



Aula - AWS EC2

"O Amazon Elastic Compute Cloud (Amazon EC2) é um serviço Web que disponibiliza capacidade computacional segura e redimensionável na nuvem. O EC2 oferece muitas opções que permitem criar e executar virtualmente qualquer aplicativo."

Isso é como a própria Amazon define o serviço EC2 da AWS. Quando falamos em "instâncias EC2" estamos falando de servidores completos na nuvem, preparados para receber e executar qualquer aplicação de forma segura e escalável, sem demandar muito trabalho para configuração.

Na versão gratuita, podemos executar instâncias EC2 por até 750 horas/mês com algumas limitações de recursos, durante 12 meses.

Vamos ver como podemos subir uma aplicação numa instância EC2.

Etapa 1 - Criar conta na Amazon

Caso ainda não possua uma conta na Amazon, você pode criá-la gratuitamente.

[Criar conta na AWS](#)

Importante!

Durante as etapas de criação será solicitado o cadastramento de um cartão de crédito, siga as recomendações abaixo para evitar problemas futuros e cobranças indevidas:

Não cadastre um cartão de crédito de uso familiar

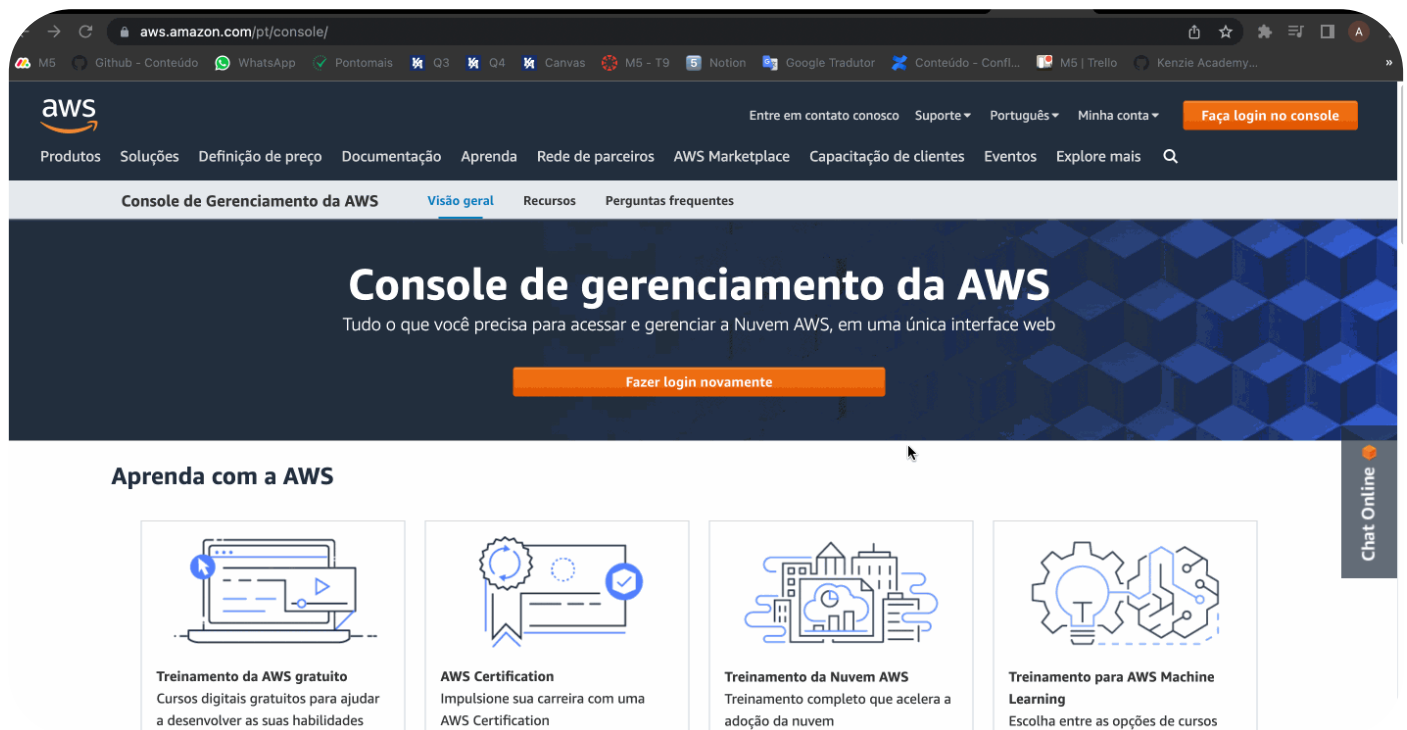
Se possível utilize um cartão pré-pago ou virtual

Etapa 2 - Definir orçamento

Essa é outra etapa importante para que possamos evitar problemas como cobranças futuras e surpresas indesejáveis.

