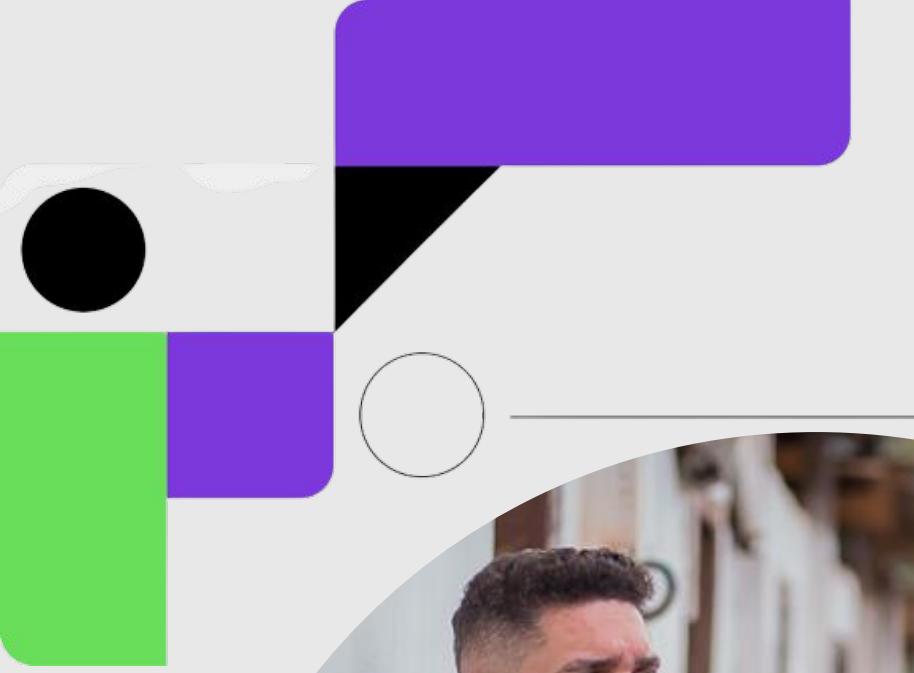


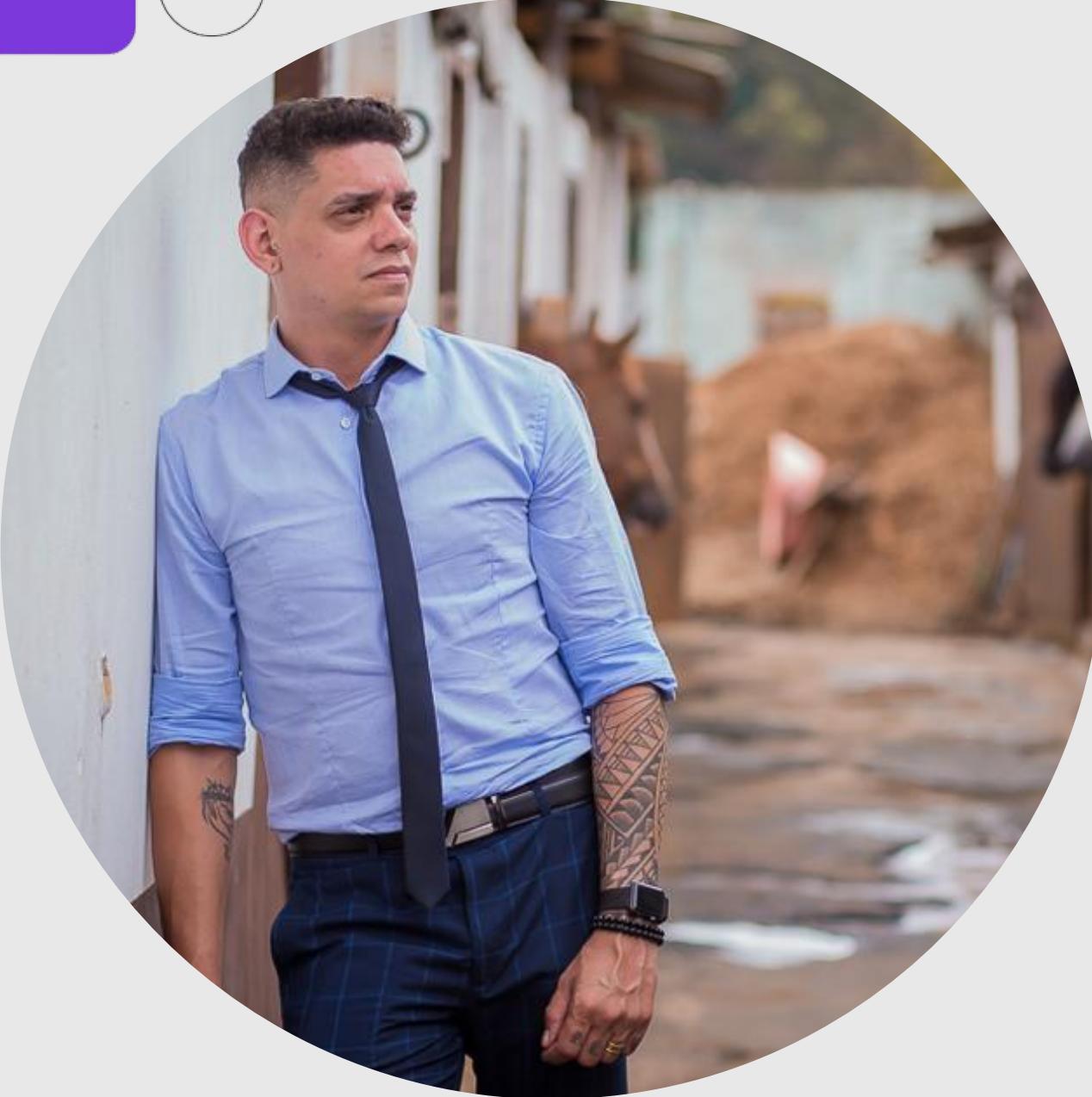


AWS

Material Complementar



Douglas Morais



34 anos, atua no mercado de desenvolvimento a cerca de 14 anos atualmente Techlead na Kenlo. Apaixonado por Tecnologia e todas as novidades do segmento.

Especialista em desenvolvimento frontend um eterno entusiasta do desenvolvimento mobile.



linkedin.com/in/douglasmoraisdev/



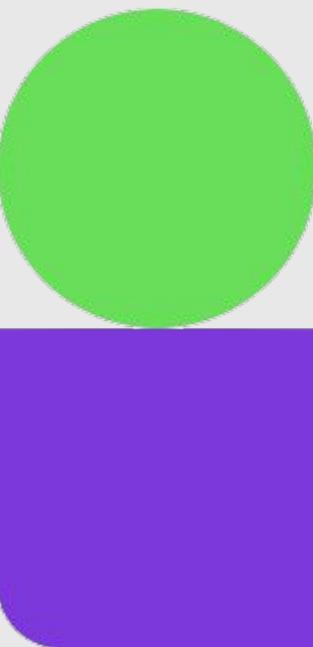
<https://github.com/mrdouglasmorais>

#PraCegoVer: Fotografia do autor Douglas
Morais.

Índice

- + [Segurança na nuvem AWS IAM.](#)
- + [VPC's e gestão de DNS com AWS Route53.](#)
- + [Servidores com AWS EC2](#)
- + [Balanceadores de carga \(load balancers\)](#)
- + [Funções públicas com AWS Lambda](#)
- + [Armazenamento e cacheamento AWS S3](#)
- + [Banco de dados AWS RDS](#)
- + [Auto scaling com AWS EKS](#)

Introdução



O mercado de tecnologia vem evoluindo com um dinamismo impressionante e hoje as soluções da AWS (Amazon Web Services) são uma das mais completas e mais utilizadas no setor.

Nesta apostila você vai entender as principais soluções de dentro do maior provedor de Cloud Computing do mercado nos dias atuais.

Segurança na nuvem

Gerenciamento de usuários e grupos

Vamos falar sobre o **AWS IAM (Identity Access Management)**.

E qual o papel do **IAM**?

Ele é responsável por gerenciar as permissões de usuários a nível de criar ou alterar serviços dentro da plataforma.

Ou seja com as configurações de **IAM** é possível iniciar um serviço novo dentro da AWS a partir de um novo projeto, nos passos a seguir teremos um pouco mais de clareza a nível de processos.

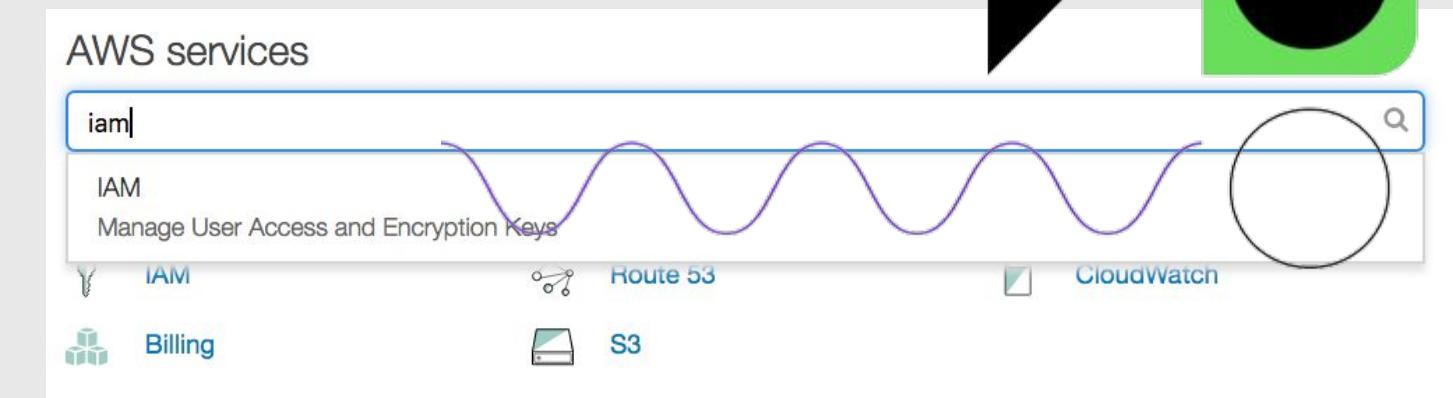
Agora vamos colocar a mão na massa e aprender a configurar no **IAM**.

Configurando nosso IAM - parte 1

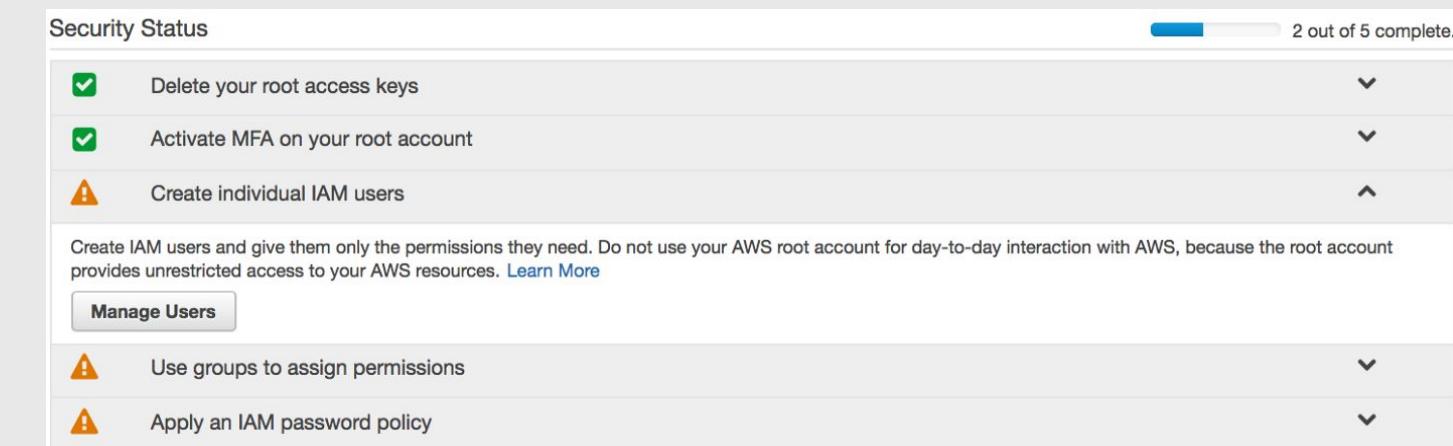
Acessando o console da AWS, caso ainda não saiba como criar uma conta clique a [aqui](#) e siga o passo a passo neste tutorial do medium.

Dentro do seu console, você vai encontrar os principais serviços AWS, conforme a imagem ao lado e na caixa de busca digite "IAM" e em seguida clique no resultado apresentado.

Em seguida devemos criar o usuário conforme a imagem ilustra ao lado, clique em **Manage Users**.



#PraCegoVer: Imagem com console da amazon, busca de serviços acionada com o termo "iam".



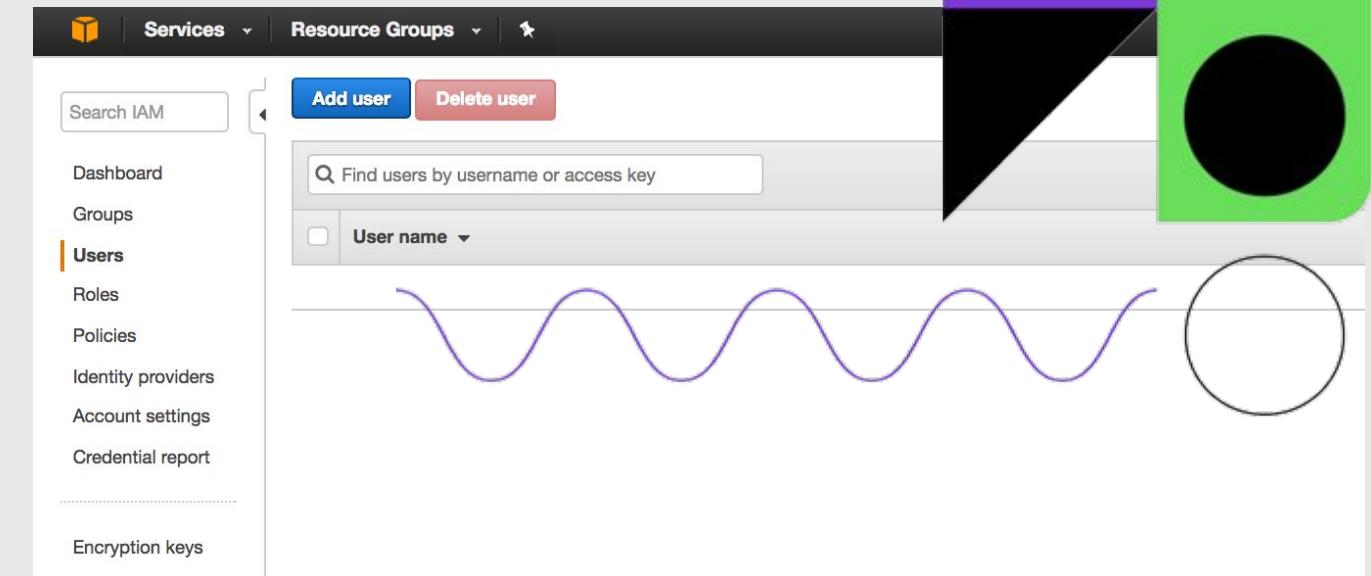
#PraCegoVer: Imagem ilustrativa com console da AWS configurações iniciais do IAM após o passo mencionado na primeira imagem

Configurando nosso IAM - parte 2

A seguir vamos criar 2 grupos e 4 usuários para gerir os recursos por meio do console.

