without additional charges:

- Organizations save on licensing fees for commercial Git hosting platforms while gaining enterprise features like encryption, compliance certifications, and AWS service integration
- The pay-per-active-user model ensures costs scale with actual usage rather than provisioned capacity
- Additional savings come from reduced administrative overhead, faster deployment cycles through AWS service integration, and elimination of third-party tool licensing for CI/CD workflows when combined with other AWS DevOps services