Após criar sua conta na AWS, você pode definir um orçamento, para ser avisado quando chegar ao limite definido.

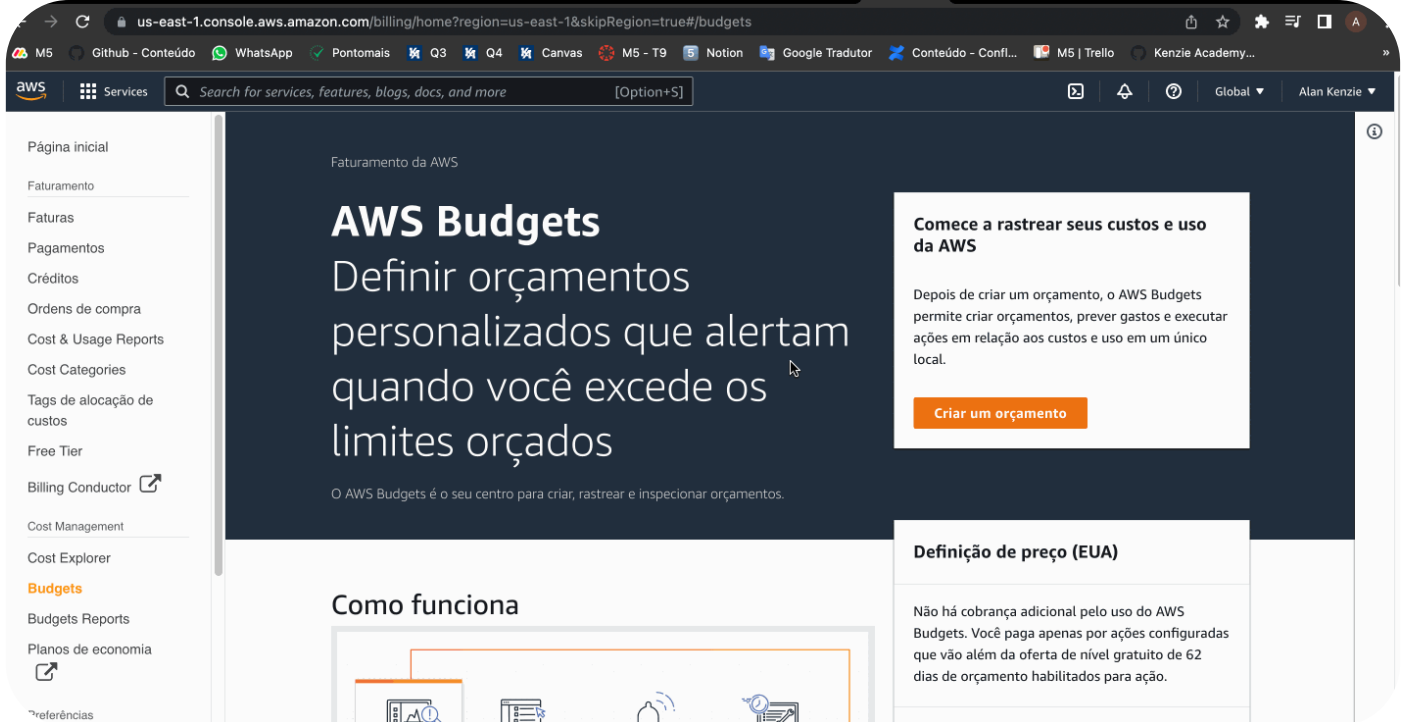
Ao acessar sua conta AWS, procure pelo serviço **AWS Budgets**, para que você possa ter acesso a orçamentos específicos, você precisa habilitar o **Cost Explorer**.



