

Ao invés de fazer a criação do arquivo index.html na mão, podemos deixar isso com o terraform. Isso pode ser feito através da tag user_data.
O EOF serve para indicarmos o início e o fim do script que estamos fazendo.

```
user_data = <<-EOF
    #!/bin/bash
    cd /home/ubuntu
    echo "<h1>Feito com Terraform</h1>" > index.html
    nohup busybox httpd -f -p 8080 &
EOF
```

Como alteramos o resource, ao usarmos o terraform apply, a nossa instância será destruída e outra será criada.

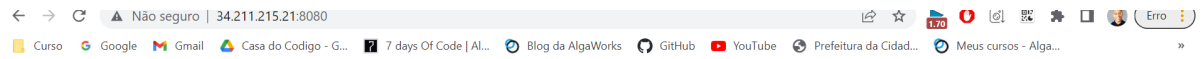
Isso significa que o nosso ipv muda e a ssh de conexão também vai mudar. Obs: O Par de chaves permanece a mesma.

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there's a status bar with a checkmark, the instance name 'Teste AWS', its ID 'i-04d006a5c8d3b0846', a green 'Running' status, a magnifying glass icon, the instance type 't2.micro', a green checkmark indicating '2/2 checks passed', and 'No alarms'. Below this is a tabbed interface with 'Details' selected. The 'Instance summary' section shows the Instance ID, Public IPv4 address (34.211.215.21), Private IPv4 addresses (172.31.4.76), IPv6 address, Instance state, and Public IPv4 DNS. The bottom of the console shows the footer with '© 2022 Amazon Web Services, Inc. or its affiliates'.

Feita a conexão, podemos até ver o arquivo que passamos para o terraform criar

The screenshot shows a terminal window with the following content: A warning message about adding a host to the list of known hosts, followed by a welcome message to Ubuntu 22.04.1 LTS. It then displays system information, including system load, processes, memory usage, and swap usage. It also shows the output of the 'ls' command, listing the 'index.html' file, and the output of the 'cat index.html' command, which shows the HTML content: '<h1>Feito com Terraform</h1>'. The terminal prompt is 'ubuntu@ip-172-31-4-76:~\$'.

Podemos nos conectar ao servidor também.



Feito com Terraform