

Sumário

Setup inicial	2
Instalação	3
Alterando a página inicial	5
Próximos passos	7

Setup inicial

Antes de mergulharmos no tutorial de configuração do NGINX, precisamos preparar o ambiente na AWS. Este guia passo-a-passo irá ajudá-lo a configurar uma instância EC2 que servirá como base para a instalação do NGINX. Esta configuração é um pré-requisito essencial.

Pré-requisitos

- Conta na Amazon Web Services (AWS);
- Configuração da Instância EC2;

Lançamento da Instância EC2:

- Acesse o console da AWS;
- Navegue até o serviço EC2 e clique em "Launch Instance" para criar uma instância;

Configurações de Rede:

- Durante o processo de configuração, certifique-se de que a instância tenha Acesso à Internet;
- Atribua um IP Público à instância. Essa etapa é crucial para garantir que sua instância seja acessível pela internet;

Atribuir IP público automaticamente [Informações](#)

Habilitar ▼

- Na seção de configuração de rede, escolha ou configure uma sub-rede Pública. Se você já tem uma VPC configurada com sub-redes e um Internet Gateway, selecione a sub-rede pública apropriada;

Configurações de Segurança:

- Para permitir o tráfego HTTP, é necessário configurar o grupo de segurança da instância para liberar a Porta 80;
- No grupo de segurança, adicione uma regra para permitir todo o tráfego na porta 80 (HTTP). Isso é essencial para que o NGINX, um servidor web, possa receber requisições HTTP;

Tipo Informações	Protocolo Informações	Intervalo de portas Informações
TCP personalizado ▼	TCP	80
Tipo de origem Informações	Origem Informações	Descrição (opcional) Informações
Qualquer lugar ▼	Add CIDR, prefix list or security	navegacao web

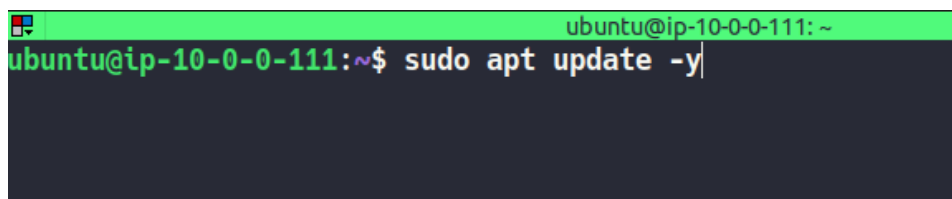
NGINX: configurando seu primeiro site em um WebServer

Revisão e Lançamento:

- Revise todas as configurações. Certifique-se de que a instância tenha acesso à internet, um IP público atribuído e a porta 80 liberada.
- Selecione uma chave de acesso (ou crie uma nova) que será usada para conectar-se à instância via SSH.
- Clique em "Launch Instances" para criar sua instância EC2.

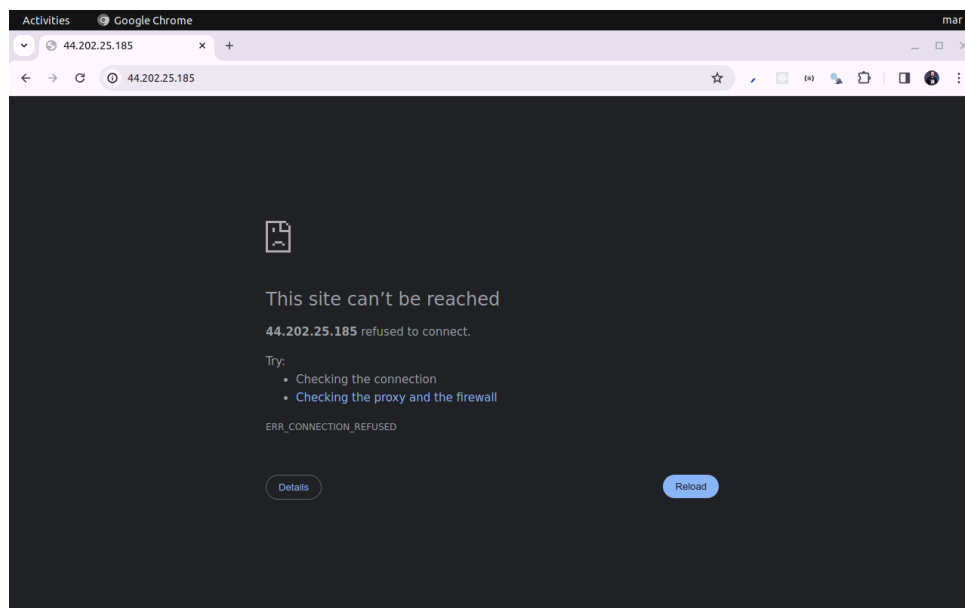
Instalação

Conecte-se à sua instância por SSH e assegure-se de que o sistema operacional está atualizado executando o comando `sudo apt update -y`.



```
ubuntu@ip-10-0-0-111: ~  
ubuntu@ip-10-0-0-111:~$ sudo apt update -y
```

A porta padrão para navegação na web é a 80, mas simplesmente liberar essa porta e digitar o endereço IP no navegador não resultará em sucesso, isto é, você não encontrará nenhum site ou arquivo disponível para ser acessado na web:



Prosseguiremos com a instalação do **NGINX**, que pode desempenhar três papéis cruciais em nosso projeto:

- Atuar como um balanceador de carga
- Funcionar como um proxy reverso
- Servir como um servidor web.