O Grupo **Devops** com os seguintes usuários "Adilson" e "Carlos".

O Grupo **SysAdmin** com os usuários "Anderson" e "Fábio".

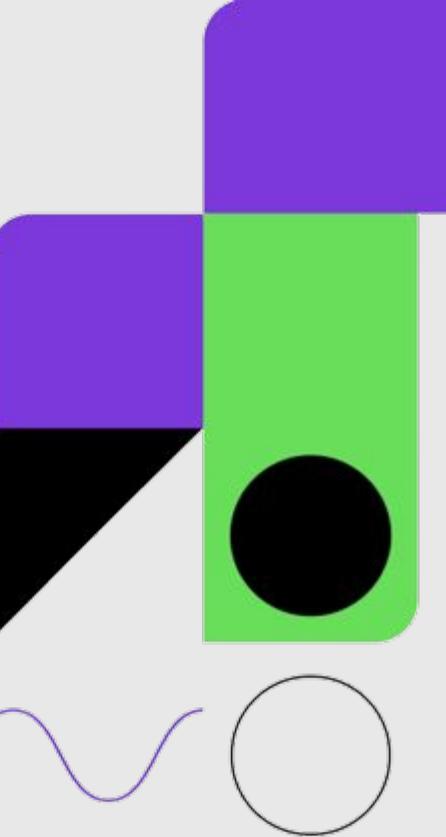


#PraCegoVer: Imagem com console da AWS, para criação de novos usuários.

A screenshot of the 'Add user' process at step 2, 'Permissions'. It shows two users added: 'adilson' and 'carlos'. Under 'Select AWS access type', both 'Programmatic access' and 'AWS Management Console access' are checked. Under 'Console password', 'Autogenerated password' is selected. A note says 'Require password reset' is checked. At the bottom right are 'Cancel' and 'Next: Permissions' buttons.

#PraCegoVer: Imagem ilustrativa com console da AWS configurações passo a passo de informações e acessos de novos usuários.

Configurando nosso IAM - parte 3



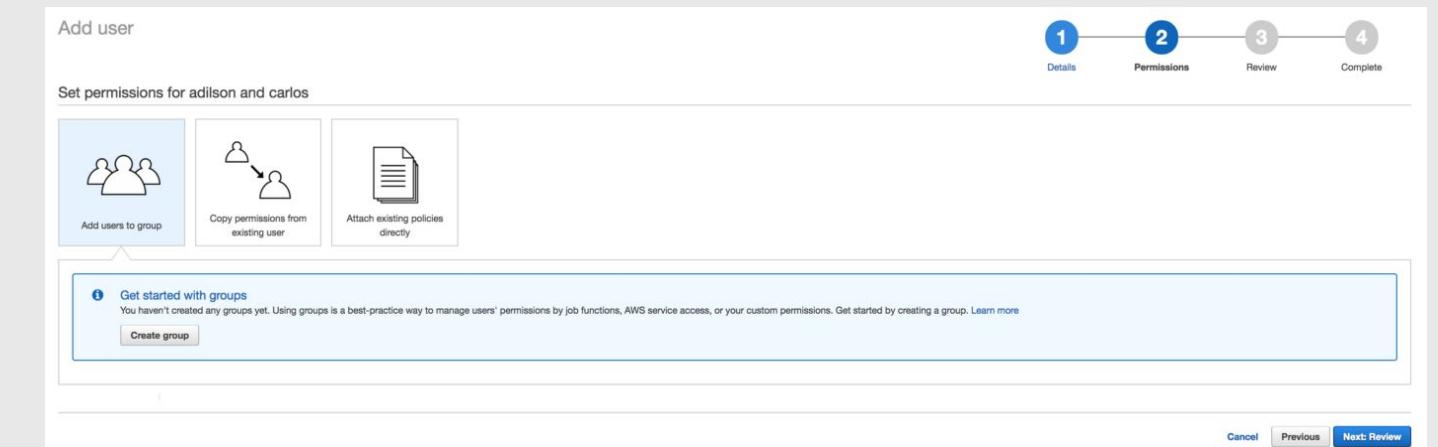
Selecionando os tipos de acesso:

Programmatic access: Habilita **access key ID** and **secret access key** para acesso às ferramentas de desenvolvimento AWS API, CLI, SDK, etc.

AWS Management Console access: Habilita **senha** que permite que o usuário tenha acesso no console da AWS.

Require password reset: Habilita usuário a trocar a senha no próximo login e automaticamente ter acesso a política **IAMUs ChangePassword** que permite mudança da própria senha.

Vamos a criação dos grupos clicando em *Next Permissions*:



#PraCegoVer: Imagem ilustrativa com console da AWS configurações passo a passo de informações e acessos de novos usuários passo 2.

Configurando nosso IAM - parte 4



Criando o grupo devops com acessos FULL ao AWS S3.

Vamos seguir os mesmos passos para criação do grupo **sysAdmin** os usuários "Anderson" e "Fábio" terão acessos FULL apenas ao EC2.

Importante: Antes de escolher os grupos, ler com atenção a descrição (*Item Description*).

*Faça o download do arquivo .csv vais ser necessário fazer uso de 2 dados **Access Key ID** e **Secret Access Key** para acesso por meio da linha de comando [AWS CLI](#).

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name: devops

Create policy Refresh

Filter: Policy type s3

Policy name	Type	Attachments	Description
AmazonDMSRedshiftS3Role	AWS managed	0	Provides access to manage S3 settings for Redshift endpoints for DMS.
AmazonS3FullAccess	AWS managed	0	Provides full access to all buckets via the AWS Management Console.
AmazonS3ReadOnlyAccess	AWS managed	0	Provides read only access to all buckets via the AWS Management Console.

#PraCegoVer: Imagem ilustrativa com console da AWS configurações passo a passo de informações e acessos de grupos passo 4.

Add users to an existing group or create a new one. Using groups is a best-practice way to manage users' permissions by job functions. [Learn more](#)

Create group Refresh

Search

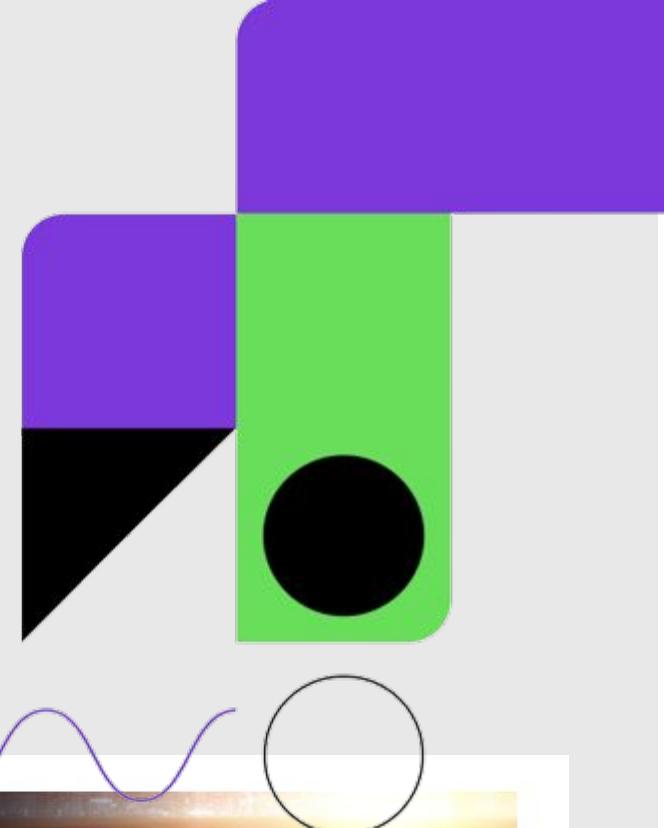
Group	Attached policies
devops	AmazonS3FullAccess

#PraCegoVer: Imagem ilustrativa com console da AWS confirmação e seleção de grupo de acesso.

Acessando a conta recém criada

Agora vamos acessar o nosso usuário "Adilson" no console da AWS página de login, caso não encontre o mesmo clique aqui.

Selecione os acessos por meio de um IAM e informe o **account ID** conforme a imagem a seguir (o account ID está no arquivo CSV que baixamos nos passos anteriores).



Sign in

Root user
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user
User within an account that performs daily tasks. [Learn more](#)

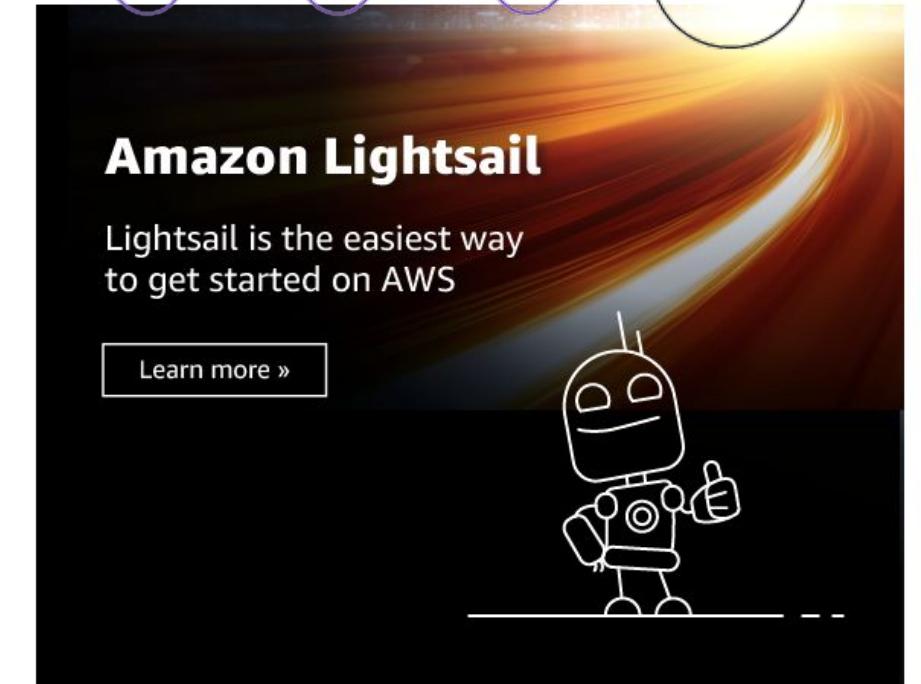
Account ID (12 digits) or account alias

[Next](#)

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

New to AWS?

[Create a new AWS account](#)

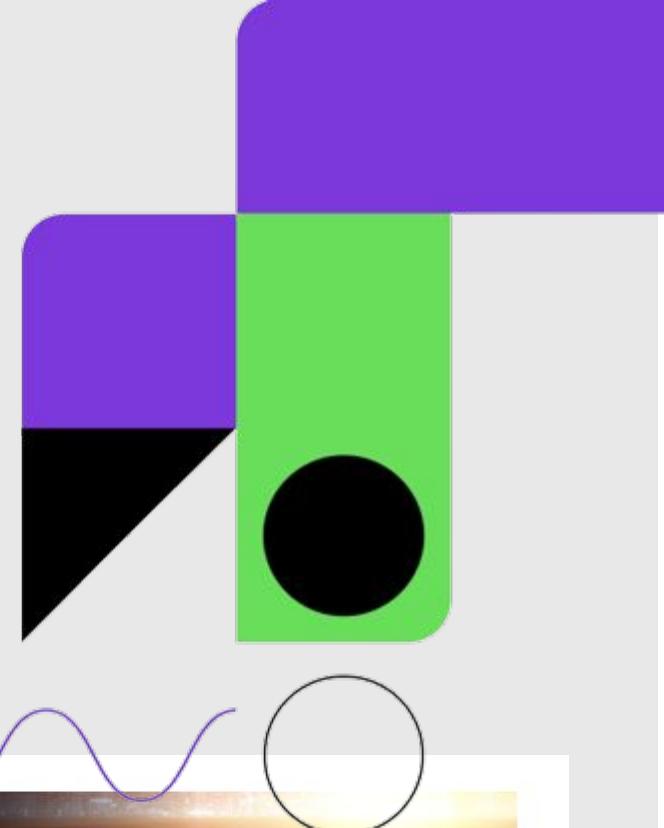


#PraCegoVer: Imagem ilustrativa tela de login AWS acessos por meio de IAM.

Acessando a conta recém criada

Agora vamos acessar o nosso usuário "Adilson" no console da AWS página de login, caso não encontre o mesmo clique aqui.

Selecione os acessos por meio de um IAM e informe o **account ID** conforme a imagem a seguir (o account ID está no arquivo CSV que baixamos nos passos anteriores).



Sign in

Root user
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user
User within an account that performs daily tasks. [Learn more](#)

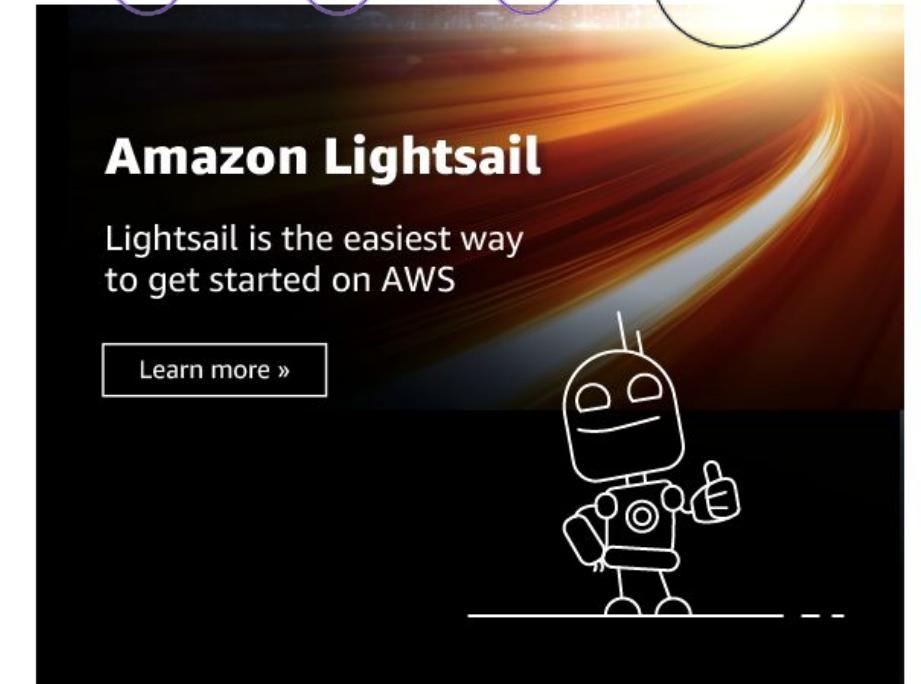
Account ID (12 digits) or account alias

[Next](#)

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

New to AWS?

