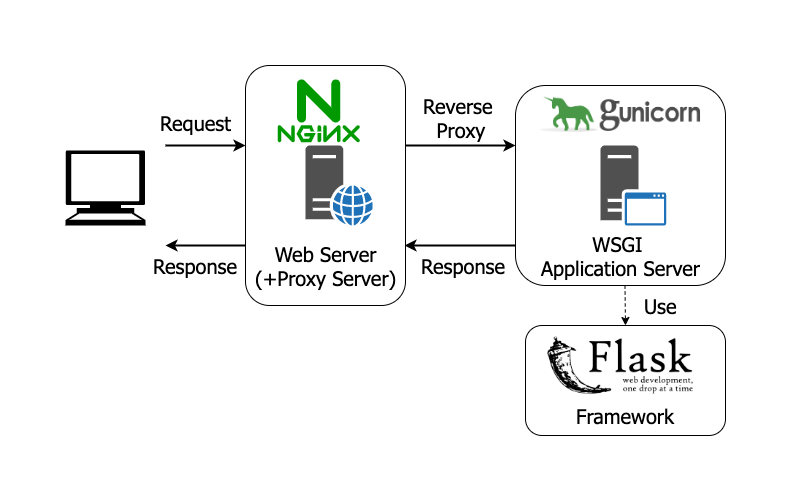
**Arquitetura**



Premissas:

* Será utilizado o OpenSSL para geração do certificado
* O documento tem como objetivo explicar a utilização do certificado para o servidor HTTP GUNICORN rodando no sistema operacional Ubuntu

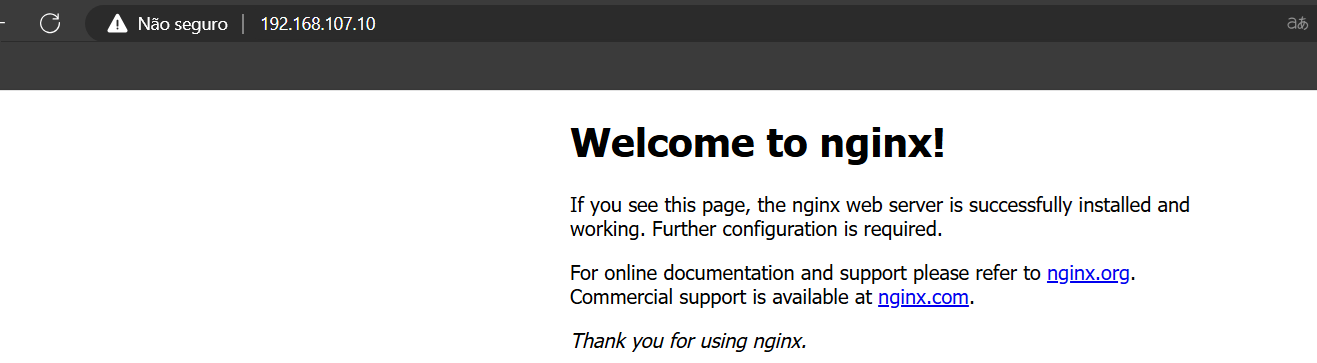
1. Atualize repositório linux:

**#sudo apt-get update**

2. Instale o nginx:

**# sudo apt-get install nginx**

3. Digite o seu ip corrente no browser e certifique-se que a tela de boas vindas do ngix aparece:



4. Remover os arquivos padrões:

**#sudo rm /etc/nginx/sites-enabled/default**

**#sudo rm /etc/nginx/sites-available/default**

5. Para fazer proxy do tráfego para a porta 8000, é necessário configurar o Nginx. Para isso, crie um arquivo de configuração dentro da pasta sites-available, eu com o nome de "myapp.com" e então adicione um symlink para sites-enabled no Nginx:

**#sudo touch /etc/nginx/sites-available/myapp.com**

**#sudo chown -R $USER:$USER /etc/nginx/sites-available/myapp.com**

6. Altere o diretório para sites-available, se você ainda não estiver lá, abra o arquivo de configuração (myapp.com):

**#cd /etc/nginx/sites-available/**

7. Edite o arquivo myapp.com e cole as linhas abaixo (substituindo valores se necessário). No exemplo abaixo, a porta 80 direciona para o endereço do flask:

**server {**

**listen 80; #(porta que irá responder)**

**location / {**

**proxy\_pass http://127.0.0.1:8000/; #(endereço do flask)**

**}**

**}**

8. Adicionar um symlink ao diretório sites-enabled:

**#sudo ln -f -s /etc/nginx/sites-available/myapp.com /etc/nginx/sites-enabled/myapp.com**

9. Reinicie o Nginx:

**#sudo service nginx restart**

10.Gere um par de chaves e um certificado usando OPENSSL

openssl req -x509 -nodes -days 3650 -newkey ec:<(openssl ecparam -name prime256v1) -keyout private\_key.pem -out certificate.pem

openssl x509 -outform der -in certificate.pem -out certificate.crt