

FIAP
CYBER SECURITY GOVERNANCE & MANAGEMENT
67SEG

TRABALHO DE EXECUÇÃO DOS TUTORIAIS DE IMPLANTAÇÃO DE UMA ESTEIRA DEVOPS/DEVSECOPS NA AWS

Tutoriais 1 e 2

EDSON VITORIO	RM349992
GABRIELA ARAUJO	RM348132
MARCOS BELEZI	RM349545
PAUL HARO	RM348682

São Paulo, 18 Outubro de 2023

Tutorial de Esteira de DevOps para Ferramentas de Teste

1. Tutorial: Configurar uma esteira de DevOps na AWS com ferramentas de teste é fundamental para garantir a qualidade e a confiabilidade do seu código antes de implantá-lo em produção.

Para execução do tutorial foram realizados através da gravação de dois vídeos, um que executam os passos do 1 ao 7 e outro somente o passo 4, conforme o link a seguir:

Link dos Vídeos

https://drive.google.com/drive/folders/15WCdrBgSCPM28JMcQ2hvt5ayl8gqtBZ?usp=share_link

Implantação Prática de uma Esteira de DevOps na AWS com Python e Ênfase em Ferramentas de Infraestrutura como Código (IaC)

1. Este tutorial tem o objetivo de configurar uma esteira de DevOps na Amazon Web Services (AWS) com foco em ferramentas de Infraestrutura como Código (IaC). Usar serviços como AWS CloudFormation, AWS CodePipeline, AWS CodeBuild e Python como linguagem de programação. Automatizar a construção, teste, implantação e atualização de infraestrutura e aplicativos Python sempre que houver uma alteração no repositório Git.

Para execução do tutorial foi através da gravação do vídeo dos passos 1 ao 5 e a resposta do questionário das 10 perguntas..

Link do Vídeo

https://drive.google.com/drive/folders/15YZcXXgYNlrgFUeTazYwXs5lpk_WARgs?usp=share_link

QUESTIONÁRIO

- 1. No contexto deste estudo de caso, explique o que significa Infraestrutura como Código (IaC) e por que é uma prática importante no DevOps.**

Infraestrutura como Código (IaC) é uma prática de administração da infraestrutura de TI por meio de código de programação. Essa abordagem automatiza a configuração e o provisionamento de recursos, proporcionando agilidade, consistência, rastreabilidade e eficiência na utilização de recursos. No contexto do DevOps, essa prática desempenha um papel fundamental ao acelerar o ciclo de desenvolvimento e operação de software.

- 2. Descreva os benefícios de usar ferramentas de IaC, como o AWS CloudFormation, na criação e gerenciamento de recursos de infraestrutura na AWS.**

Ferramentas de Infraestrutura como Código (IaC), como o AWS CloudFormation, proporcionam uma série de benefícios essenciais, incluindo Automatização e Agilidade, Consistência e Padronização, Escalabilidade, Rastreabilidade, Eficiência de custos e conformidade. Essas ferramentas simplificam significativamente a criação e o gerenciamento de recursos na AWS, integrando-se harmoniosamente ao DevOps e viabilizando uma infraestrutura mais eficiente e confiável.

- 3. Quais são os componentes principais do aplicativo Python utilizado neste estudo de caso, e como a infraestrutura relacionada a esses componentes é definida usando AWS CloudFormation?**

Python pode incluir um servidor de aplicação, banco de dados, balanceador de carga, armazenamento de arquivos e componentes de rede. Usando o AWS CloudFormation, estes recursos são definidos em um modelo que descreve suas configurações e relacionamentos, permitindo a implantação e gerenciamento automatizado da infraestrutura, proporcionando escalabilidade e consistência.

- 4. Quais são os principais passos para criar um modelo AWS CloudFormation, como o `infraestrutura.yml`, para definir a infraestrutura necessária no AWS?**

a) Estabelecer os recursos essenciais é o ponto de partida. Inicialmente, identificar os elementos necessários da AWS para a construção da sua infraestrutura. Esses recursos podem incluir instâncias EC2, volumes EBS, bancos de dados RDS, balanceadores.

b) Criar um modelo CloudFormation. O modelo CloudFormation é um arquivo que descreve os recursos da AWS que você deseja criar.

c) Provisionar a infraestrutura. Depois de criar o modelo CloudFormation, você pode usá-lo para provisionar a infraestrutura na AWS.

d) Teste. Realizar testes

e) Implantar em produção.

5. Como o AWS CodePipeline é usado neste estudo de caso para automatizar a entrega contínua de aplicativos Python e infraestrutura como código?

O AWS CodePipeline é utilizado para automatizar a entrega contínua de aplicativos Python e infraestrutura como código na integração do código, na criação da pipeline, no provisionamento da infraestrutura e na implantação do aplicativo Python.

6. Quais são as etapas essenciais configuradas no AWS CodePipeline para orquestrar a implantação da infraestrutura e aplicativos?

As etapas essenciais configuradas no AWS CodePipeline para orquestrar a implantação são:

a) Ter um código fonte versionado no AWS CodeCommit ou algum outro repositório.

b) Criar o build/compilador e verificar se ele passa.

c) Implantação.

7. Qual é o propósito do AWS CodeBuild neste estudo de caso, e como ele é usado para compilar e testar o aplicativo Python e as definições de infraestrutura?

O AWS CodeBuild é usado neste estudo de caso para compilar e testar o aplicativo Python e as definições de infraestrutura. Ele busca o código-fonte do aplicativo e dos modelos de infraestrutura, compila o código, executa testes e gera artefatos de implantação

8. Descreva as etapas dentro do arquivo buildspec.yml e como ele é usado para definir as ações a serem executadas pelo AWS CodeBuild

arquivo buildspec.yml é usado para definir as etapas a serem executadas pelo AWS CodeBuild, conforme abaixo:

version: 0.2: Define a versão da especificação de compilação.

phases: Especifica as fases do processo de compilação.

build: Esta é a fase de compilação.

artifacts: Define quais arquivos ou diretórios devem ser salvos como artefatos após a conclusão bem-sucedida da fase de compilação.

9. Como a execução de testes automatizados é integrada à esteira de DevOps no estudo de caso? Quais são os benefícios de realizar testes automatizados em infraestrutura e aplicativos?

No estudo de caso, os testes automatizados são incorporados à esteira de DevOps, sendo executados após a compilação e implantação permitindo identificar problemas rapidamente, fornecer feedback e garantir que alterações não quebrem o aplicativo ou a infraestrutura. Os benefícios incluem detecção de erros mais rápida, consistência na execução de testes, redução de erros humanos e aceleração da entrega de software.

10. Discuta as vantagens de usar ferramentas de IaC e enfatizar a automação na esteira de DevOps ao implantar infraestrutura e aplicativos na AWS. Como essa abordagem contribui para a eficiência, confiabilidade e escalabilidade do processo de implantação?

O uso de Infraestrutura como Código (IaC) e automação em DevOps ao implantar na AWS oferece vantagens cruciais. A automação economiza tempo e evita erros, enquanto a IaC permite configurações consistentes e escaláveis. Isso torna o processo mais eficiente, seguro e confiável, permitindo entregas rápidas e melhor resiliência das aplicações. A colaboração é aprimorada, e a rastreabilidade garante que todas as mudanças sejam bem documentadas. Em suma, é uma abordagem essencial para operações de TI modernas na AWS.