[Create a new AWS account](#)



#PraCegoVer: Imagem ilustrativa tela de login AWS acessos por meio de IAM.

Finalizando nosso IAM

E para finalizar as configurações, devemos criar as configurações de políticas de acesso e senha dos nossos usuários.

Lembrando que não existe um regra em específico para este tipo de configuração.

*Dica utilize critérios de segurança conforme a sua necessidade.

▼ Password Policy

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to [Managing Passwords](#) in Using IAM.

Modify your existing password policy below.

Minimum password length:

Require at least one uppercase letter [i](#)

Require at least one lowercase letter [i](#)

Require at least one number [i](#)

Require at least one non-alphanumeric character [i](#)

Allow users to change their own password [i](#)

Enable password expiration [i](#)

Password expiration period (in days):

Prevent password reuse [i](#)

Number of passwords to remember:

Password expiration requires administrator reset [i](#)

[Apply password policy](#) [Delete password policy](#)

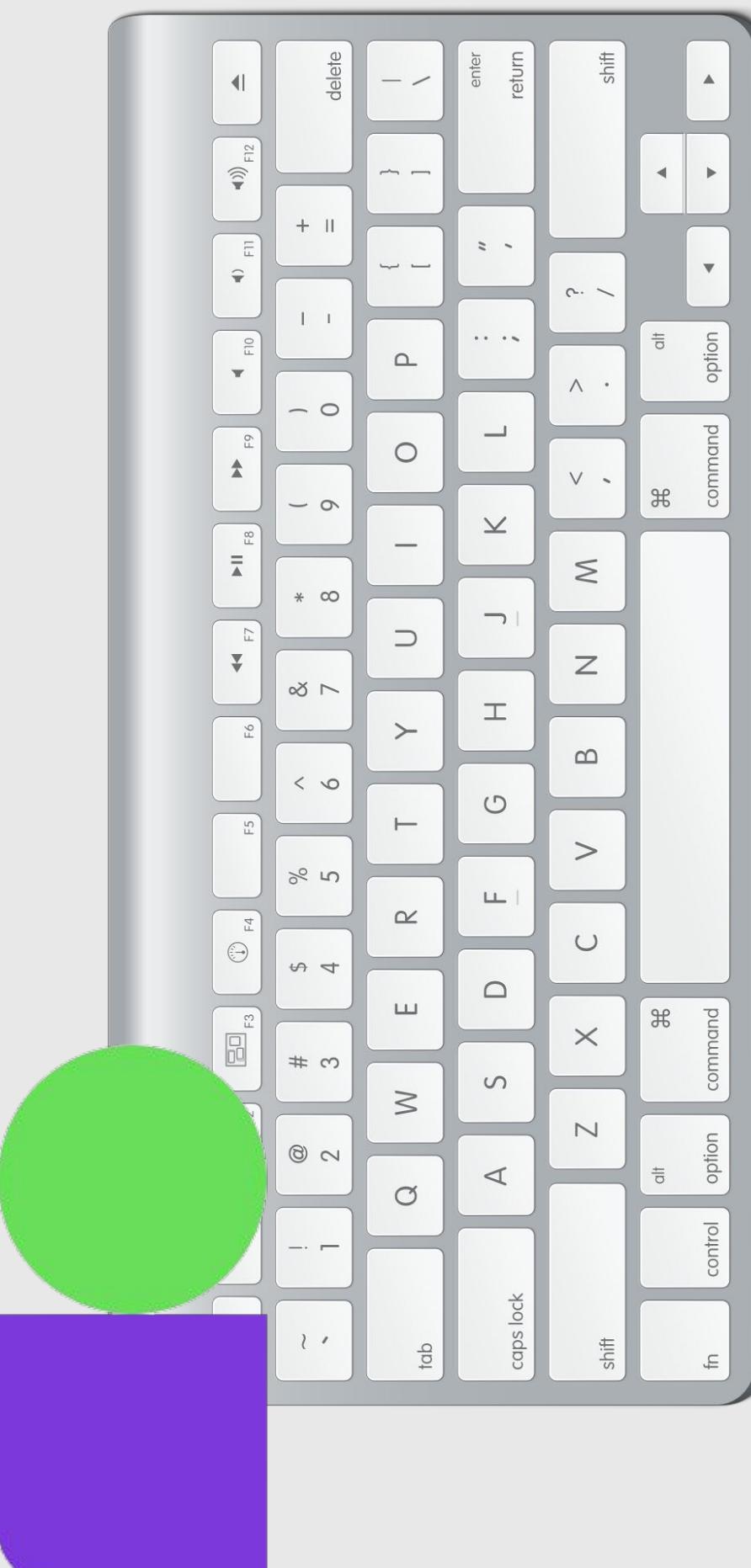
#PraCegoVer: Imagem ilustrativa tela console AWS acessos por meio de IAM.

VPC e Gerenciamento de DNS com AWS route 53.

Antes de qualquer coisa, o que é o route 53?

Basicamente ele é um Web Service Domain System com gerenciamento refinado e o mais importante escalável.

Permite vincular serviços por meio de IP a domínios e subdomínios em um exemplo mais prático, se você tem uma aplicação com o seguinte endereço ex: *myapp.io* onde fica armazenado a interface de acesso do seu cliente você pode armazenar e gerenciar micro serviços dentro do mesmo domínio ex: *api.myapp.io* por meio de IP dos serviços.

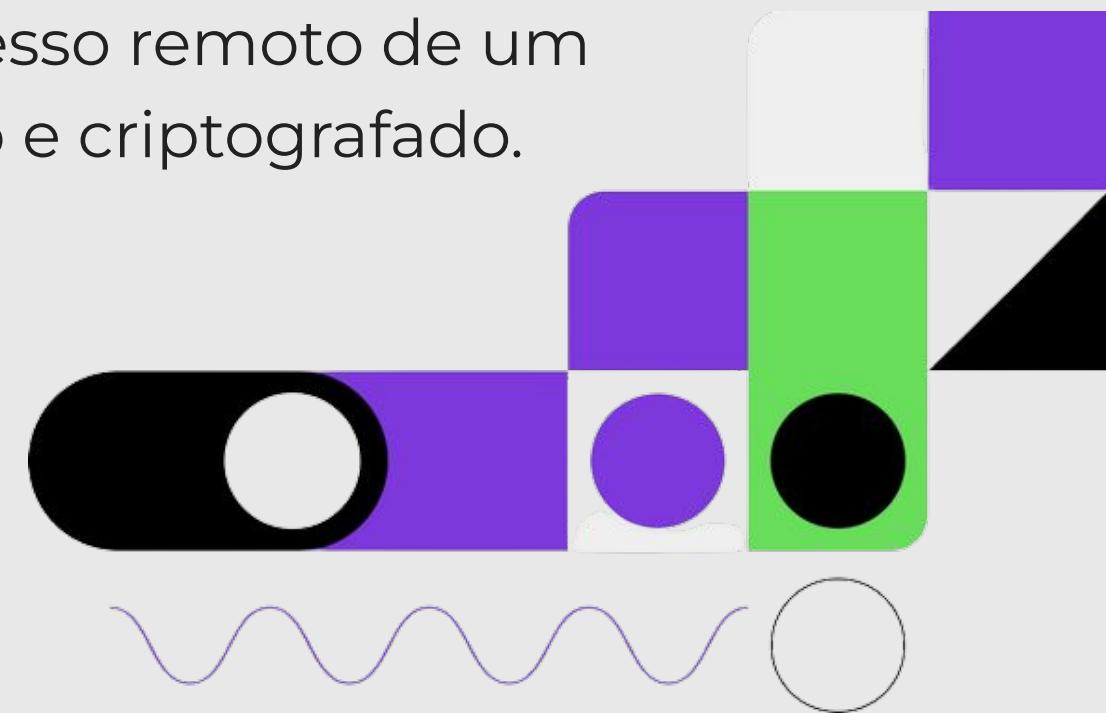


VPC o que é?

Virtual private cloud (VPC) ou **nuvem privada virtual** é uma demanda configurável de recursos compartilhados de computação alocados dentro de um ambiente de **nuvem** pública, fornecendo um certo nível de isolamento entre as diferentes organizações (denotado como usuários), utilizando os recursos. O isolamento entre um usuário VPC e todos os outros usuários da mesma nuvem (outros usuários VPC, bem como outros usuários da nuvem pública) é feito normalmente através da atribuição de uma sub-rede **IP** privada e um recurso de comunicação virtual (tal como uma **VLAN** ou um conjunto de canais de comunicação **criptografada**) por usuário.^[1]

Em uma VPC o mecanismo acima descrito para criar o isolamento dentro do cloud, é complementado com um serviço de VPN que garante a segurança do acesso remoto de um usuário aos seus próprios recursos através de um sistema de autenticação e criptografado.

Fonte: wikipedia
https://pt.wikipedia.org/wiki/Virtual_private_cloud





VPC e gestão de DNS

Neste módulo vamos aprender a gerenciar recursos de DNS e VPC no console da AWS

#PraCegoVer: Dois notebooks
o primeiro apresentando o teclado
o segundo apresentando o
monitor.

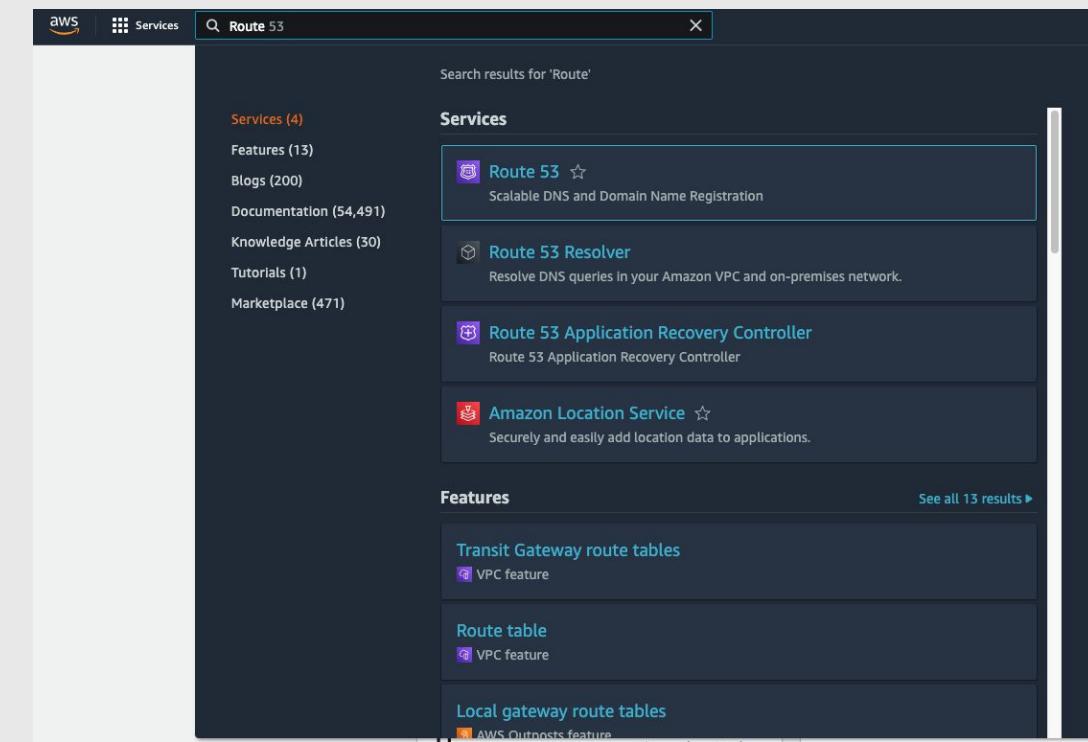
Route 53

Para localizar o route 53 no console da AWS basta você introduzir na barra de pesquisa Route 53 e selecionar o serviço.

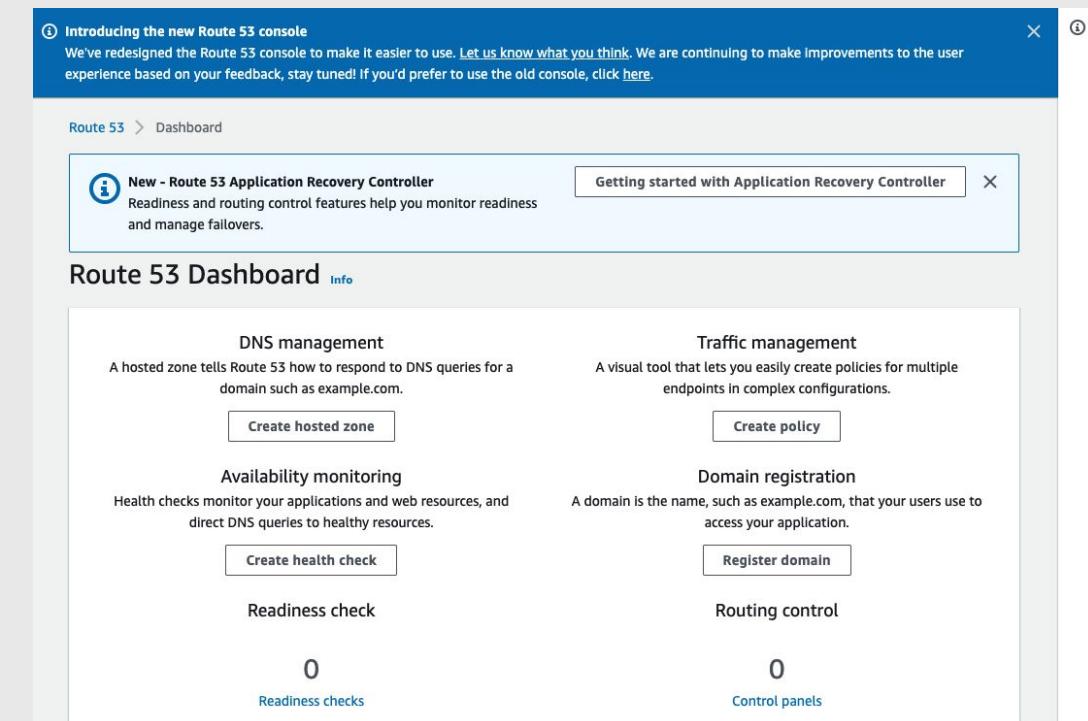
Após selecionado o painel estará disponível para uso.

Antes de qualquer coisa é necessário ter um domínio que possa ser gerenciado através do painel.

É possível contratar novos domínios pelo próprio painel route 53 ou gerenciar fazendo os devidos apontamentos do register para a AWS.



