AppSpec 'hooks' Section for Amazon ECS Deployment

Overview

AWS CodeDeploy uses AppSpec files to manage ECS deployments. The hooks section enables integration of custom logic via AWS Lambda functions at various stages of the deployment lifecycle, allowing for validation, testing, and custom actions.

AppSpec File Structure

For ECS deployments, the AppSpec file is in YAML or JSON format and specifies the ECS task definition, load balancer information, and optional Lambda functions for lifecycle hooks.

ECS Deployment Lifecycle Hooks

- **BeforeInstall:** Run tasks before replacement task set is created
- **AfterInstall:** Run tasks after replacement task set is created
- **AfterAllowTestTraffic:** Run tests after test traffic is routed
- **BeforeAllowTraffic:** Run tasks before production traffic shift
- **AfterAllowTraffic:** Run tasks after production traffic shift

Hook Configuration

- Each hook specifies a Lambda function ARN

- Lambda functions receive deployment lifecycle event data
- Functions must call CodeDeploy to continue or fail deployment
- Timeout can be configured for each hook (default 1 hour)
- Hooks enable automated validation and testing

Common Use Cases

- **Health Checks:** Verify new task set is healthy before traffic shift
- **Integration Tests:** Run automated tests against new version
- **Smoke Tests:** Basic functionality validation
- **Monitoring Setup:** Configure CloudWatch alarms for new tasks
- **Notifications:** Send alerts about deployment progress
- **Rollback Logic:** Custom conditions for automatic rollback

Best Practices

- Keep Lambda functions lightweight and focused
- Implement proper error handling in hook functions
- Set appropriate timeouts for each hook
- Log detailed information for troubleshooting
- Test hooks thoroughly in non-production environments
- Use IAM roles with least privilege for Lambda functions

Deployment Validation: AppSpec hooks for ECS enable automated validation and testing at critical points in the deployment lifecycle, ensuring safe and reliable container deployments with custom logic.