

Guia de Hospedagem de Site Django com MySQL em Ubuntu

Este guia cobre o processo completo de hospedagem de um site Django com MySQL como banco de dados usando seu desktop Ubuntu como servidor. Também cobre a integração com Gunicorn e Nginx para produzir um ambiente de produção estável. Além disso, ele inclui a opção de usar um ambiente conda do Anaconda para gestão de dependências.

1. Pré-requisitos

- Ubuntu Desktop atualizado
- Conexão de internet confiável com IP público
- Firewall configurado para permitir as portas 80 (HTTP) e 443 (HTTPS)
- Anaconda (opcional) instalado

2. Instalar pacotes necessários

Execute os seguintes comandos para instalar o Python, MySQL, Django e outros pacotes essenciais:

```
sudo apt update
```

```
sudo apt install python3-pip python3-dev libmysqlclient-dev mysql-server nginx curl
```

3. Configurar o Banco de Dados MySQL

1. Acesse o MySQL como root:

```
sudo mysql
```

2. Crie um banco de dados e um usuário para o Django:

```
CREATE DATABASE meu_site_db CHARACTER SET UTF8;
```

```
CREATE USER 'meu_usuario_db'@'localhost' IDENTIFIED BY 'senha_segura';
```

```
GRANT ALL PRIVILEGES ON meu_site_db.* TO 'meu_usuario_db'@'localhost';  
  
FLUSH PRIVILEGES;  
  
EXIT;
```

4. Configurar o Projeto Django

1. Instale o Django e o conector MySQL:

```
conda install django mysqlclient
```

2. Crie o projeto Django:

```
django-admin startproject meu_site  
  
cd meu_site
```

3. Configure o banco de dados no arquivo settings.py:

```
DATABASES = {  
  
    'default': {  
  
        'ENGINE': 'django.db.backends.mysql',  
  
        'NAME': 'meu_site_db',  
  
        'USER': 'meu_usuario_db',  
  
        'PASSWORD': 'senha_segura',  
  
        'HOST': 'localhost',  
  
        'PORT': '3306',  
  
    }  
  
}
```

4. Migre o banco de dados:

```
python manage.py migrate
```

```
python manage.py createsuperuser
```

5. Configurar Gunicorn e Nginx

1. Instale o Gunicorn e inicie o serviço:

```
conda install gunicorn
```

2. Crie um arquivo de serviço systemd para o Gunicorn:

```
[Unit]
```

```
Description=gunicorn daemon
```

```
After=network.target
```

```
[Service]
```

```
User=seu_usuario_ubuntu
```

```
Group=www-data
```

```
WorkingDirectory=/caminho/para/seu/projeto/meu_site
```

```
ExecStart=/caminho_para_conda_env/bin/gunicorn --access-logfile - --workers 3 --bind
```

```
unix:/caminho/para/seu/projeto/meu_site.sock meu_site.wsgi:application
```

```
[Install]
```

```
WantedBy=multi-user.target
```

3. Configure o Nginx como proxy reverso:

```
server {  
    listen 80;  
  
    server_name seu_dominio_ou_IP_publico;  
  
    location / {  
        proxy_pass http://unix:/caminho/para/seu/projeto/meu_site.sock;  
    }  
}
```

6. Testar se o site está ativo

1. Verifique se o Django está em execução usando:

```
python manage.py runserver
```

2. Verifique o status do Gunicorn e Nginx:

```
sudo systemctl status gunicorn
```

```
sudo systemctl status nginx
```

3. Acesse o site no navegador pelo endereço IP ou domínio configurado.