#PraCegoVer: Console AWS pesquisa do serviço route53



#PraCegoVer: Console AWS painel route53

Tipos de Registros



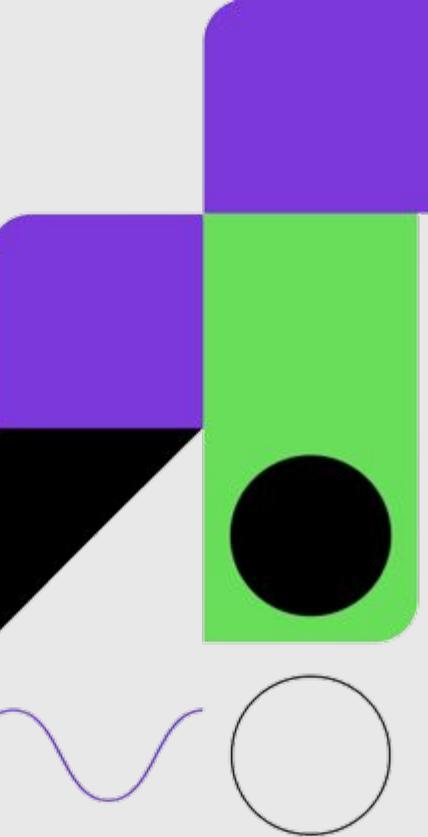
Os tipos de registro de DNS com suporte pelo route 53 está disponível na documentação oficial da AWS e você clique [aqui](#) para visualizar.



Os tipos de registros são responsáveis por apontamentos de emails (MX e TXT) ou seja o seu domínio pode estar sendo gerenciado dentro do painel AWS e utilizar servidores de emails do Google ou até mesmo dentro de qualquer outro provedor de sua preferência.

Gerenciamento de subdomínios podendo fazer uso de de soluções distintas. Ex: em um SAAS (Software As A Service) existe uma aplicação institucional ([nomedasolucao.com.br](#)) e uma outra aplicação distinta para uma área logada ([app.nomedasolucao.com.br](#)) o gerenciamento é comumente realizado por um registro tipo A dentro do próprio painel .

VPC Resolvers



Para entender um pouco melhor sobre os Resolvers do Route 53 você encontra na documentação oficial clicando [aqui](#).

Utilizado para criar endpoints tanto para entrada quanto para saída de dados garantindo mais segurança para sua aplicação, ele é capaz de receber regras para acesso que funcionam basicamente como middlewares que permite ou não ser acessado a partir de uma origem.

The screenshot shows the AWS Route 53 Resolver VPCs interface. At the top, there's a navigation bar with 'Route 53 > Resolver > VPCs'. Below it, a message says 'You are signed in to the following Region: us-east-1 (N. Virginia) To change your Region, use the Region selector in the upper-right corner.' The main area is titled 'VPCs (1) Info' and contains a table with one row. The table columns are: ID, Name, Rules, Outbound endpoints, Inbound endpoints, Query logging status, and Query logging configs. The single row shows: ID 'vpc-Of5f79afa959f9131', Name '-', Rules '1', Outbound endpoints '0', Inbound endpoints '0', Query logging status 'Not set up', and Query logging configs '0'.

#PraCegoVer: Imagem ilustrativa do Console AWS painel route53 resolver

AWS EC2

O que é EC2?

São recursos computacionais alocados por instâncias ou seja máquinas virtuais onde é possível gerenciar recursos como memória, HD, sistema operacional (windows ou linux) e afins para infinitos usos.

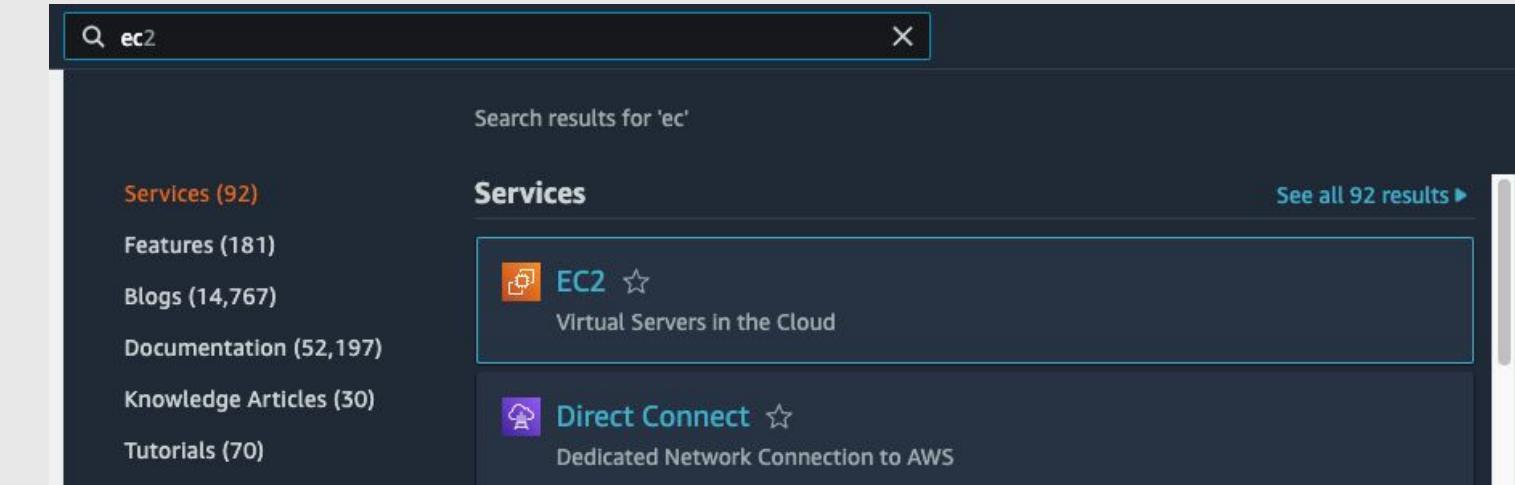
Ex: Alocação de micro serviços até mesmo uma aplicação monolítica onde é possível verticalizar uma solução centralizando na mesma instância o frontend e o backend da solução (este tipo de uso não é indicado para aplicações que tem a necessidade de escalar).

Criando uma instância EC2 - parte 1

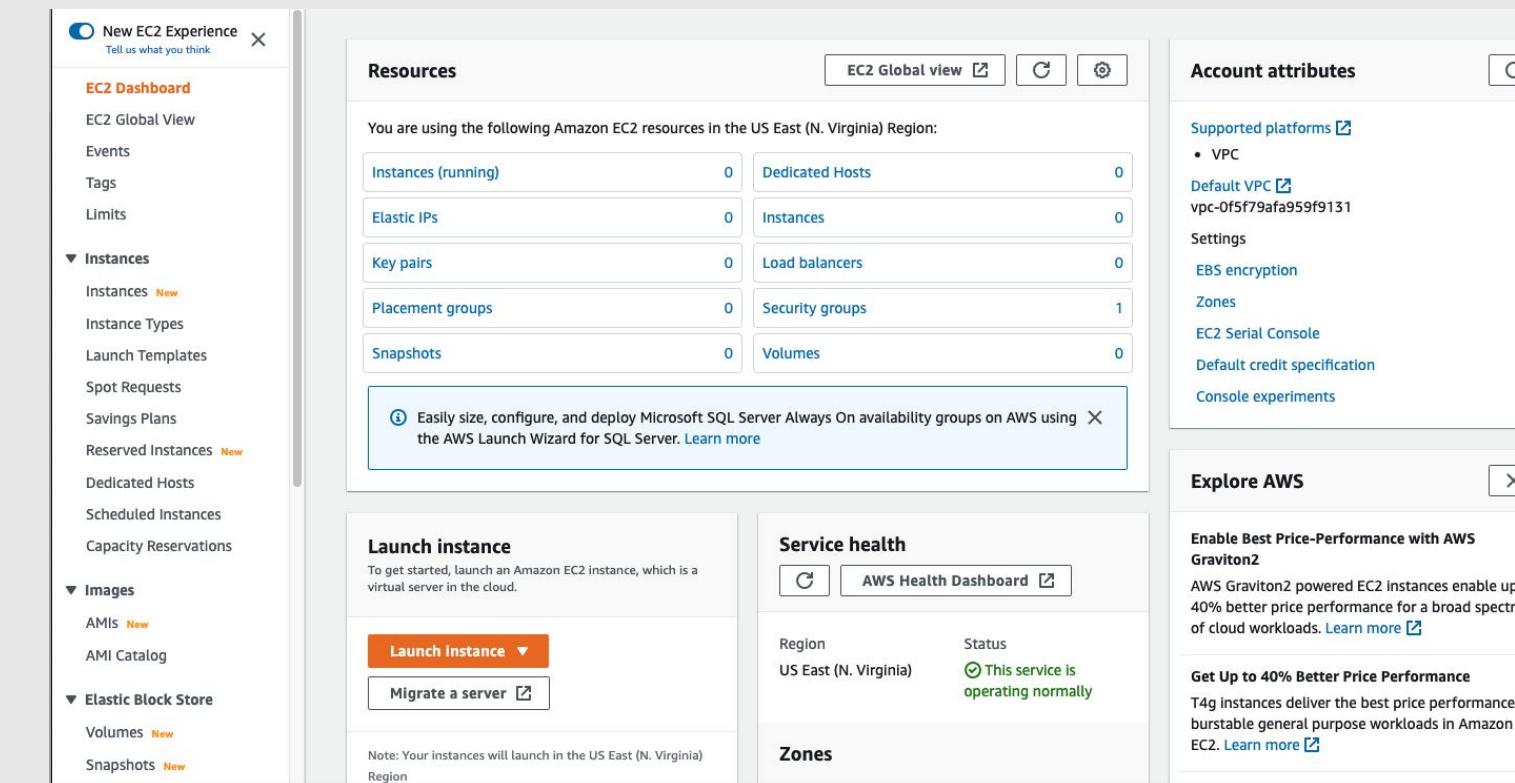
Para criar uma instância EC2 é necessário acessar o serviço, dentro do console no mecanismo de busca escreva EC2 conforme a imagem a direita.

Em seguida você será redirecionado para o painel do serviço, neste painel é possível enxergar a região na qual está logado e o status dos respectivos serviços habilitados até o presente momento.

Para criar uma nova instância devemos clicar no botão **Launch Instance** em seguida teremos alguns modelos prontos para fazer uso além da possibilidade de iniciar instâncias a partir de templates.



#PraCegoVer: Console AWS busca por serviços ec2



#PraCegoVer: Console AWS painel ec2

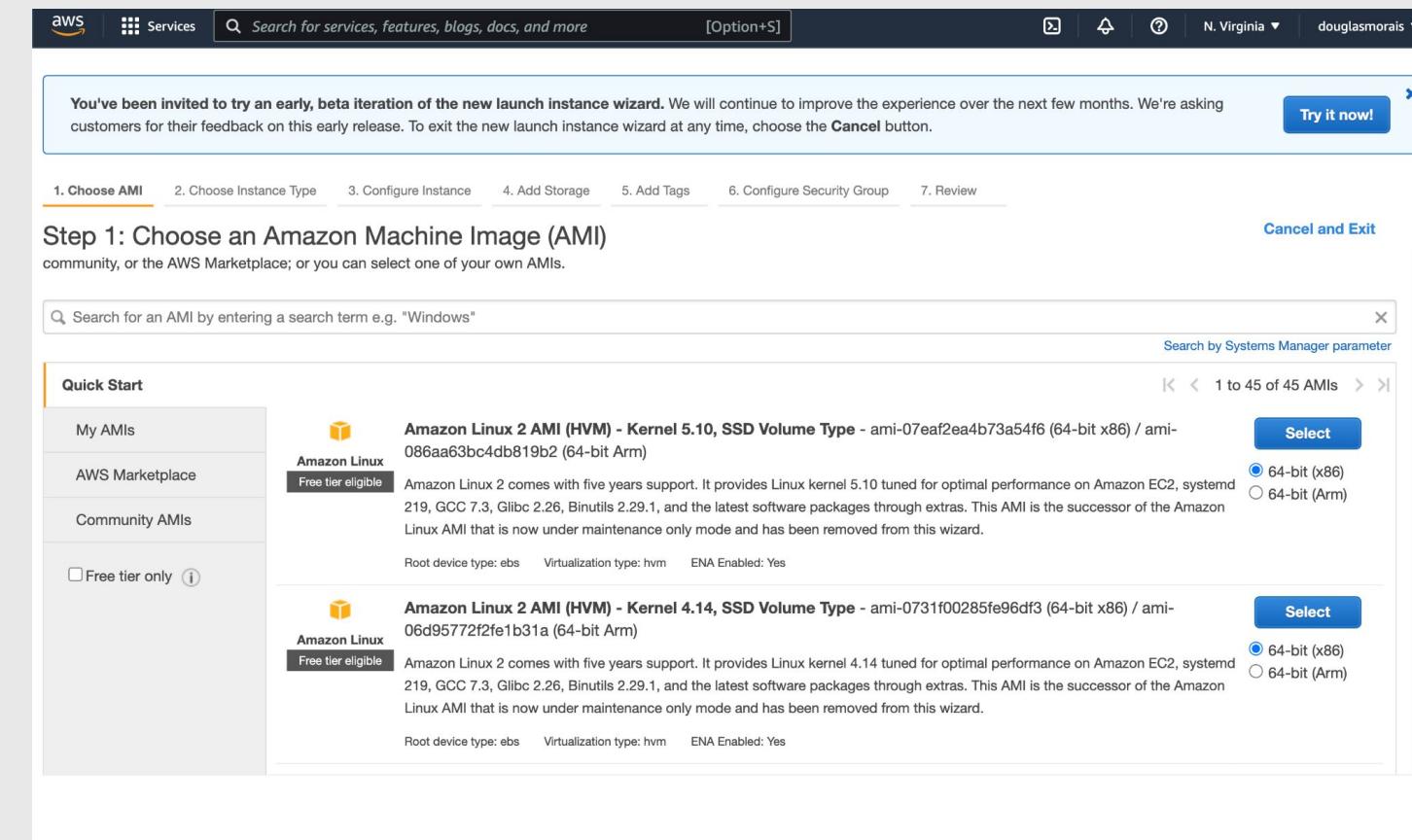
Criando uma instância EC2 - parte 2

Podemos iniciar nosso servidor fazendo uso de uma imagem dentro do **Quick Start** no caso uma máquina linux com 64bits para fazer uso devemos clicar em select ao lado da imagem que desejamos fazer uso.

Em seguida devemos selecionar o tamanho da instância que vamos criar, para entender de forma detalhada tamanho e preço de instâncias EC2 clique [aqui](#) e visualize na documentação oficial.

Em nosso exemplo vamos selecionar uma camada gratuita de serviços (free tier) em seguida clique em

Next.



#PraCegoVer: Console AWS ec2 quick start selecionando imagens de instâncias.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

#PraCegoVer: Console AWS painel ec2 seleção de tamanho de instâncias.

Criando uma instância EC2 - parte 3

Para os passos a seguir devemos selecionar os usuários que terão acessos o **IAM** e demais configurações.

Para o nosso exercício vamos apenas clicar em **Next** para adicionar o storage associado a nossa instância.

