AWS CodeDeploy Deployment Strategies

Overview

AWS CodeDeploy automates application deployments to various compute platforms including EC2, Lambda, and ECS. It offers multiple deployment strategies to minimize downtime, manage risk, and ensure reliable software releases.

Supported Platforms

- Amazon EC2 instances
- On-premises servers
- AWS Lambda functions
- Amazon ECS services

In-Place Deployment

- Application stopped on each instance in deployment group
- Latest revision installed on same instances
- New version started and validated
- Load balancer can deregister instances during deployment
- Only available for EC2/On-Premises platform
- Cost-effective but may have brief downtime

Blue/Green Deployment

- Provision entirely new set of servers (green environment)
- Current blue environment remains active during deployment
- Traffic switched from blue to green after validation
- Old blue environment decommissioned after successful deployment
- Minimizes downtime and simplifies rollback
- Only deployment type for Lambda and ECS
- Also supported for EC2/On-Premises
- Requires additional infrastructure during transition

Rolling Deployment

- Gradually replaces instances with new version
- Deploys to subset of instances at a time
- Reduces downtime compared to in-place
- Slower rollout process
- Maintains partial availability during deployment

Canary Deployment

- Specialized form of blue/green deployment
- New version rolled out to small percentage of users first
- Early detection of issues with minimal impact
- Gradually shift more traffic to new version
- Controlled and gradual rollout
- Applicable to Lambda and ECS

Deployment Configurations

- **AllAtOnce:** Deploy to all instances simultaneously
- **HalfAtATime:** Deploy to 50% of instances at once
- **OneAtATime:** Deploy to one instance at a time
- **Custom:** Define your own deployment configuration

Strategy Selection: Choose deployment strategy based on acceptable downtime, rollback requirements, and risk tolerance. Blue/green offers safest rollback, canary enables gradual testing, and in-place is most cost-effective.