NGINX: configurando seu primeiro site em um WebServer

Para isso, execute o comando **sudo apt install nginx -y**.

```
ubuntu@ip-10-0-0-111: ~  
ubuntu@ip-10-0-0-111:~$ sudo apt install nginx -y
```

Normalmente, após a instalação, o NGINX estará em funcionamento, operando como um serviço padrão do Linux. Esse serviço pode ser administrado por meio do comando **systemctl**. Para conferir se o serviço está ativo, utilize o comando

sudo systemctl status nginx.

```
ubuntu@ip-10-0-0-111:~$ sudo systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)  
   Active: active (running) since Wed 2024-03-20 17:32:09 UTC; 1min 57s ago  
     Docs: man:nginx(8)  
  Process: 2130 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
  Process: 2131 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
 Main PID: 2225 (nginx)  
    Tasks: 2 (limit: 1121)  
   Memory: 4.6M  
      CPU: 26ms  
   CGroup: /system.slice/nginx.service  
           └─2225 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"  
             └─2228 "nginx: worker process" ...  
  
Mar 20 17:32:09 ip-10-0-0-111 systemd[1]: Starting A high performance web server and a reverse proxy server...  
Mar 20 17:32:09 ip-10-0-0-111 systemd[1]: Started A high performance web server and a reverse proxy server.  
ubuntu@ip-10-0-0-111:~$
```

Se o serviço não estiver ativo, inicie-o executando o comando

systemctl start nginx.

Após concluir esse processo, tente acessar o IP pelo navegador para verificar se houve alguma alteração. É provável que você se depare com a seguinte tela:



Voilà! Sua primeira página foi publicada por meio de um servidor web.

Alterando a página inicial

Intitulada "Welcome to NGINX", essa página se localiza em um diretório na sua EC2, geralmente em /var/www. Vamos acessá-la utilizando o comando **cd /var/www/html**.

```
ubuntu@ip-10-0-0-111: /var/www/html$ cd /var/www/html/
ubuntu@ip-10-0-0-111: /var/www/html$ ll
total 12
drwxr-xr-x 2 root root 4096 Mar 20 17:32 ./
drwxr-xr-x 3 root root 4096 Mar 20 17:32 ../
-rw-r--r-- 1 root root 612 Mar 20 17:32 index.nginx-debian.html
ubuntu@ip-10-0-0-111: /var/www/html$
```

A seguir, visualize o conteúdo desse arquivo utilizando o comando

cat index.nginx-debian.html:

```
ubuntu@ip-10-0-0-111: /var/www/html$ cat index.nginx-debian.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-10-0-0-111: /var/www/html$
```

O conteúdo do arquivo confirma que esta é, de fato, a página servida por padrão na porta 80 quando acessamos pelo navegador, digitando o IP da nossa EC2. Para verificar, podemos modificar a mensagem ou o conteúdo dessa página. Eu utilizarei o Vim, mas sinta-se à vontade para usar a ferramenta de edição de texto de sua preferência, como o Nano, o Emacs ou qualquer outro editor de sua escolha.

Execute o seguinte comando:

sudo vim index.nginx-debian.html

NGINX: configurando seu primeiro site em um WebServer

```
!DOCTYPE html
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

"index.nginx-debian.html" 25L, 612B      1,1      All
```

Mude o programa para o modo de edição pressionando a tecla "Insert" ou digitando "i" no teclado.

```
<p><em>Thank you for using
</body>
</html>
~
-- INSERT --
```

Ao efetuar essa operação, a palavra **"INSERT"** aparecerá no canto inferior esquerdo da tela.

Decidi modificar o texto dentro da tag h1 para o clássico **"Hello from NGINX"**, afastando assim qualquer maldição de programação.

Para concluir a edição, pressione a tecla **ESC**. Em seguida, digite **:wq** e aperte **ENTER** para salvar as alterações e sair.

```
</body>
</html>
~
~
~
~
~
~
~
~
:wq|
```

NGINX: configurando seu primeiro site em um WebServer

Ao tentar acessar novamente o ip da nossa EC2, veremos a nossa mensagem que configuramos no step anterior:



Hello from NGINX!

Isso eh tudo pessoal :)
Congrats!!!

Próximos passos

Os mais curiosos provavelmente estão se perguntando: "Como faço para hospedar meu site neste servidor web?" Nas próximas aulas, vamos detalhar e ajudar com esse processo, mas já podemos antecipar algumas dicas:

Normalmente, a maioria dos modelos iniciais de React já chega com aquele comando de build caprichado, algo na linha de um **npm build** bem charmoso.

- "Ah, mas eu sou mais Yarn, e agora?"

Não tem mistério! Dá uma olhada nos arquivos de configuração do seu projeto. Utilizarei como exemplo o package.json

Aqui vai um exemplo, direto do package.json:

```
▸ Debug
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
```

```
brito@lenovo:~/Docum
brito in pagina-exemplo-react
→ npm run build
```

NGINX: configurando seu primeiro site em um WebServer

Este comando cria um diretório que, dependendo da configuração e do projeto, pode ser chamado de "dist" ou "build". Dentro deste diretório, você encontrará seu código React transpilado e otimizado. Afinal, seu navegador não tem a mínima ideia de como lidar com esse monte de TypeScript ou componentes, certo? Então, é só zipar esse diretório, transferi-lo para /var/www/html no servidor, assegurando que o index.html do seu projeto esteja na raiz e os demais arquivos organizados dentro dele.

ENJOY!