Para mais de detalhes de configurações de Ec2, subnet, vpc e afins vou deixar um vídeo com um material de apoio para maior compreensão desta solução, clique [aqui](#) para acessar o vídeo do canal **Um Inventor Qualquer**.

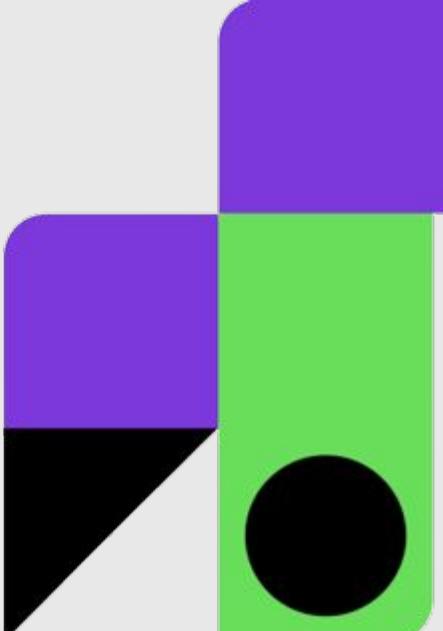
The screenshot shows the 'Step 3: Configure Instance Details' page of the AWS EC2 instance creation wizard. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags, Configure Security Group, and Review. The 'Configure Instance' tab is selected. The main section is titled 'Step 3: Configure Instance Details' with the sub-instruction 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' Below this, there are several configuration fields:

- Number of instances:** Set to 1, with a link to 'Launch into Auto Scaling Group'.
- Purchasing option:** A checkbox for 'Request Spot instances' is unchecked.
- Network:** Set to 'vpc-0f5f79afa959f9131 (default)', with a link to 'Create new VPC'.
- Subnet:** Set to 'No preference (default subnet in any Availability Zone)', with a link to 'Create new subnet'.
- Auto-assign Public IP:** Set to 'Use subnet setting (Enable)'.
- Hostname type:** Set to 'Use subnet setting (IP name)'.
- DNS Hostname:** Includes checkboxes for 'Enable IP name IPv4 (A record) DNS requests' (unchecked), 'Enable resource-based IPv4 (A record) DNS requests' (checked), and 'Enable resource-based IPv6 (AAAA record) DNS requests' (unchecked).
- Placement group:** A checkbox for 'Add instance to placement group' is unchecked.
- Capacity Reservation:** Set to 'Open'.
- Domain join directory:** Set to 'No directory', with a link to 'Create new directory'.
- IAM role:** Set to 'None', with a link to 'Create new IAM role'.
- Shutdown behavior:** Set to 'Stop'.

At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage'.

#PraCegoVer: Console AWS ec2 configure instances passo 3.

Criando uma instância EC2 - parte 4



Neste passo devemos selecionar o espaço em disco de nossa instância.

Atenção os serviços amazon são cobrados por demanda, e para selecionar o tamanho do seu storage não é permitido o downgrade.

Em nosso exercício vamos manter o tamanho padrão de 8gb em seguida clique em **Next**.

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard at Step 4: Add Storage. The steps are: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (highlighted), 5. Add Tags, 6. Configure Security Group, 7. Review.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0bea3fd719c0a0d36	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/> Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

Add file system

Cancel Previous Review and Launch Next: Add Tags

#PraCegoVer: Console AWS ec2 configure instances passo 4 adicionando storage.

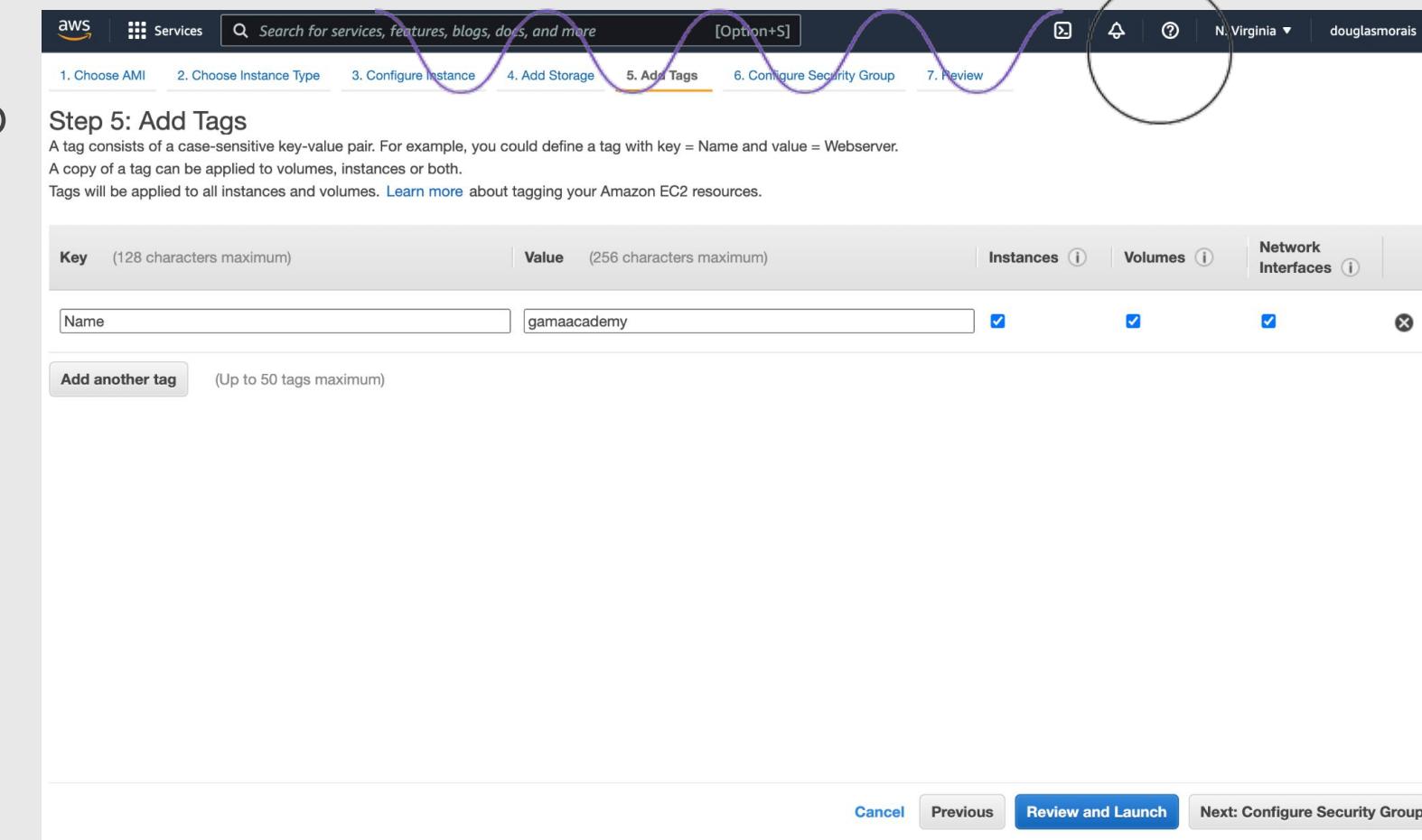
Criando uma instância EC2 - parte 5



Para o passo a seguir podemos adicionar tags ao nosso serviço, mas no final de contas o que são estas TAGs?

Com uso das TAGs podemos acompanhar o consumo de nossa instância tanto pelo painel de custos quanto pelo painel do EC2. Imagine que em nosso painel temos muitos micro serviços e de alguma maneira é necessário ter um acompanhamento regular dos mesmos, esta TAGs nos ajudam neste sentido.

Em nossas tags teremos o valor **gamaacademy** em seguida clique em **Next**.



#PraCegoVer: Console AWS ec2 configure instances passo 5 adicionando tags.

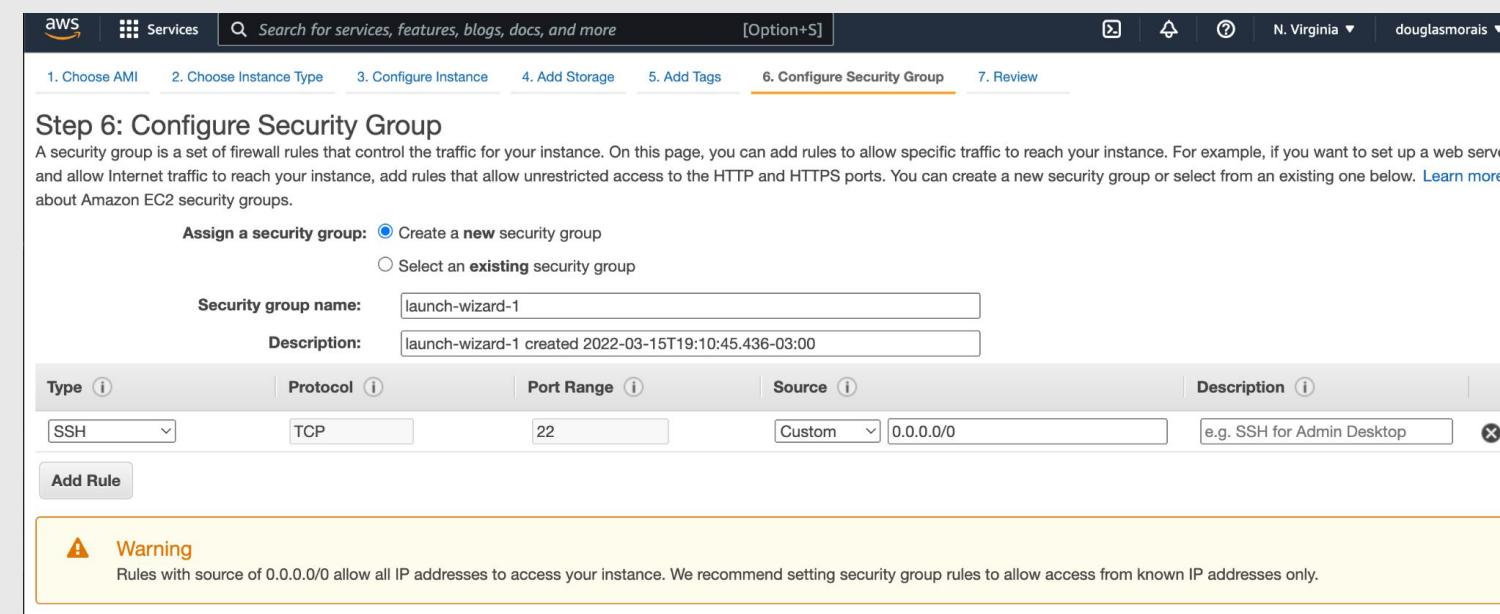
Criando uma instância EC2 - parte 6

Por fim devemos escolher um grupo de segurança já existe ou criar um novo. Em nosso exercício vamos criar deixar selecionado para criar um novo grupo de segurança, em seguida clique em **Next**.

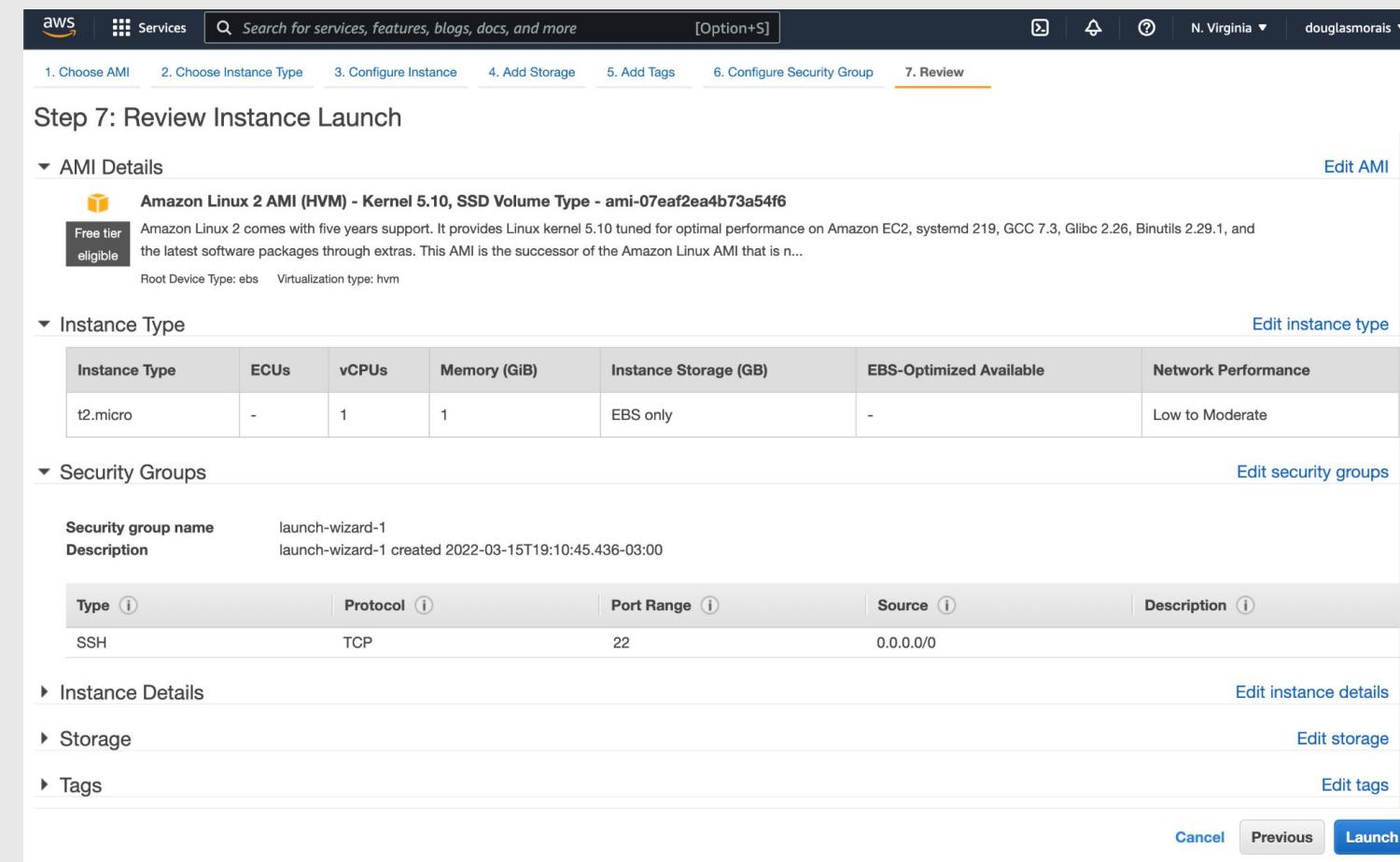
E por fim devemos confirmar as configurações de nossa as configurações selecionadas antes adicionar criar a nossa instância.

Para iniciar o nosso EC2 clique em **Launch**.

Em seguida precisamos criar uma chave para conectar em nossa instancia uma **Key Pair** no menu suspenso selecione **Create a new key pair** em seguida informe um nome para ela e clique em **Download Key Pair** e em seguida **Launch Instances**.