Após habilitar o **Cost Explorer**, vamos criar nosso primeiro orçamento.

Ainda na área da **AWS Budgets**, vamos acessar **Budgets** e após em **Criar um orçamento**.

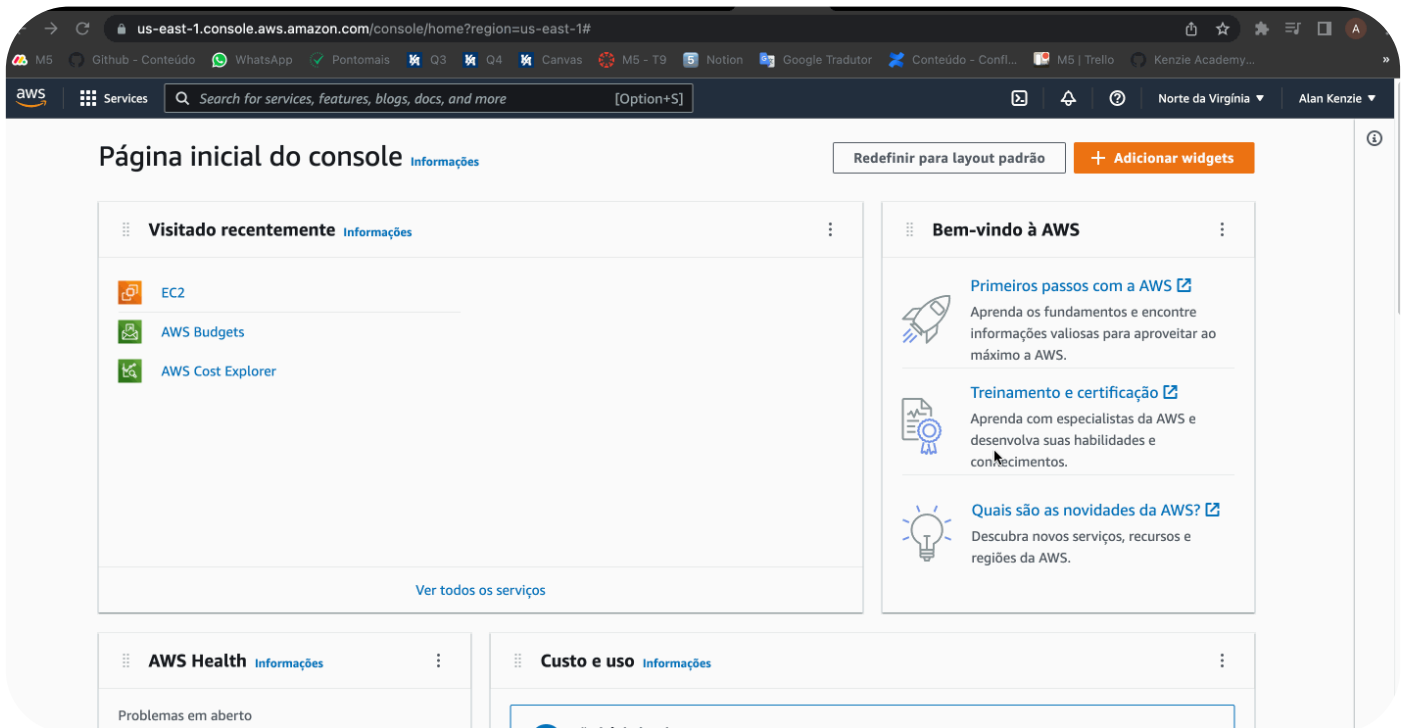
Nesse caso, criamos um orçamento de US\$ 0,01 (um centavo de dólar), e quando chegar a 80% do valor, receberemos uma notificação por e-mail.



Etapa 3 - Criar Instância EC2

1. Procure pelo serviço **EC2**
2. Clique em **Executar Instância**
3. Dê um nome para sua instância
4. Escolha uma imagem de máquina (AMI) dentre as diversas disponíveis
5. Escolha um Tipo de Instância
6. Crie um Par de Chaves para se conectar com segurança à sua instância EC2
7. Clique em **Executar instância**

8. Crie alertas de faturamento para receber alertas quando o nível gratuito estiver prestes a expirar



Importante!

