

Instalação e Configuração de Servidores

Prof. Diego Cirilo

Aula 12: Deploy de um sistema Django

Deploy

- *Implantar*
- Enviar seu sistema para um servidor de *produção*
- Desenvolvimento/Produção

Preparação

- Clone do seu *virtualenv*
- Na máquina de desenvolvimento:
 - `pip freeze > requirements.txt`
- No servidor:
 - `pip install -r requirements.txt`
- **ATENÇÃO:** nunca reutilize seu *venv*, crie um em cada máquina que usar.

Preparação

- Verifique que o *nginx* e o *MySQL* estão funcionando corretamente (aulas anteriores)
- Instale as dependências:

```
$ sudo apt-get install python3-pip python3-venv libmariadb-dev pkg-config
```

Preparação

- Copie a pasta do seu projeto Django para sua *home* (use o Filezilla)
- Re-crie o *venv* dentro da pasta do seu projeto e ative-o (perceba que o comando do *venv* é diferente do *Windows*):

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

- Instale as dependências do seu projeto, se houver:

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

- Instale também as novas dependências para o servidor

```
$ pip install mysqlclient gunicorn django
```

Configure o banco de dados

- Crie um usuário e uma tabela no banco para seu projeto (nunca rode como *root*)
- **Substitua o <exemplo> com suas informações**

```
$ sudo mysql -u root -p
$ CREATE DATABASE <banco-do-app> CHARACTER SET 'utf8';
$ CREATE USER <user-do-app>;
$ GRANT ALL ON <banco-do-app>.* TO '<user-do-app>'@'localhost' IDENTIFIED BY '<senha-do-user-do-app>';
$ quit
```

Modifique as configurações

- Modifique o arquivo `settings.py` para configurar os *hosts* e o banco de dados.
- Adicione `import os` no início do arquivo.

```
DEBUG = False

ALLOWED_HOSTS = ['127.0.0.1', 'localhost']

...

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'OPTIONS': {
            'sql_mode': 'traditional',
        },
        'NAME': '<banco-do-app>',
        'USER': '<user-do-app>',
        'PASSWORD': '<senha-do-user-do-app>',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

...

STATIC_URL = '/static/'
STATIC_ROOT=os.path.join(BASE_DIR, 'static/')

MEDIA_URL='/media/'
MEDIA_ROOT=os.path.join(BASE_DIR, 'media/')
```

Migrations

- Faça as *migrations*

```
python manage.py makemigrations  
python manage.py migrate
```

- Faça as colete os arquivos estáticos para a pasta *static*

```
python manage.py collectstatic
```

- Crie também o *superuser* do seu sistema

```
python manage.py createsuperuser
```

- Execute o servidor de teste para saber se o sistema ainda sobe

```
python manage.py runserver
```

Configure o *gunicorn*

- *Gunicorn* - Unicórnio Verde: servidor HTTP para Python.
- Processa e encaminha as requisições entre o servidor web e a aplicação Python
- Deve rodar como um *daemon* no sistema

Gunicorn Daemon

- Crie um arquivo <app>_gunicorn.service

```
sudo nano /etc/systemd/system/<app>_gunicorn.service
```

- Com o seguinte conteúdo:

```
[Unit]
Description=gunicorn service
After=network.target

[Service]
User=<user>
Group=ww-data
WorkingDirectory=/home/<user>/<projeto-django>/
ExecStart=/home/<user>/<projeto-django>/venv/bin/gunicorn --access-logfile - --workers 3 --bind unix:/home/<user>/<projeto-django>/<projeto-django>.sock <projeto-django>.wsgi:application

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

Gunicorn Daemon

- Habilite o *daemon*

```
sudo systemctl enable <app>_gunicorn.service  
sudo systemctl start <app>_gunicorn.service  
sudo systemctl status <app>_gunicorn.service
```

- Sempre que fizer alterações do seu sistema, reinicie os *daemons*

```
sudo systemctl daemon-reload  
sudo systemctl restart <app>_gunicorn.service
```

Configurando o Nginx

- Crie um novo arquivo de configuração do nginx

```
sudo nano /etc/nginx/sites-available/<projeto-django>
```

- Adicione os seguintes conteúdos, alterando o que for necessário:

```
server {
    listen 80;
    server_name 127.0.0.1;
    location = /favicon.ico {access_log off;log_not_found off;}

    location /static/ {
        alias /home/<user>/<projeto-django>/static/;
    }

    location /media/ {
        alias /home/<user>/<projeto-django>/media/;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/home/<user>/<projeto-django>/<projeto-django>.sock;
    }
}
```

Configurando o Nginx

- Habilite o site (desabilite o default se for necessário)

```
sudo ln -s /etc/nginx/sites-available/<projeto-django> /etc/nginx/sites-enabled
```

- Verifique se está tudo ok

```
sudo nginx -t
```

Se não houver nenhum erro, reinicie o nginx

```
sudo systemctl restart nginx
```

- Acesse a aplicação em <http://localhost:8888/>

Solucionando erros

- Leia o *status* do *nginx* com o `systemctl status`
- Leia o *log* de erros do *nginx* para mais informações usando:

```
sudo tail /var/log/nginx/error.log
```

- Caso haja erros de acesso negado ao acessar o site, dê acesso completo ao seu diretório (o `-R` significa recursivo), reinicie o serviço e o *nginx*:

```
chmod -R 777 /home/<usuario>
```

Dúvidas?

