

Manual para Construção de Script Bash para Implantação de Aplicações FastAPI

Este manual explica passo a passo como construir um script bash independente, permitindo que qualquer desenvolvedor possa automatizar o processo de implantação de uma aplicação FastAPI, realizando testes automatizados, gerenciamento de dependências e execução da aplicação usando Uvicorn, utilizando variáveis definidas em um arquivo `.env`.

1. Estrutura Inicial do Script

Crie um novo arquivo bash (`deploy.sh`) com a seguinte linha inicial para garantir segurança na execução do script:

```
...  
  
#!/usr/bin/env bash  
set -euo pipefail  
...
```

2. Carregar Variáveis de Ambiente

Utilize um arquivo `.env` na raiz da aplicação para definir variáveis essenciais (instale `dotenv-cli` caso não possua):

```
`pip install python-dotenv`
```

No seu script, adicione a linha abaixo para carregar as variáveis:

```
...  
set -a  
source .env  
set +a  
...
```

Exemplo de arquivo `.env`:

```
...  
  
APP_DIR="/caminho/da/aplicacao"  
VENV_PATH="$APP_DIR/.venv"  
PORTA=8080  
HOST="localhost"  
API_KEY="sua_chave_api"  
...
```

3. Entrar no Diretório da Aplicação

Garanta que o script execute comandos no diretório correto:

```
`cd "$APP_DIR" || { echo "Falha ao entrar em $APP_DIR"; exit 1; }
```

4. Configuração do Ambiente Virtual Python

Verifique e configure automaticamente um ambiente virtual:

```
...  
if [ ! -d "$VENV_PATH" ]; then  
    python3 -m venv "$VENV_PATH"  
fi  
  
source "$VENV_PATH/bin/activate"  
...
```

5. Instalação de Dependências

Atualize e instale dependências do projeto via **pip**:

```
...  
pip install -e ".[standard,dev]" --upgrade  
pip install pytest  
...
```

6. Execução de Testes Automatizados

Execute testes presentes na pasta **tests** com **pytest**:

```
`pytest tests --maxfail=1 --disable-warnings -q -o python_files="test*.py"`
```

Avalie o resultado para determinar continuidade:

- Sucesso: continue.
- Nenhum teste encontrado: exiba aviso, mas continue.
- Falha nos testes: encerre o script com erro.

7. Finalização de Processos Existentes

Identifique e encerre processos existentes que usam a porta definida:

```
...  
PID_API=$(lsof -ti tcp:"$PORTA" || true)  
if [ -n "$PID_API" ]; then  
    kill "$PID_API"  
fi
```

```
    sleep 2
fi
...
```

8. Execução da Aplicação FastAPI com Uvicorn

Inicie a aplicação em segundo plano utilizando **nohup** e registre logs:

```
`nohup bash -c "python -m uvicorn main:app --host $HOST --port $PORTA --workers 1" >
fastapi.log 2>&1 &`
```

Como Executar o Script

- Salve o script em **deploy.sh**.
- Dê permissão de execução:

```
`chmod +x deploy.sh`
```

- Execute no terminal:

```
./deploy.sh
```

Automatização Diária com Crontab

Para garantir que a aplicação seja reiniciada diariamente à meia-noite, adicione a seguinte linha ao crontab:

```
0 0 * * * /caminho/da/aplicacao/deploy.sh
```

Boas Práticas

- Revise e atualize variáveis no arquivo **.env** antes de executar.
- Monitore a execução pelo arquivo de log:

```
`tail -f fastapi.log`
```

Seguindo este manual, sua equipe será capaz de criar scripts semelhantes, facilitando implantações rápidas, seguras e padronizadas para projetos FastAPI.

Exemplo de código utilizado no copilotpdf:

...

```
#!/usr/bin/env bash
set -euo pipefail

# --- Configurações ---
APP_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
VENV_PATH="$APP_DIR/.venv"
PORTA=8080
HOST="195.35.40.52"
#HOST="localhost"
API_KEY="ff371b9acaf05edaef5082d5dc35f54fe0819bce7b06afe91174542253c5a8c2"

# --- Entrar no diretório da aplicação ---
cd "$APP_DIR" || { echo "Falha ao entrar em $APP_DIR"; exit 1; }

# --- Configurar PYTHONPATH para pytest e imports ---
export PYTHONPATH="$APP_DIR"

# --- Verificar/Configurar o virtualenv ---
if [ ! -d "$VENV_PATH" ]; then
    echo "Virtualenv não encontrado em $VENV_PATH. Criando..."
    python3 -m venv "$VENV_PATH"
fi
# shellcheck source=/dev/null
source "$VENV_PATH/bin/activate"

# --- Variáveis de ambiente ---
export API_KEY

# --- Atualizar dependências ---
echo "Atualizando dependências em modo editável com extras de desenvolvimento..."
pip install -e ".[standard,dev]" --upgrade
# Garantir pytest instalados
pip install pytest

# --- Rodar suíte de testes na pasta tests ---
echo "Rodando testes na pasta tests..."
pytest tests --maxfail=1 --disable-warnings -q -o
python_files="test*.py" || TEST_EXIT=$?
TEST_EXIT=${TEST_EXIT:-0}

if [ "$TEST_EXIT" -eq 0 ]; then
    echo "✅ Testes passaram com sucesso."
elif [ "$TEST_EXIT" -eq 5 ]; then
    echo "⚠️ Nenhum teste encontrado na pasta tests (exit code 5). Prosseguindo."
fi
```

```
else
    echo "⊘ Testes falharam (exit code $TEST_EXIT). Abortando
restart."
    exit 1
fi

# --- Parar processo que esteja usando a porta da API ---
PID_API=$(lsof -ti tcp:$PORTA || true)
if [ -n "$PID_API" ]; then
    echo "Encerrando API na porta $PORTA (PID $PID_API)..."
    kill "$PID_API"
    sleep 2
else
    echo "Nenhum processo API escutando na porta $PORTA."
fi

# --- Iniciar a aplicação FastAPI ---
echo "Iniciando API via Uvicorn..."
nohup bash -c "python -m uvicorn main:app --host $HOST --port
$PORTA --workers 1" > fastapi.log 2>&1 &

echo "API iniciada em http://$HOST:$PORTA (logs em fastapi.log)"
```