Ao escolher a imagem de máquina (AMI) e o tipo de instância, lembre-se sempre de verificar se estão **qualificados para o nível gratuito**.

Etapa 4 - Conectar à sua instância

1. Por medidas de segurança, vamos mover a chave que baixamos para o diretório `.ssh`. Esse é um diretório que fica oculto em nossa home e guarda todas as informações referentes a permissões com `ssh`.

Copiar para área de transferência

```
mv /path/your-key.pem ~/.ssh/
```

2. Agora precisamos alterar as permissões para que a chave esteja no modo somente leitura, caso contrário, a Amazon considera que a chave está desprotegida e não permite que você a utilize.

Copiar para área de transferência

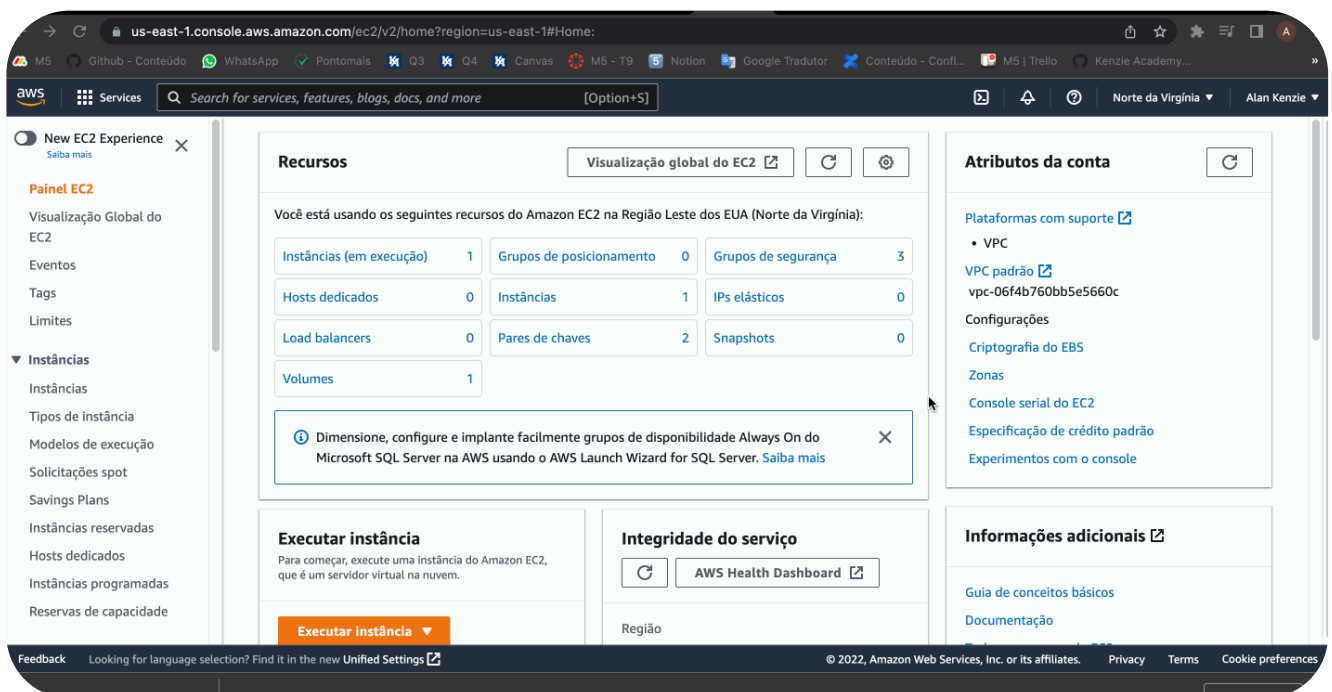
```
chmod 400 ~/.ssh/your-key.pem
```

3. Para realizar a conexão, vamos precisar do username da instância e do DNS público. Existem alguns usernames padrão dependendo de qual imagem foi escolhida para a instância. No caso da imagem do Ubuntu, o username é `ubuntu`. Para outras imagens, verifique [aqui](#) como acessar. Então, no terminal só precisaremos fazer:

Copiar para área de transferência

```
ssh -i ~/.ssh/your-key.pem instance-username@
```

Para termos acesso ao DNS público, podemos acessar a instância e encontraremos a informação em **DNS pública (IPv4)**.