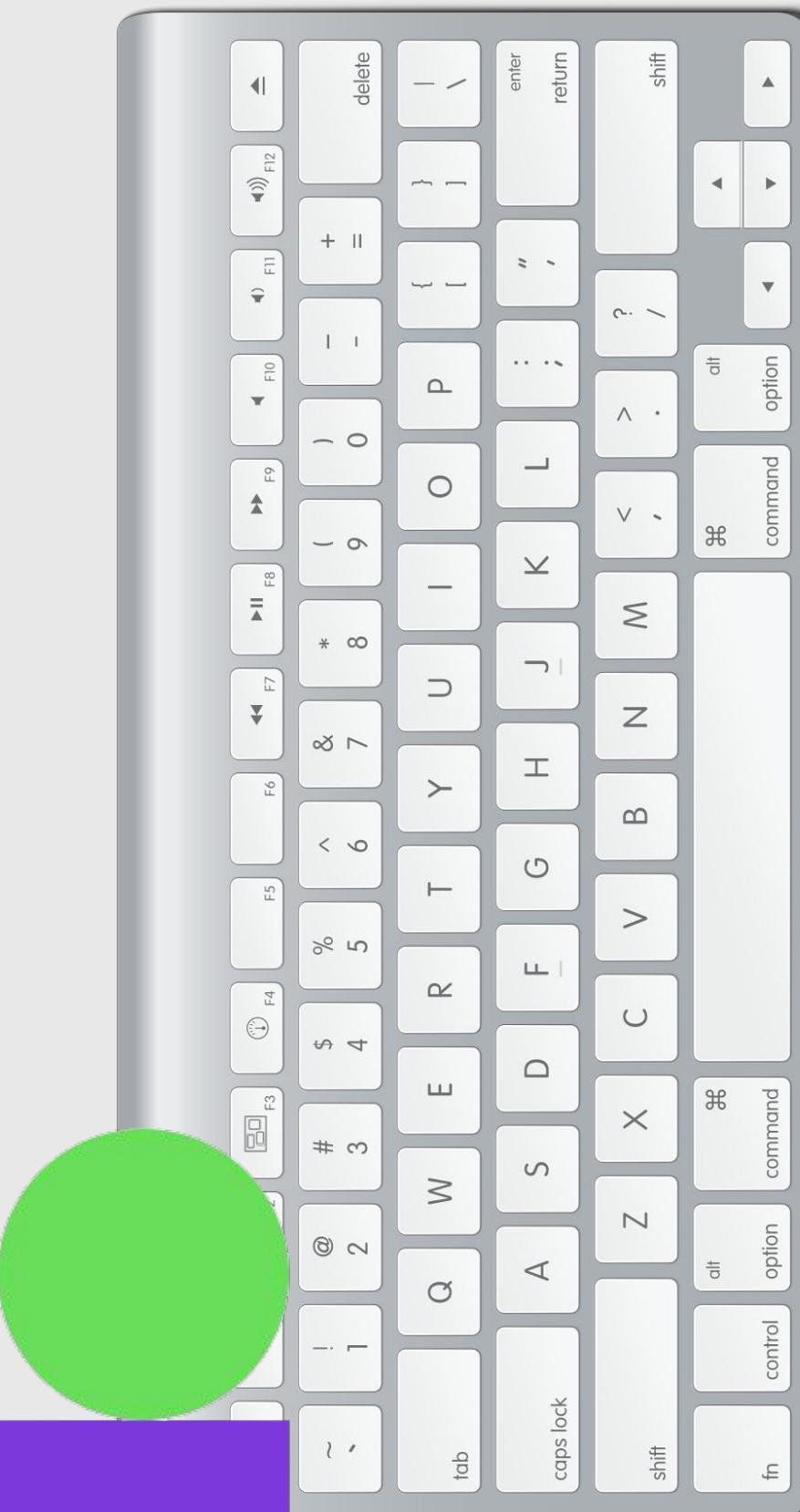


#PraCegoVer: Console AWS ec2 configure instances passo 6 adicionando grupos de segurança.



#PraCegoVer: Console AWS ec2 configure instances passo 7 visualizando as configurações de nossa instância.

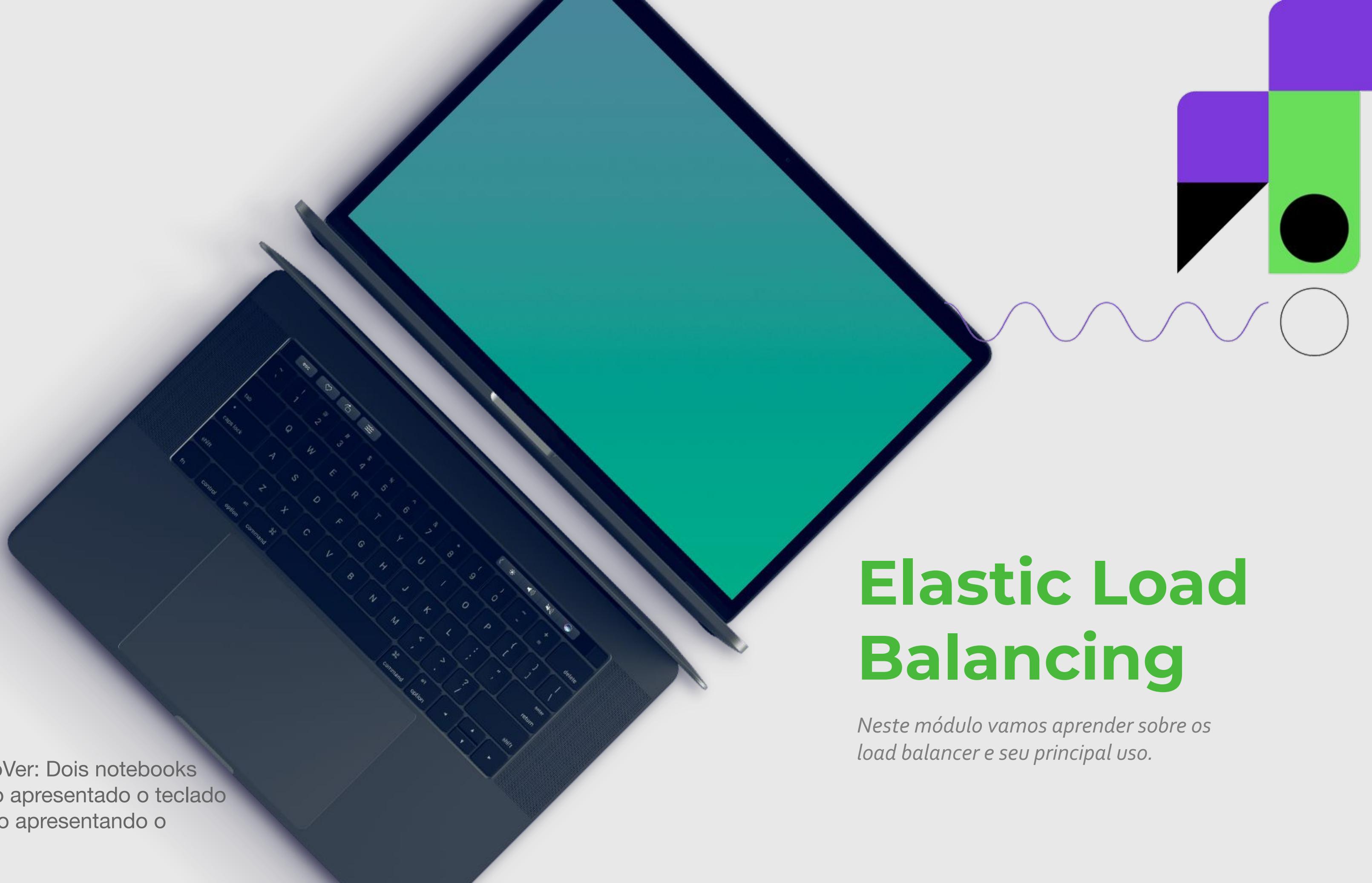
AWS EC2



Neste módulo aprendemos como criar instâncias EC2 via console AWS, lembrando que existem outras maneiras de iniciar o mesmo serviço, uma delas é via **AWS CLI** linha de comando.

Para os módulos sequenciais vamos aprender a publicar serviços dentro da instância e entender conceitos importantes relacionados a eles.

Segue material de apoio onde temos um passo a passo de como publicar uma api em **NODEJS** dentro da **AWS EC2** conteúdo do **Canal Sam Meech-Ward** clique [aqui](#) para visualizar.



Elastic Load Balancing

Neste módulo vamos aprender sobre os load balancer e seu principal uso.

#PraCegoVer: Dois notebooks
o primeiro apresentando o teclado
o segundo apresentando o
monitor.

Elastic Load Balancing

O que é Elastic Load Balancing?

Responsável por distribuir o tráfego de rede para melhorar escalabilidade e performance de sua aplicação.

Em outras palavras ele dimensiona de maneira autônoma as cargas de trabalhos dos servidores ou instâncias atuando na redução de latência (o tempo de resposta do servidor) por meio da rede virtual.



Criando um Load Balancing



Antes de qualquer coisa, o Load Balancing é uma feature tanto do **AWS EC2** quanto do **AWS LightSail** (Uma solução similar ao **EC2** porém com imagens prontas, como *wordpress*, *NODE JS* pré instalado e afins) as maiores diferenças entre as 2 soluções, o **EC2** te permite configurações mais avançadas enquanto o **LightSail** soluções de início rápido e contido.

Para iniciar o Load Balancer é necessário ter mais de uma instância EC2, por tanto para iniciar o nosso exercício vamos precisar ter ao menos mais um EC2 rodando em nosso console.

Por tanto, vamos lá.



AWS Lambda

*Poderosas e eficientes, vamos aprender
um sobre cloud functions.*

#PraCegoVer: Dois notebooks
o primeiro apresentando o teclado
o segundo apresentando o
monitor.

AWS Lambda

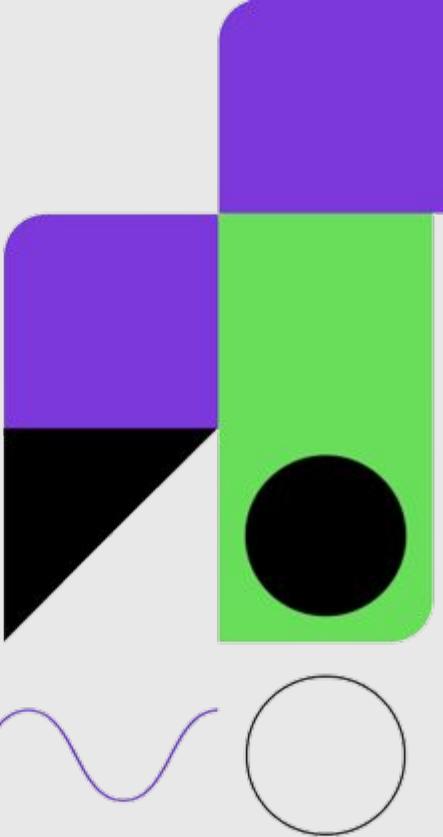
O que é uma Lambda?

Lambda é um serviço AWS Serverless (sem servidor) com foco em backend ou seja te permite criar funções ou API's específicas sem o provisionamento de servidores com um custo muito baixo.

São infinitas a aplicabilidade deste serviço, que pode ser desde um integração de serviços a um redimensionamento e gestão de imagens até serviços mais complexos com chamadas a API e resposta com diversas tratativas.



AWS Lambda



Para este exercício vamos publicar um projeto desenvolvido com **Serverless Framework** dentro de uma cloud function (lambda).

Quer saber mais sobre o Serverless Framework clique [aqui](#) e veja a documentação oficial.

E antes de qualquer coisa vamos precisar dos seguintes itens:

Clonar projeto que estou disponibilizando do **GitHub**, clique [aqui](#).

Instalar o **AWS CLI** selecione o seu sistema operacional na documentação, para acompanhar passo a passo da instalação clique [aqui](#).

*IMPORTANTE: Devemos ter o nosso [IAM](#) configurado.

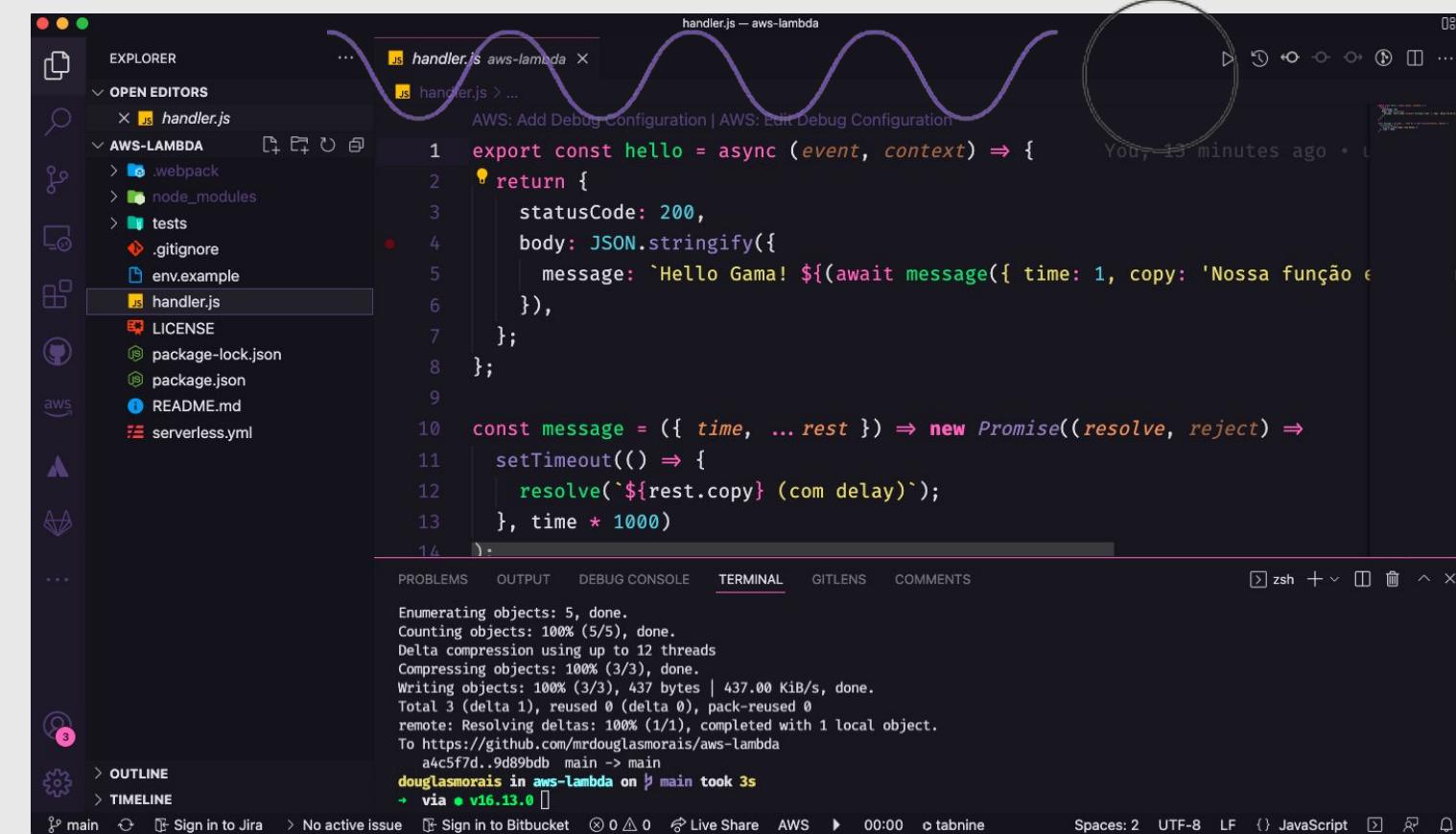
Iniciando uma Lambda

Diferente dos demais serviços todo o processo de criação de nossa lambda vai ser realizado por meio do nosso terminal (**AWS CLI**), criei um projeto em Node JS utilizando *JavaScipt* e o *Serverless Framework*.

