

TEMPLATE PARA ENTREGA DO PROJETO DA DISCIPLINA

DevOPs na Prática

Fase 1: Configuração e Automação Inicial

NOME DO ESTUDANTE:

Alex Brustolin Teixeira

SEÇÃO 1

Documentação de Planejamento

a) DESCRIÇÃO DO PROJETO, SEUS OBJETIVOS E REQUISITOS:

O Demand Tracker é uma aplicação web desenvolvida em Python utilizando o framework Flask com objetivo de gerenciar demandas e tarefas de forma organizada e escalável. O projeto foi adaptado para contemplar as práticas de DevOps, como integração contínua, automação de testes e provisionamento automatizado de infraestrutura em nuvem tentando abranger uma diferentes gama de aspectos de infraestrutura como armazenamento de arquivos e banco de dados relacional.

Objetivos principais:

- Garantir qualidade contínua do código por meio de testes e análise estática.
- Automatizar a integração contínua com GitHub Actions.
- Provisionar infraestrutura escalável na AWS com CloudFormation.

Requisitos:

- Python 3.9.
- Framework Flask.
- Banco de dados relacional.
- Armazenamento de Arquivos
- Infraestrutura em nuvem.
- Testes automatizados.
- Segurança básica.

b) PLANO DE INTEGRAÇÃO CONTÍNUA:

O pipeline de CI foi criado para garantir que cada alteração submetida ao repositório passe por etapas de validação antes de ser integrada a branch main:

1. **Checkout do código:** garante que o pipeline execute sobre o código atualizado.
2. **Configuração do ambiente Python (v3.9):** compatível com requisitos do projeto.
3. **Instalação de dependências:** via requirements.txt, incluindo dependências de testes.
4. **Análise estática:** flake8 garante boas práticas e detecção de possíveis erros.
5. **Testes automatizados:** executados com pytest.

c) ESPECIFICAÇÃO DETALHADA DA INFRAESTRUTURA NECESSÁRIA:

A infraestrutura é provisionada via AWS CloudFormation, permitindo reprodutibilidade e escalabilidade. Os principais recursos definidos:

Rede (VPC, Subnets, Gateway e Roteamento):

- VPC dedicada (10.0.0.0/16).
- Subnet pública (para o EC2 de acesso externo).
- Duas subnets privadas (para o rds e alta disp).
- Internet Gateway associado à VPC.

Servidores e Banco de Dados:

- EC2 baseado em Amazon Linux 2.
- RDS para persistência de dados, com senha definida via parâmetros seguros.

Armazenamento:

- S3 para armazenamento de artefatos.

Segurança:

- Uso de Security Groups controlando acesso via SSH
- Restrições de senha para o banco
- Exibição de senhas nos logs desativada.
- **Obs:** A idéia era utilizar também o Secrets Manager e os IAM Roles da própria AWS, mas o com acesso limitado da conta academy não foi possível.

SEÇÃO 2

Pipeline de Integração Contínua (CI)

a) CONFIGURAÇÃO DE UM REPOSITÓRIO DE CÓDIGO: Github

Insira o link de acesso ao repositório (GitHub).

<https://github.com/AlexBTeixeira/demand-tracker-devops>

O Banco está publico somente pela necessidade de entrega desse trabalho, uma vez que não foi indicado nenhum email para ser adicionado como colaborador.

Explicação das escolhas no arquivo ci new version.yml

- Disparo em push e pull_request na branch main: garante que alterações submetidas à branch principal passem por validação obrigatória.
- Runner ubuntu-latest: ambiente leve e compatível com Python.
- Linting (com flake8): mantém consistência do código e evita más práticas.

b) IMPLEMENTAÇÃO DE UM PIPELINE DE CI USANDO UM GITHUB ACTIONS:

Insira o link de acesso ao repositório (GitHub).

<https://github.com/AlexBTeixeira/demand-tracker-devops/blob/main/.github/workflows/ci.yml>

Última Run:

Triggered via push yesterday

Status

Total duration

Artifacts

AlexBTeixeira pushed -o- `ede93a3` `main`

Success

30s

—

ci.yml
on: push

Teste e Lint 27s

SEÇÃO 3

Scripts de Infraestrutura como Código

a) SCRIPTS PARA PROVISIONAMENTO DE INFRAESTRUTURA A PARTIR DA FERRAMENTA: AWS CloudFormation



Insira o link de acesso ao repositório (GitHub).

<https://github.com/AlexBTeixeira/demand-tracker-devops/blob/main/cloudformation/template.yml>

Explicação das escolhas

- **Parâmetros configuráveis:** GitHubRepoURL, GitHubPAT, DBPassword e SSHKeyName foram definidos como parâmetros, permitindo reutilização sem alterar o template.
- **Uso do NoEcho para senhas e tokens:** evita exposição de credenciais no console da AWS.
- **Rede organizada:** VPC + subnets públicas e privadas, garantindo isolamento do banco de dados e exposição apenas do necessário.
- **Alta disponibilidade:** duas subnets privadas em zonas de disponibilidade diferentes para redundância.
- **AMI gerenciada via SSM Parameter Store:** sempre atualizada com a última versão do Amazon Linux 2.
- **Security Groups restritivos:** permite limitar o acesso SSH apenas ao IP informado.
- **RDS com senha forte e restrição alfanumérica:** aumenta a segurança.

Ultima Execução da Stack

Stack name	Status	Created time	Description
 demand-tracker-stack	 CREATE_COMPLETE	2025-08-22 13:18:12 UTC-0300	Provisiona a infraestrutura para a aplica??o Demand Tracker (VPC, EC2, RDS, S3) utilizando AWS CloudFormation e levando em conta as limita??es do Perfil de estudante.

DEMAND TRACKER

Demandas

Registrar Horas

Relatórios

Olá, Usuario

Painel de Demandas

Priorização

Todas

+ Nova Demanda

Nenhuma demanda pendente para priorizar.

GRADUAÇÃO
PUCRS online