Testando:

Ao seguir os passos acima, entraremos na instância que criamos:

Etapa 5 - Instalar Docker e Docker Compose em nossa instância EC2

Lembre-se que estamos acessando uma VM novinha em folha, então precisamos fazer algumas configurações básicas para rodarmos uma aplicação de exemplo.

Copiar para área de transferência

```
# a partir daqui estamos no shell da instância

sudo apt update
sudo apt install docker
sudo apt install docker-compose
```



```
alanpires — ubuntu@ip-172-31-91-71: ~ — ssh -i ~/.ssh/ec2linux.pem ubuntu@ec2-44-208-166-229.compute-1.amazona...
ubuntu@ip-172-31-91-71:~$ sudo apt-update
sudo: apt-update: command not found
ubuntu@ip-172-31-91-71:~$
```

I

Etapa 6 - Transferir arquivos para nossa instância

Agora vamos transferir nossos arquivos locais para nossa instância EC2 utilizando SCP (protocolo de cópia segura).

1. Saia da instância EC2, para isso, basta digitar o seguinte comando em seu terminal:

Copiar para área de transferência

```
# shell da instância EC2  
exit  
#logout  
#Connection to ec2-44-208-166-229.compute-1.a
```

2. Digite o seguinte comando para transferir os arquivos:

Copiar para área de transferência

```
# no terminal local  
scp -r -i ~/.ssh/your-key.pem ./path/ ubuntu@
```

Para transferência de arquivos via SCP, você deve informar os seguintes itens:

1. O caminho da nossa chave de segurança
2. O caminho onde nossos arquivos estão
3. O username da instância ec2, no nosso caso

ubuntu

4. O DNS público

5. O caminho na instância onde os arquivos serão gravados, nesse caso, `/home/ubuntu/`

3. Acesse a instância EC2 novamente e veja seus arquivos:

Copiar para área de transferência

```
ssh -i ~/.ssh/your-key.pem instance-username@
```

Testando



The image shows a terminal window with a title bar that reads "diaripres - ubuntu@ip-172-31-91-71: ~ - ssh -i ~/.ssh/ec2nix.pem ubuntu@ec2-44-200-100-229.compute-1.amazonaws.com". The terminal content shows the prompt "ubuntu@ip-172-31-91-71:~\$" followed by a cursor. The terminal has a light gray background and a dark gray border.

Etapa 7 - Executando a aplicação na instância EC2

1. Entre no diretório que copiamos com o comando

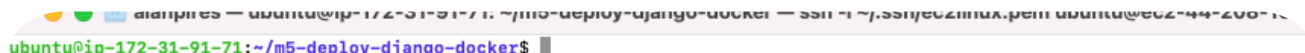
```
cd <directory>
```

2. Já na instância EC2, basta executar o seguinte comando para subir nossos contêineres:

Copiar para área de transferência

```
# shell da instância ec2
```

```
sudo docker-compose up
```

A terminal window with a title bar showing the application name and the current directory. The terminal prompt is 'ubuntu@ip-172-31-91-71: ~/m5-deploy-django-docker\$'.

```
diandres — ubuntu@ip-172-31-91-71: ~/m5-deploy-django-docker — ssh -i ~/.ssh/ec2.pem ubuntu@ec2-44-200-1-...
```

