

Formação AWS Cloud Foundations

Alexsandro Lechner

Arquiteto de Soluções AWS

linkedin.com/in/alexandrolechner

Conteúdo Programático

□ Módulo 11 Automação e DevOps na AWS

- Como automatizar tarefas
- Executando tarefas automatizadas com Lambda Function
- Criação de recursos com Terraform
- O que é DevOps
- **Aplicando conceito de DevOps na AWS**
- Ferramentas da AWS para DevOps

Automação e DevOps na AWS

Aplicando conceito de DevOps na AWS

DevOps na AWS

O que é DevOps?

- DevOps é um conjunto de práticas que une desenvolvimento (Dev) e operações (Ops) para automatizar e melhorar processos.
- Objetivo: entregar software com mais rapidez, qualidade e confiabilidade.
- A AWS oferece ferramentas para cada etapa do ciclo de vida DevOps.



DevOps na AWS

Principais Ferramentas DevOps da AWS

Ferramentas AWS	Uso
AWS CodeCommit	Repositório Git gerenciado
AWS CodeBuild	Compilação e testes automatizados
AWS CodeDeploy	Automação da implantação
AWS CodePipeline	Orquestração do pipeline CI/CD
AWS CloudFormation	Infraestrutura como Código (IaC)
Amazon CloudWatch	Monitoramento e logs
AWS IAM	Gestão de permissões e segurança

DevOps na AWS

Pipeline CI/CD na AWS

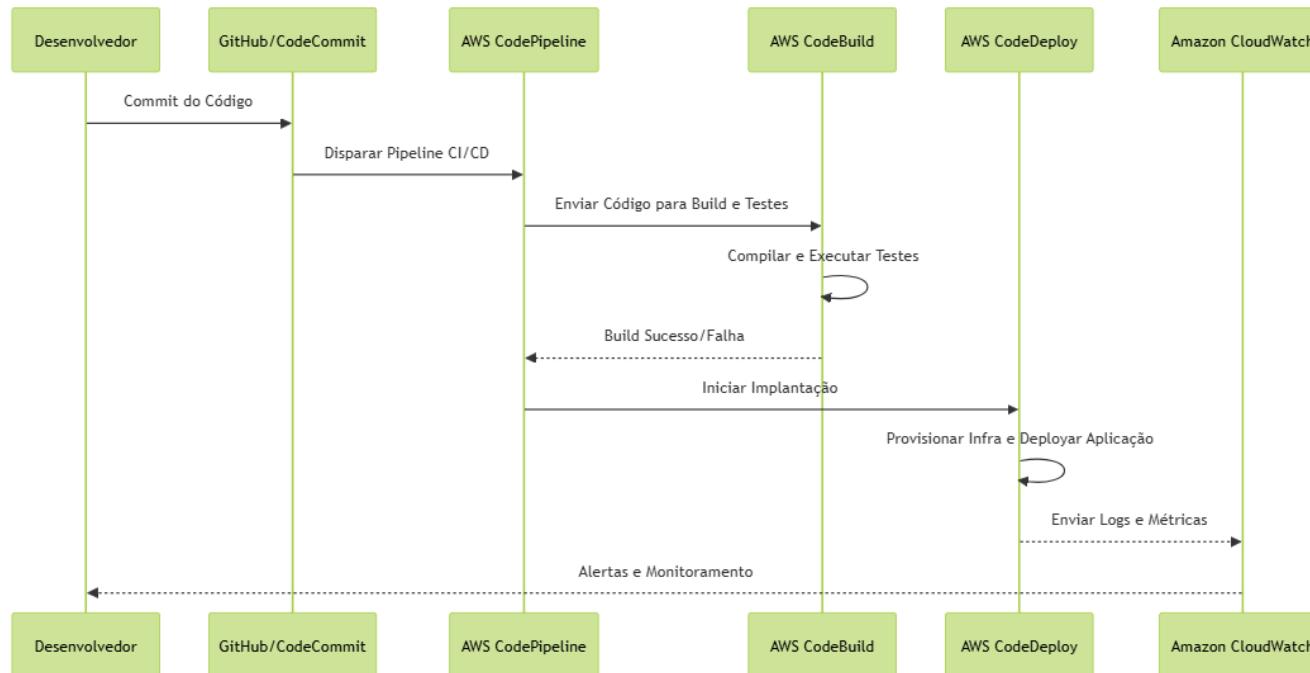
- CI/CD significa Continuous Integration (CI) e Continuous Deployment (CD)
- CI: Automatiza testes e builds sempre que há uma mudança no código
- CD: Automatiza a entrega do código em produção
- Benefícios: Rapidez, consistência e menos erros manuais.