Antes de mais nada precisamos vincular o nosso usuário ao **AWS CLI**.

```
douglasmorais in react-aws-s3 on main
→ via v16.13.0 aws configure
AWS Access Key ID [*****HUXI]:
AWS Secret Access Key [*****aKKh]:
Default region name [None]:
Default output format [None]:
```

a seguir informe o **Access key ID** de seu **IAM** e aperte enter, a mesma coisa para o seu **Secret access key** em seguida enter para as demais opções pode ser omitida apertando **Enter**.



#PraCegoVer: VSCode com projeto em serverless disponibilizando para criação deste projeto.

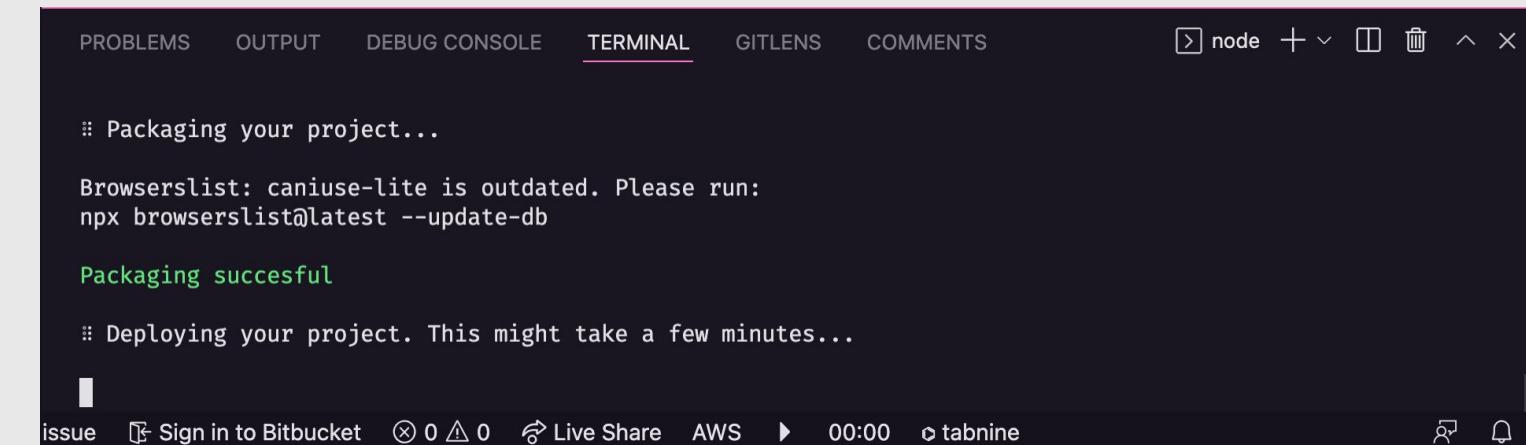
Iniciando uma Lambda

Para criar a nossa lambda tudo que precisamos fazer é abrir o nosso projeto e dentro do terminal instruir o comando deploy que está em nosso package.json

```
douglasmorais in aws-lambda on main took 26s
→ via v16.13.0 npm run deploy
```

Instruindo o comando automaticamente a **AWS CLI** fará a leitura do nosso projeto e iniciará o processo de

Ou seja assim que o projeto é disponibilizado e **AWS CLI** tem as devidas credenciais vinculadas toda a estrutura se encarrega do resto.



The screenshot shows a terminal window with the following output:

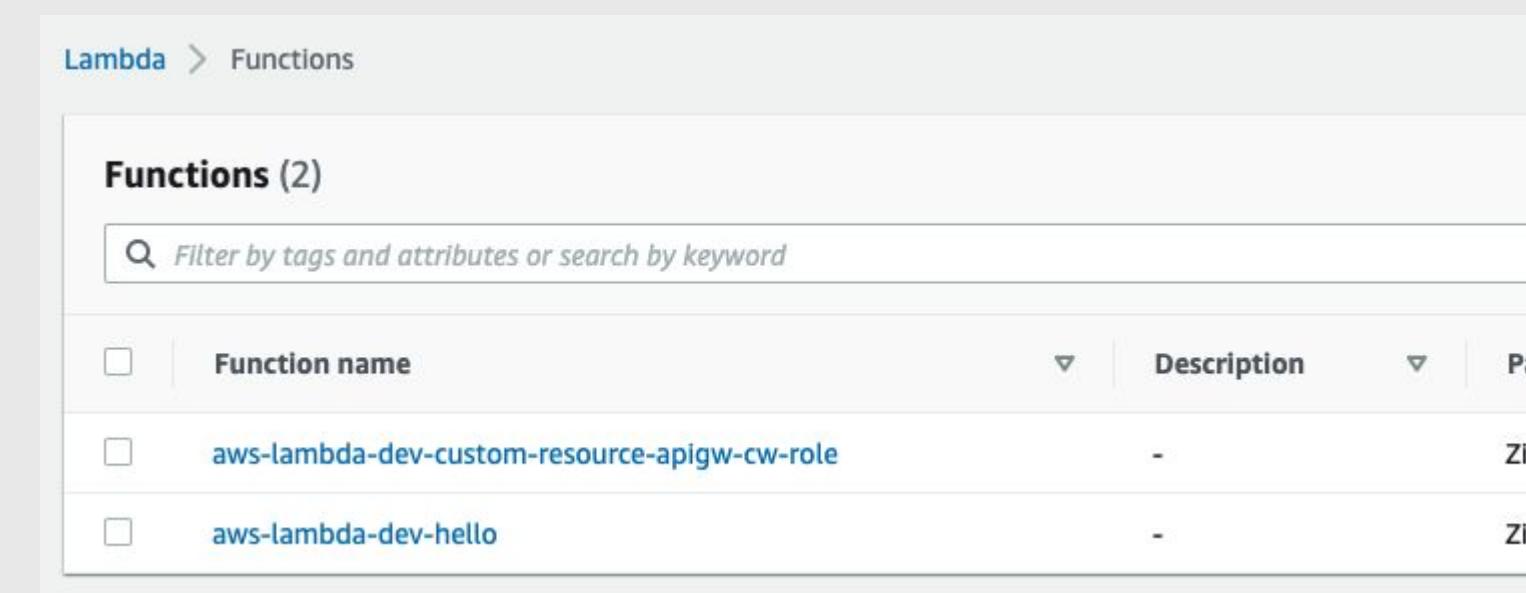
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS COMMENTS node + □ ×
```

```
:: Packaging your project...
Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Packaging succesful
:: Deploying your project. This might take a few minutes...
```

At the bottom, there is a status bar with icons for issue, sign in to Bitbucket, live share, AWS, and tabnine.

#PraCegoVer: Terminal com mensagem de serviço criado.



The screenshot shows the AWS Lambda console under the Functions section. It displays a table with the following data:

Function name	Description	Zip
aws-lambda-dev-custom-resource-apigw-cw-role	-	Zip
aws-lambda-dev-hello	-	Zip

#PraCegoVer: Console AWS Listando as lambdas ativas.

Iniciando uma Lambda



Agora nós temos a nossa função mediada por um serviço chamado **API Gateway** e nossos arquivos dentro de um bucket (**AWS S3**).

Basicamente neste caso **API Gateway** é responsável por criar uma URL para que a função seja acessada de maneira pública.

Este conjunto de soluções é altamente poderoso e pode ser utilizado em diversas soluções.

The screenshot shows the AWS Lambda function configuration page. The 'Configuration' tab is selected. On the left, there's a sidebar with 'General configuration' and 'Triggers' selected. Under 'Triggers', there's a search bar with 'API Gateway' and a list with one item: 'API Gateway: dev-aws-lambda'. Below it, the API endpoint is shown as <https://lhlmaistk.execute-api.us-east-1.amazonaws.com/dev/hello>. To the right, a browser window displays the response from this endpoint, which is a JSON object with a single key 'message' and the value 'Internal server error'.

#PraCegoVer: Console AWS Listando api GateWay URL para acesso a lambda e em seguida o resultado.



AWS S3

*Publicando conteúdos estáticos e
distribuição de*

#PraCegoVer: Dois notebooks
o primeiro apresentado o teclado
o segundo apresentando o
monitor.

AWS S3

O que é AWS S3 (Amazon Simple Storage Service)?

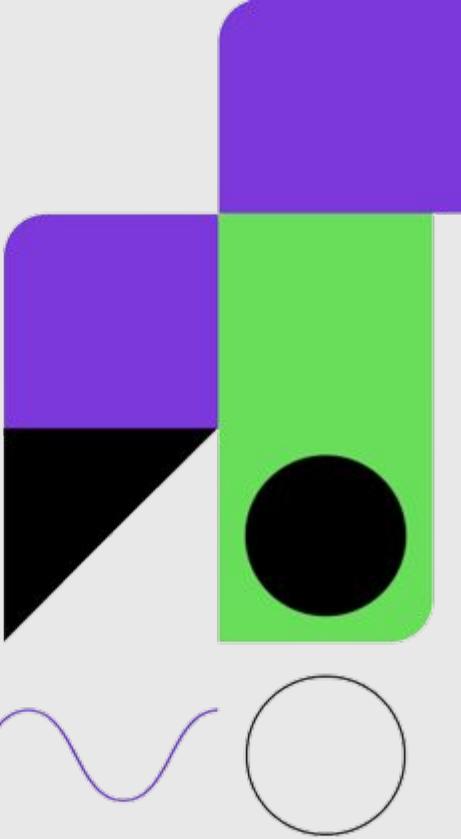
E um storage de armazenamento de objetos altamente escalável com uma gama de recursos muito completa.

É muito comum o seu uso associado a outros serviços como Lambda, onde os arquivos da função ficam armazenados dentro do bucket (S3).

É possível hospedar sites estáticos ou até mesmo como uma CDN para distribuir arquivos de imagem ou vídeo tanto para uma camada pública ou privada, além de uma de estar vinculado a uma série de soluções.



AWS S3



Para este exercício, vamos iniciar um bucket (S3) dentro do nosso console e hospedar uma aplicação em React (SPA) dentro desta solução no modelo de um site estático.

E antes de qualquer coisa vamos precisar dos seguintes itens:

Clonar projeto que estou disponibilizando do **GitHub**, clique [aqui](#).

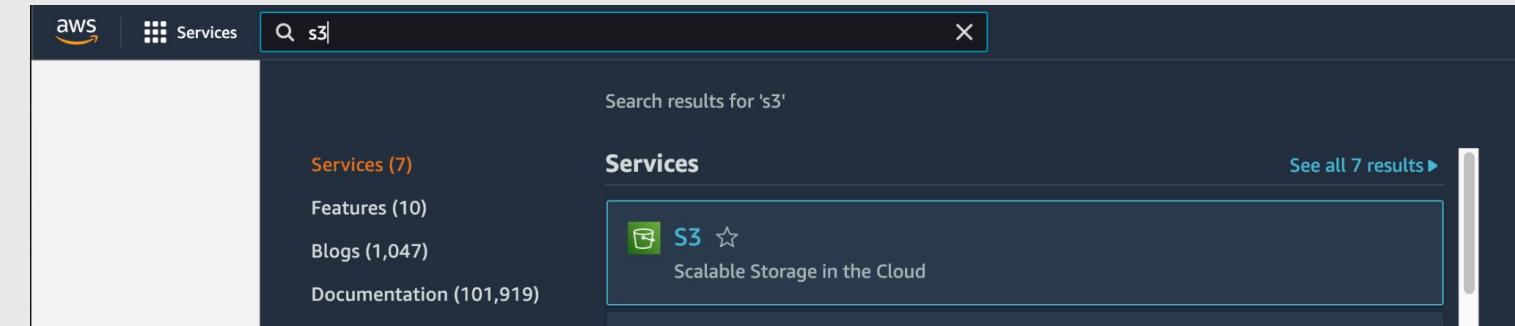
Instalar o **AWS CLI** selecione o seu sistema operacional na documentação, para acompanhar passo a passo da instalação clique [aqui](#).

*IMPORTANTE: Devemos ter o nosso [IAM](#) configurado.

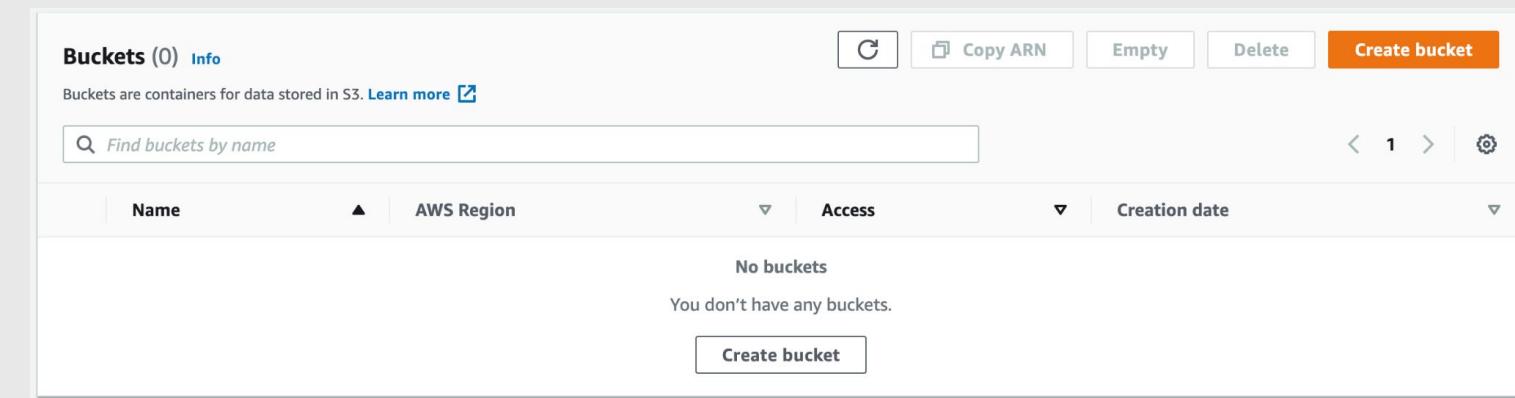
Criando um bucket.

Para iniciar devemos acessar o nosso console da Amazon e no mecanismo de pesquisa selecionar o serviço S3 conforme a imagem ao lado, em seguida selecione o serviço.

Assim que o painel mostrar o console do serviço, devemos criar o nosso bucket e para isso clique em **Create Bucket** e em seguida vamos iniciar as configurações necessárias para visualizar e tipo de serviço a ser publicado.



#PraCegoVer: Console AWS mecanismo de busca termo S3.



#PraCegoVer: Console AWS S3 página inicial dos serviços de S3.

Criando um bucket.

Em seguida vamos precisar iniciar as configurações básicas como nome do Bucket e região onde ele vai estar alocado.

O nome do bucket vou chamar *gama-s3* e a região alocada *Virginia us-east-1* (os custos de cada serviço pode variar de acordo com a região).

Em seguida vamos precisar desmarcar a opção de bloquear acesso público ao bucket (por padrão esta opção vem selecionada), em seguida marque o consentimento de conteúdo público **Turning off block all public access might result in this bucket and the objects within becoming public**, em seguida podemos manter as demais configurações e clicar em **Create Bucket**.

Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

#PraCegoVer: Console AWS S3 primeiro passo das configurações de nosso bucket.

Criando um bucket.

Assim o bucket for criado o mesmo vai estar listado em nosso console conforme imagem ao lado, agora vamos precisar concluir as configurações do nosso serviço.

Clique no bucket desejado, parte superior temos algumas abas com as configurações de nosso serviço vamo clicar em **Properties** em seguida ir até o box com texto **Static website hosting** ele vai estar marcado como *Disabled* clique em **Edit**, em seguida marque o *Static website hosting* como **Enabled** e informe em *Index document* o arquivo principal de leitura que em nosso caso é o *index.html*.

Buckets (1) Info			
Buckets are containers for data stored in S3. Learn more			
<input type="text"/> Find buckets by name			
Name	AWS Region	Access	Creation date
gama-s3	US East (N. Virginia) us-east-1	Objects can be public	March 16, 2022, 11:41:53 (UTC-03:00)

#PraCegoVer: Console AWS S3 listagem de buckets disponíveis em nosso painel.

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
 Disable Enable

Hosting type
 Host a static website Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object Redirect requests to another bucket or domain. [Learn more](#)

Index document
Specify the home or default page of the website.

Error document - optional
This is returned when an error occurs.

Redirection rules - optional

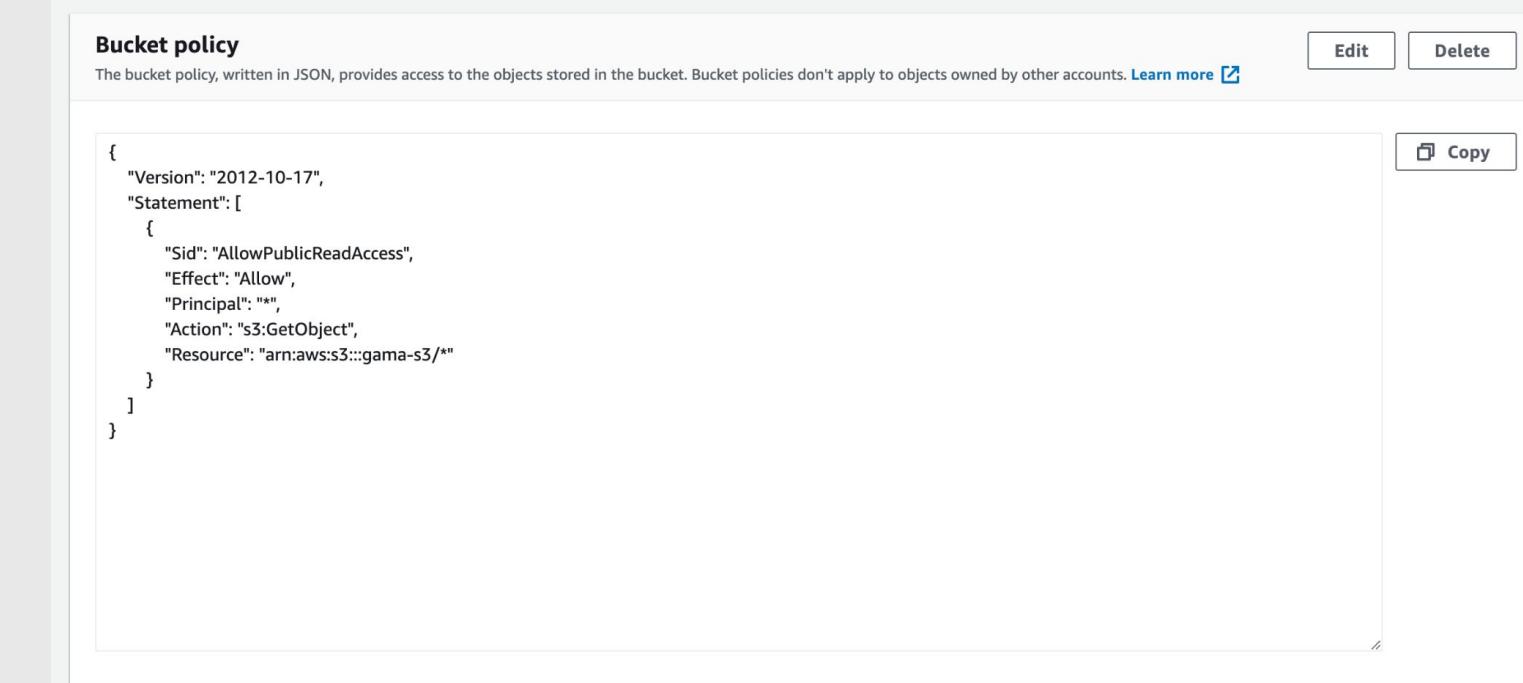
#PraCegoVer: Console AWS S3 configurações de propriedade de bucket.

Criando um bucket.

Em seguida vamos precisar configurar as políticas de bucket (Bucket policy) clique na parte superior na aba **Permissions** vá até o box **Bucket Policy** e clique em **Edit** a seguir devemos informar as políticas de uso do nosso bucket em formato JSON conforme o exemplo a seguir.

Atente-se a chave resource onde informamos o ARN de nosso bucket.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowPublicReadAccess",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": [ "s3:GetObject" ],  
      "Resource": [ "arn:aws:s3:::gama-s3/*" ]  
    }  
  ]  
}
```