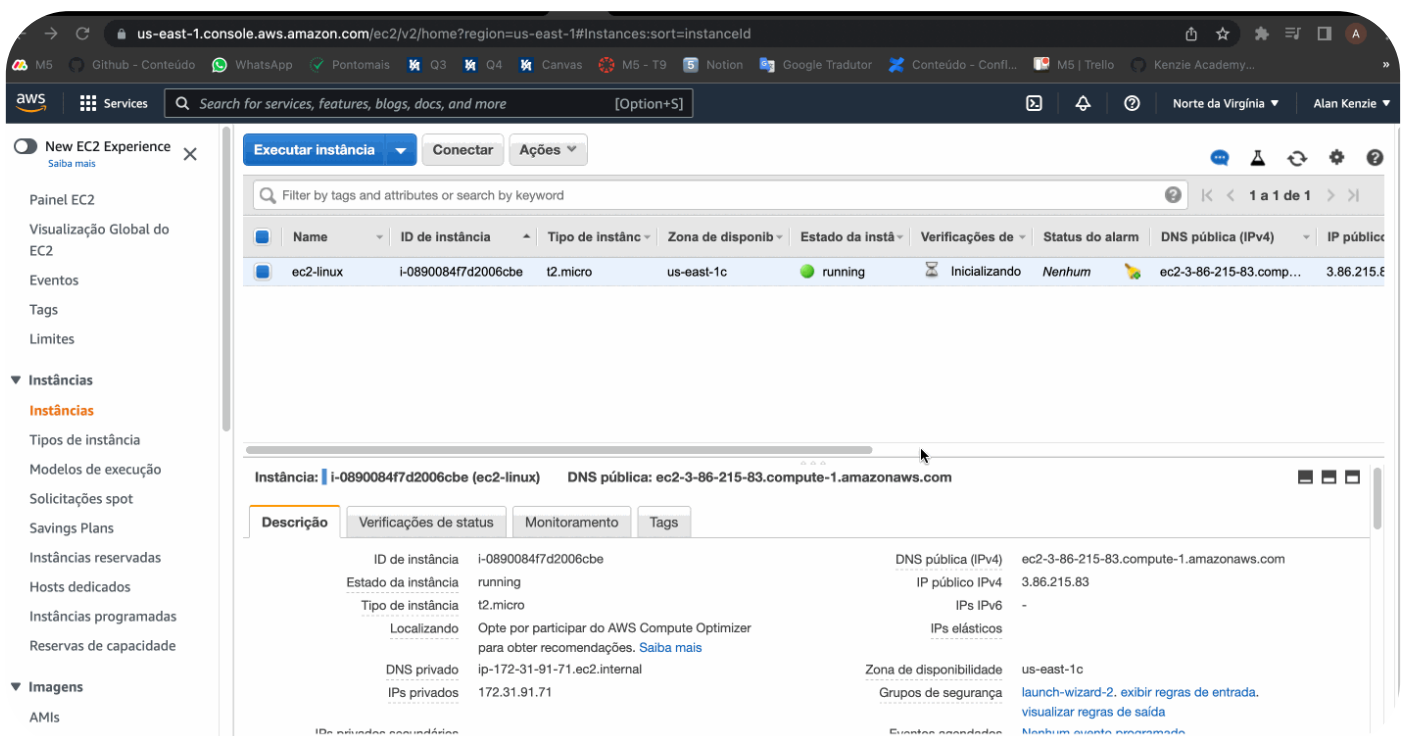
```
ubuntu@ip-172-31-91-71:~/m5-deploy-django-docker$
```

I

Etapa 8 - Ajustando configurações de conexão

Ainda não conseguiremos acessar nossa instância, pois definimos que os acessos à nossa instância seriam apenas via SSH, então vamos ajustar essa configuração nos **Grupos de segurança** da nossa máquina.

Precisamos liberar o tráfego na porta 8000, a qual utilizaremos para rodar nossa aplicação, para isso, vamos alterar a configuração TCP.



The screenshot displays the AWS Management Console for an EC2 instance. The instance is named 'ec2-linux' and is in the 'us-east-1' region. The instance is in the 'running' state. The 'Security Groups' section shows 'launch-wizard-2' with a link to 'visualizar regras de saída'.

Nome	ID de instância	Tipo de instância	Zona de disponib	Estado da instâ	Verificações de	Status do alarm	DNS pública (IPv4)	IP públic
ec2-linux	i-0890084f7d2006cbe	t2.micro	us-east-1c	running	Iniciando	Nenhum	ec2-3-86-215-83.comp...	3.86.215.83

Instância: i-0890084f7d2006cbe (ec2-linux) DNS pública: ec2-3-86-215-83.compute-1.amazonaws.com

Descrição	Verificações de status	Monitoramento	Tags
ID de instância	i-0890084f7d2006cbe		
Estado da instância	running		
Tipo de instância	t2.micro		
Localizando	Opte por participar do AWS Compute Optimizer para obter recomendações. Saiba mais		
DNS privado	ip-172-31-91-71.ec2.internal		
IPs privados	172.31.91.71		
DNS pública (IPv4)	ec2-3-86-215-83.compute-1.amazonaws.com		
IP público IPv4	3.86.215.83		
IPs IPv6	-		
IPs elásticos			
Zona de disponibilidade	us-east-1c		
Grupos de segurança	launch-wizard-2. exibir regras de entrada. visualizar regras de saída		
Eventos	Nenhum evento programado		

Aviso

Perceba que além das configurações **TCP**, também liberamos o acesso da aplicação via **SSH** para protocolos **IPV6**, pois apenas **IPV4** estavam habilitados.

Etapa 9 - **ALLOWED_HOSTS**

Estamos quase lá! Para conseguirmos acessar nossa aplicação em nossa instância EC2, precisamos incluir o endereço DNS em **ALLOWED_HOSTS** em `settings.py` da nossa aplicação.

Abra outro terminal e faça a conexão com a instância EC2.

Copiar para área de transferência

```
ssh -i ~/.ssh/your-key.pem instance-username@yo
```

Na shell da instância acesse a arquivo `settings.py` e insira o endereço DNS da instância EC2 em **ALLOWED_HOSTS**.



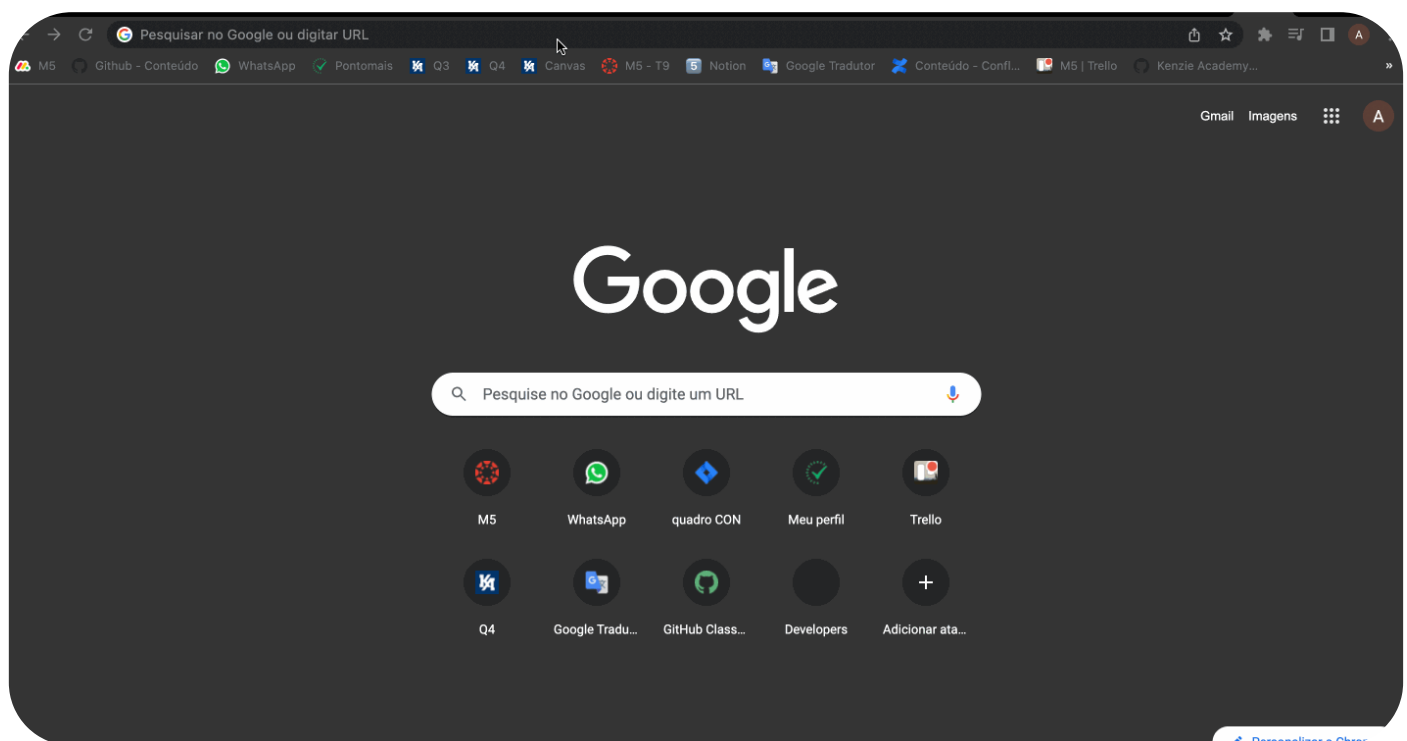
Etapa 10 - Acessando nossa aplicação

Finalmente iremos acessar nossa aplicação, para isso, basta acessarmos o navegador e acessar o endereço do DNS da nossa aplicação e a porta 8000 que liberamos o acesso.

Podemos acessar em `http://your-public-dns-`

`address:8000/api/animals/`

Lembre-se de utilizar **HTTP** ao invés de **HTTPS**, caso contrário você verá um erro de SSL, pois esse é um exemplo rodando num servidor de desenvolvimento e não tem suporte para HTTPS.

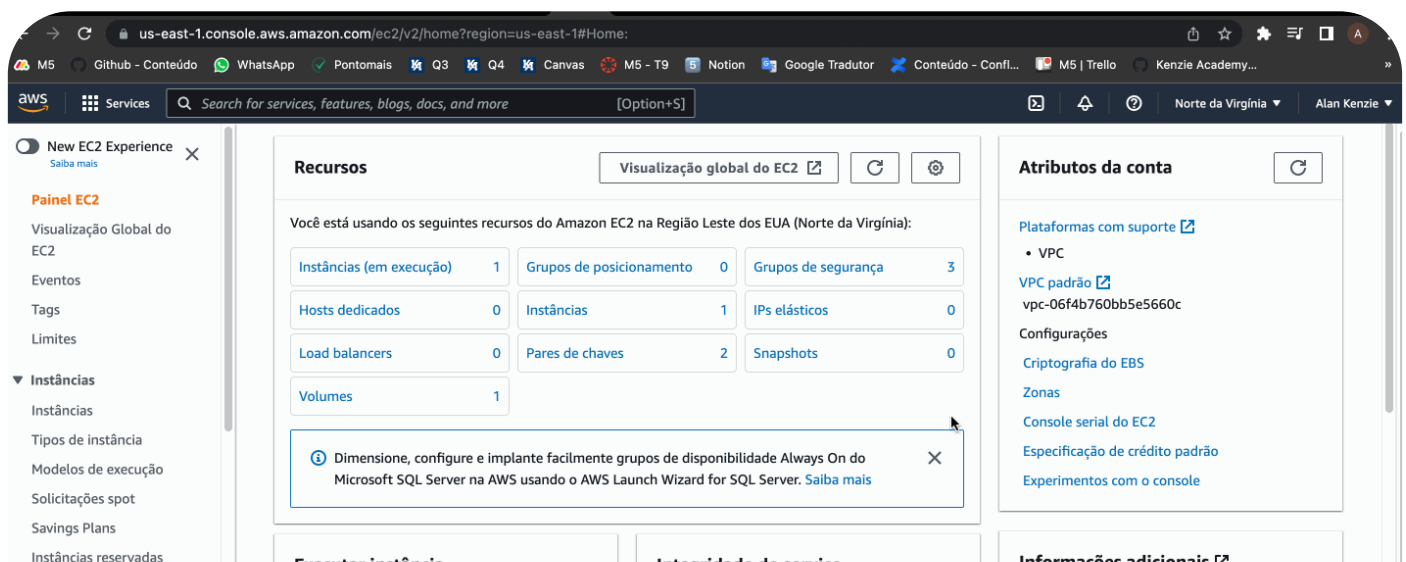


Etapa 11 - Parando sua aplicação

Agora que já acessamos nossa aplicação em nossa instância EC2, está na hora de interrompermos o

serviço para não termos problemas com cobranças indevidas e surpresas indesejáveis.

Acesse a página onde mostram suas instâncias que estão em execução, clique em **Ações -> Estado da Instância -> Interromper**



Referências!

[Amazon EC2 | AWS](#)

[Conceitos básicos do Amazon EC2 | AWS](#)

[Conecte-se à sua instância do Linux usando SSH | AWS](#)

Comece a usar instâncias de Linux do Amazon EC2 | AWS

Pré-requisitos gerais para conectar-se à instância | AWS