

# Guia de Deploy para Aplicação FastAPI na AWS

## Descrição

Este documento detalha os passos necessários para fazer o deploy de uma aplicação FastAPI utilizando AWS. Inclui a configuração do servidor, banco de dados, proxy reverso, segurança e monitoramento.

## Passo a Passo para Deploy usando AWS

### 1. Preparação do Ambiente

#### 1.1 Criação de uma Instância EC2

1. Acesse o [AWS Management Console](#).
2. Vá para o serviço EC2 e clique em "Launch Instance".
3. Escolha uma Amazon Machine Image (AMI). Para este exemplo, use a **Ubuntu Server 20.04 LTS**.
4. Selecione o tipo de instância (por exemplo, **t2.micro** se estiver no AWS Free Tier).
5. Configure as opções de instância, adicione armazenamento e tags conforme necessário.
6. Configure um grupo de segurança para permitir acesso SSH (porta 22), HTTP (porta 80) e HTTPS (porta 443).
7. Clique em "Review and Launch" e depois "Launch".
8. Baixe a chave PEM para acessar a instância via SSH.

#### 1.2 Conexão à Instância EC2

1. Abra um terminal e conecte-se à instância via SSH:

```
ssh -i "caminho/para/sua-chave.pem" ubuntu@seu-endereco-ip
```

## 2. Configuração do Ambiente no Servidor

### 2.1 Atualização do Sistema

1. Atualize os pacotes do sistema:

```
bash
Copiar código
sudo apt update
sudo apt upgrade -y
```

### 2.2 Instalação de Dependências

1. Instale Python e as dependências necessárias:

```
sudo apt install python3 python3-venv python3-pip nginx -y
```

## 2.3 Configuração do Ambiente Virtual

1. Crie e ative um ambiente virtual:

```
python3 -m venv venv  
source venv/bin/activate
```

## 2.4 Instalação das Dependências do Projeto

1. Clone o repositório do projeto:

```
git clone https://github.com/seu-usuario/seu-repositorio.git  
cd seu-repositorio
```

2. Instale as dependências do projeto:

```
pip install -r requirements.txt
```

## 3. Configuração do Banco de Dados

1. No seu arquivo `database.py`, configure o URL do banco de dados. Para este exemplo, usamos SQLite:

```
SQLALCHEMY_DATABASE_URL = "sqlite:///./test.db"
```

2. Inicialize o banco de dados:

```
python  
python  
Copiar código  
from database import Base, engine  
Base.metadata.create_all(bind=engine)
```

## 4. Configuração do Gerenciador de Processos

### 4.1 Instalação e Configuração do Gunicorn

1. Instale o Gunicorn:

```
pip install gunicorn
```

2. Crie um arquivo de serviço systemd para Gunicorn:

```
sudo nano /etc/systemd/system/gunicorn.service
```

Adicione o seguinte conteúdo:

```
[Unit]  
Description=gunicorn daemon
```

```
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/caminho/para/seu-projeto
ExecStart=/caminho/para/venv/bin/gunicorn --workers 3 --bind
unix:/caminho/para/seu-projeto/gunicorn.sock main:app

[Install]
WantedBy=multi-user.target
```

### 3. Inicie e habilite o serviço Gunicorn:

```
bash
Copiar código
sudo systemctl start gunicorn
sudo systemctl enable gunicorn
```

## 5. Configuração do Proxy Reverso

### 5.1 Configuração do Nginx

#### 1. Crie um novo arquivo de configuração para o seu site:

```
sudo nano /etc/nginx/sites-available/seu-site
```

Adicione o seguinte conteúdo:

```
server {
    listen 80;
    server_name seu-dominio.com;

    location / {
        proxy_pass http://unix:/caminho/para/seu-
projeto/gunicorn.sock;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

#### 2. Ative a configuração do site e remova a configuração padrão:

```
bash
Copiar código
sudo ln -s /etc/nginx/sites-available/seu-site /etc/nginx/sites-
enabled
sudo rm /etc/nginx/sites-enabled/default
```

#### 3. Teste a configuração do Nginx e reinicie o serviço:

```
bash
Copiar código
sudo nginx -t
```

```
sudo systemctl restart nginx
```

## 6. Configuração de Segurança

### 6.1 Configuração de HTTPS com Let's Encrypt

1. Instale Certbot:

```
sudo apt install certbot python3-certbot-nginx -y
```

2. Obtenha e instale o certificado SSL:

```
sudo certbot --nginx -d seu-dominio.com
```

3. Siga as instruções para completar a configuração SSL.

## 7. Monitoramento e Manutenção

### 7.1 Configuração de Logs e Monitoramento

1. Certifique-se de que os logs estão configurados no Nginx:

```
access_log /var/log/nginx/access.log;  
error_log /var/log/nginx/error.log;
```

2. Considere configurar ferramentas de monitoramento como Prometheus e Grafana para monitorar sua aplicação.

Segue uma foto de como ficaria o fluxograma:

