Flask アプリケーションの開発から本番環境へのデプロイ手順

Flask アプリケーションのデプロイ手順を以下の流れで説明します。これにより、開発したアプリケーションを本番環境にスムーズに配置することが可能です。

1. Flask アプリケーションの開発

Flask アプリケーション作成

- Flask を使用して、必要な機能を持つアプリケーションを開発します。
- 通常、app.py や run.py というファイルに記述します。

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, World!"

if __name__ == '__main__':
    app.run(debug=True)
```

依存パッケージのインストール

- 必要なパッケージを requirements.txt にまとめます。

```
Flask==2.1.0
gunicorn==20.1.0
```

- 仮想環境にパッケージをインストールします。

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

2. VPS の準備と環境構築

VPS の契約とセットアップ

- VPS を契約し、OS（Ubuntu）をセットアップします。
- 必要なパッケージをインストールします。

sudo apt update
sudo apt install python3 python3-pip python3-venv nginx

アプリケーションの配置

- VPS 上にアプリケーションを配置します。例：/home/your_user/apps/
- scp や rsync で必要なファイルをアップロードします。

scp -r myapp user@your_vps:/home/your_user/apps/

3. Gunicorn の設定と起動

Gunicorn のインストール

- Gunicorn は、Flask アプリケーションを WSGI サーバーとして動作させます。

pip install gunicorn

Gunicorn の起動

- アプリケーションを Gunicorn を使って起動します。

gunicorn -w 4 -b 127.0.0.1:8000 run:app

- -w 4 は 4 ワーカープロセスを起動するオプションです。

4. Nginx の設定とリバースプロキシ

Nginx の設定

- Nginx は HTTP リクエストを受け取り、Flask アプリケーションへリバースプロキシとしてリクエストを転送します。

```
server {
  listen 80;
  server_name dipalette.com www.dipalette.com;

  location / {
    proxy_pass http://127.0.0.1:8000;
    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;
  }
}
```

Nginx の有効化

- 設定ファイルを配置し、シンボリックリンクを作成します。

sudo ln -s /etc/nginx/sites-available/dipalette /etc/nginx/sites-enabled/

Nginx の再起動

- 設定を反映するため、Nginx を再起動します。

sudo systemctl restart nginx

5. SSL 証明書の取得 (Let's Encrypt)

SSL 証明書の取得

- certbot を使って SSL 証明書を取得します。

sudo certbot --nginx -d dipalette.com -d www.dipalette.com

自動更新の確認

- 証明書の自動更新を確認します。

sudo systemctl status certbot.timer

6. 最終確認とデプロイの完成

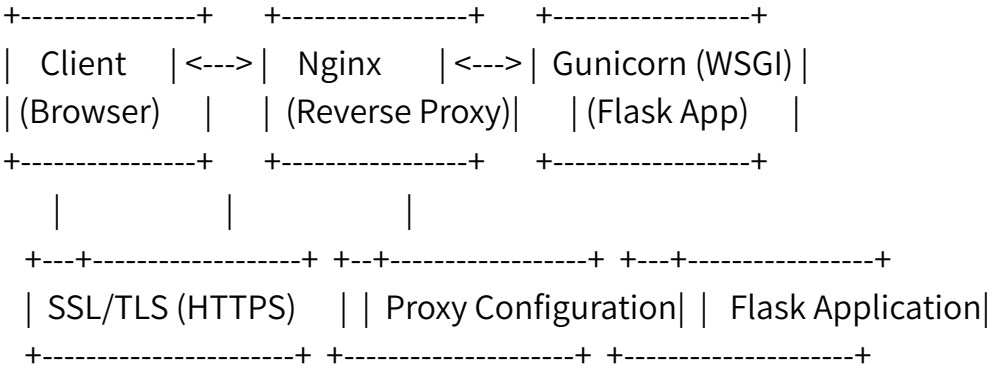
SSL 証明書の確認

- ブラウザで https://dipalette.com にアクセスし、SSL 証明書を確認します。

アプリケーションの動作確認

- アプリケーションが正常に動作していることを確認します。

構成図



デプロイの流れまとめ

1. Flask アプリケーションの開発

2. VPS のセットアップと依存パッケージのインストール
3. Gunicorn を使って Flask アプリケーションをサーバーで実行
4. Nginx をリバースプロキシとして設定
5. Let's Encrypt を使って SSL 証明書を取得
6. アプリケーションを動作させ、最終確認

この手順を新しいプロジェクトに流用することで、どのような Flask アプリケーションでも同様にデプロイが可能です。