Etapas típicas de um pipeline CI/CD:

- > Commit do código → GitHub / CodeCommit
- > Build e Testes → CodeBuild
- > Implantação → CodeDeploy
- > Monitoramento → CloudWatch

DevOps na AWS

Principais Ferramentas DevOps da AWS



DevOps na AWS

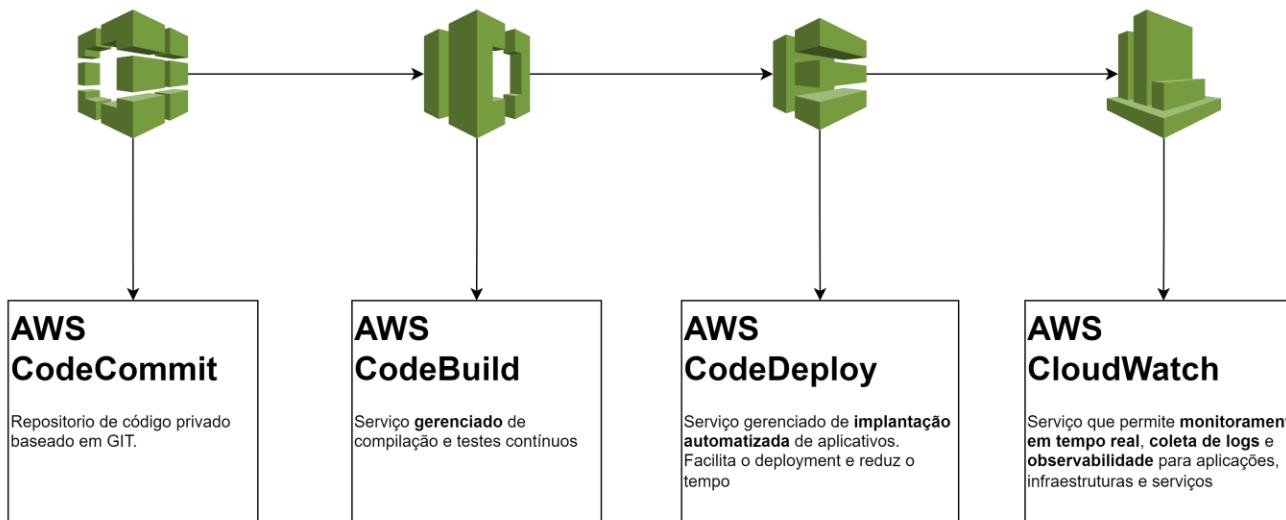
Desenvolvimento de Software com CI/CD na AWS

No desenvolvimento tradicional de software, um pipeline CI/CD permite que os times implementem mudanças rapidamente, garantindo qualidade e confiabilidade.

- 1 Commit do código → O desenvolvedor faz um push do código para o **AWS CodeCommit**.
- 2 Build e Testes → O **AWS CodeBuild** compila o código e executa testes automatizados.
- 3 Implantação → Se os testes passarem, o **AWS CodeDeploy** implanta o código em servidores EC2, containers ECS ou funções Lambda.
- 4 Monitoramento → O **Amazon CloudWatch** coleta logs e métricas da aplicação para análise de desempenho.

DevOps na AWS

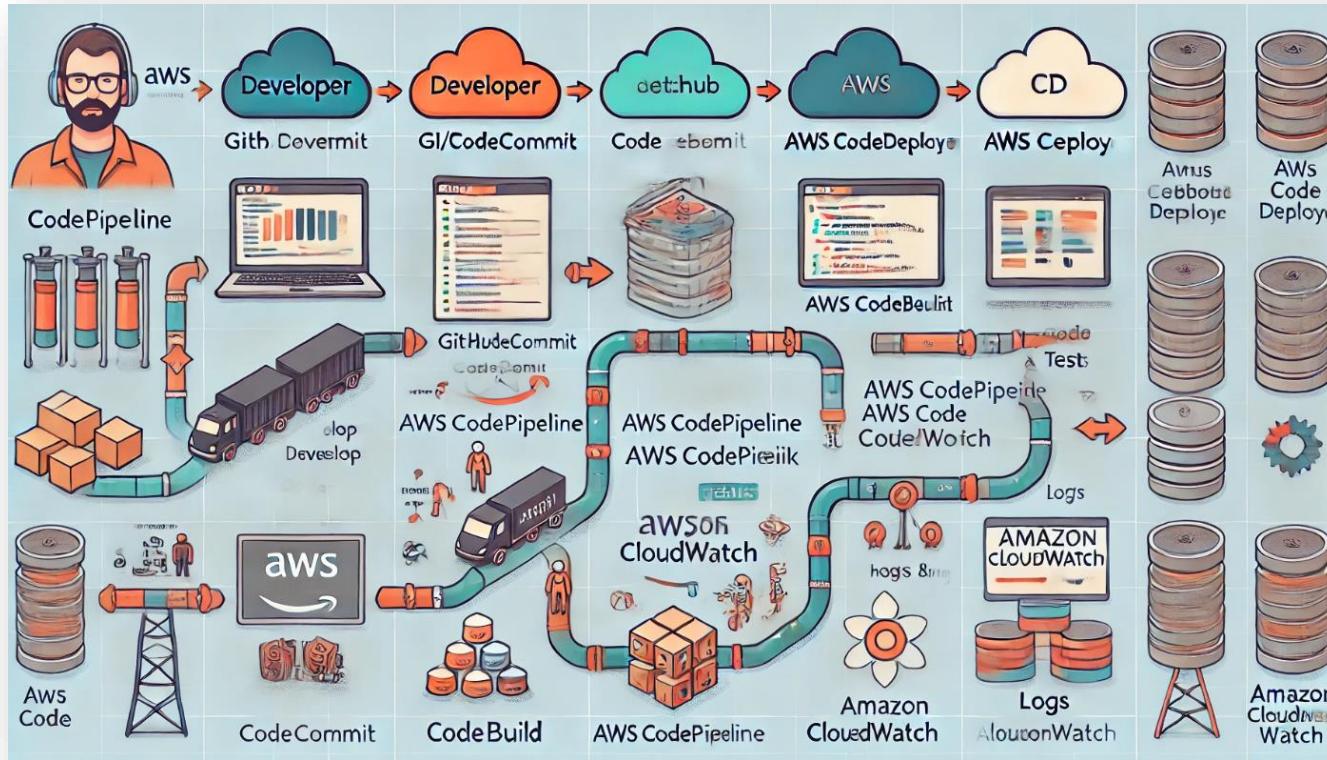
Principais Ferramentas DevOps da AWS



Benefícios:

- ✓ Deployes automatizados e sem erros
- ✓ Testes contínuos garantem qualidade
- ✓ Redução do tempo entre desenvolvimento e produção

DevOps na AWS



DevOps na AWS

Cenário:

Uma equipe de desenvolvimento precisa automatizar o deploy de uma aplicação Python Flask na AWS, garantindo controle de versão, build, testes, deploy automatizado e monitoramento. Usaremos AWS CodeCommit, CodeBuild, CodeDeploy e CloudWatch dentro do LocalStack para simular a infraestrutura AWS.

Arquitetura do Pipeline

- 1** Commit do Código → CodeCommit (GitHub)
- 2** Build e Testes → CodeBuild
- 3** Implantação → CodeDeploy (EC2 simulado ou Docker)
- 4** Monitoramento → CloudWatch

DevOps na AWS

1 Criar Tickets

- AWS Systems Manager OpsCenter para registrar problemas, bugs ou tarefas
- Outras ferramentas: JIRA, ServiceNow ou GitHub Issues, integrando ao CodePipeline

2 Associar o Ticket ao Release

- No AWS CodePipeline, utilize tags para associar um ticket específico ao deploy
- O ticket pode ser referenciado dentro do build via AWS CodeBuild

3 Aprovação Manual antes do Deploy

- Configure um Manual Approval Stage no CodePipeline.
- O pipeline só continua após aprovação no console.

4 Monitoramento e Auditoria

- AWS CloudTrail pode ser usado para rastrear ações dentro do pipeline.
- O AWS CodeDeploy Reports permite visualizar o histórico de releases.