#PraCegoVer: Console AWS S3 configurações de políticas de bucket.

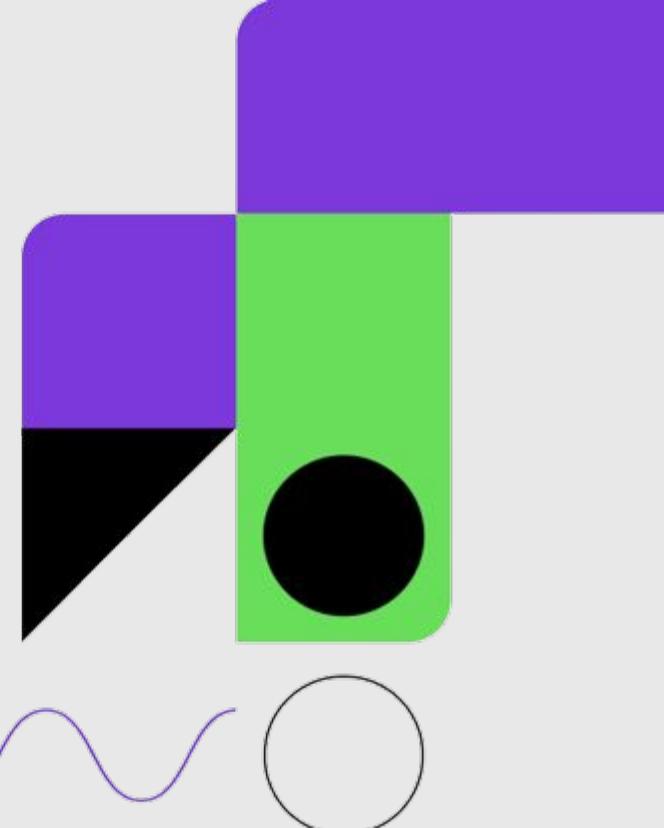
Criando um bucket.

Agora vamos precisar logar na amazon através do [AWS CLI](#) mencionado no início do nosso exercício e para logar em nosso console vamos precisar das nossas credenciais [IAM](#) conforme instruído em nossos capítulos anteriores.

Para iniciar em nosso **AWS CLI** devemos abrir o terminal após e instruir os seguinte comando.
aws configure.

```
douglasmorais in react-aws-s3 on main
→ via v16.13.0 aws configure
AWS Access Key ID [*****HUXI]:
AWS Secret Access Key [*****aKKh]:
Default region name [None]:
Default output format [None]:
```

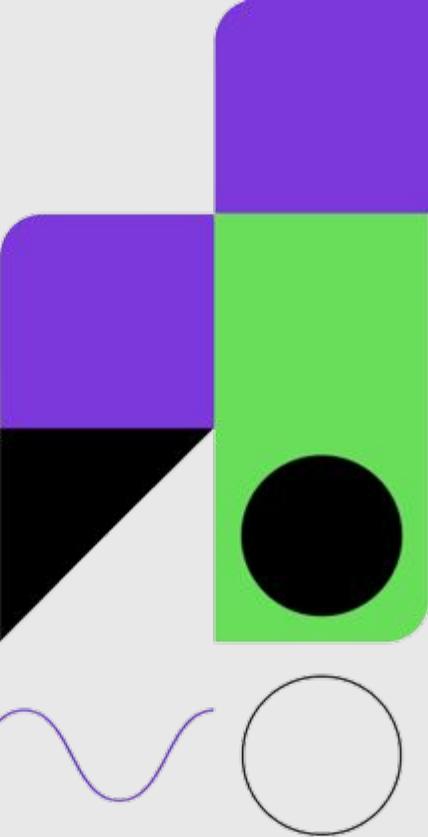
a seguir informe o **Access key ID** de seu **IAM** e aperte enter, a mesma coisa para o seu **Secret access key** em seguida enter para as demais opções pode ser omitida apertando **Enter**.



```
react-aws-s3 — douglasmorais@Douglass-MacBook-Pro —
douglasmorais in react-aws-s3 on ?main
→ via v16.13.0 aws configure
AWS Access Key ID [*****HUXI]: AKIA3XVLXY
AWS Secret Access Key [*****aKKh]: IC8sRr
NzRsPZ
Default region name [None]:
Default output format [None]:
douglasmorais in react-aws-s3 on ?main took 2m 33s
→ via v16.13.0
```

#PraCegoVer: Console AWS S3 configurações de políticas de bucket.

Criando um bucket.



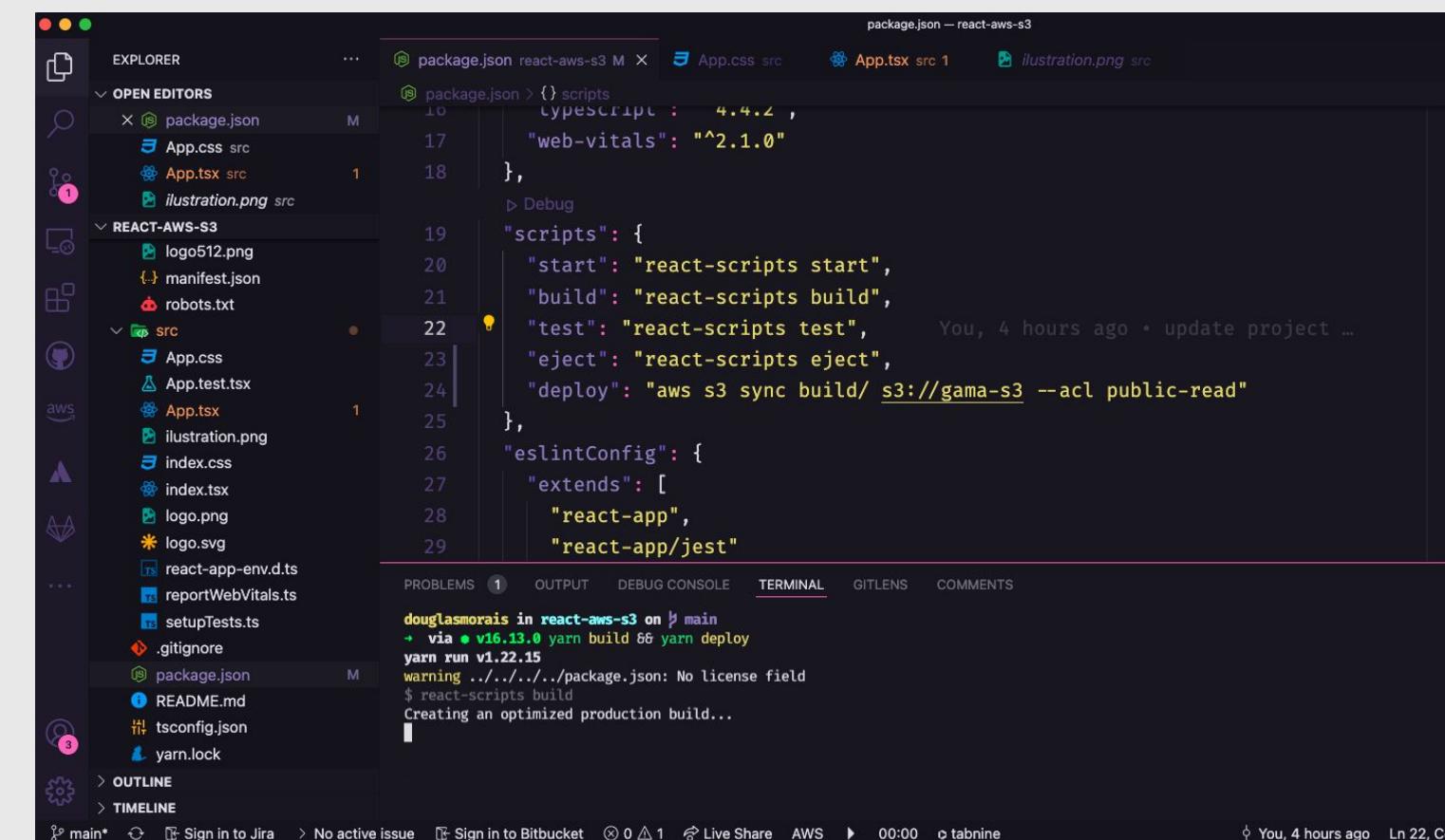
Por fim vamos acessar o projeto que disponibilizei no GitHub para criação do nosso bundle (o building de nosso projeto) e elaboração do script deploy.

Acessando o nosso projeto com o **VSCode** ou editor de sua preferência. Vamos precisar instruir um novo script dentro do nosso arquivo package.json.

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject",  
  "deploy": "aws s3 sync build/ s3://NOME-DO-SEU-BUCKET --acl  
public-read"  
}
```

Por fim instrua o seguinte comando no terminal

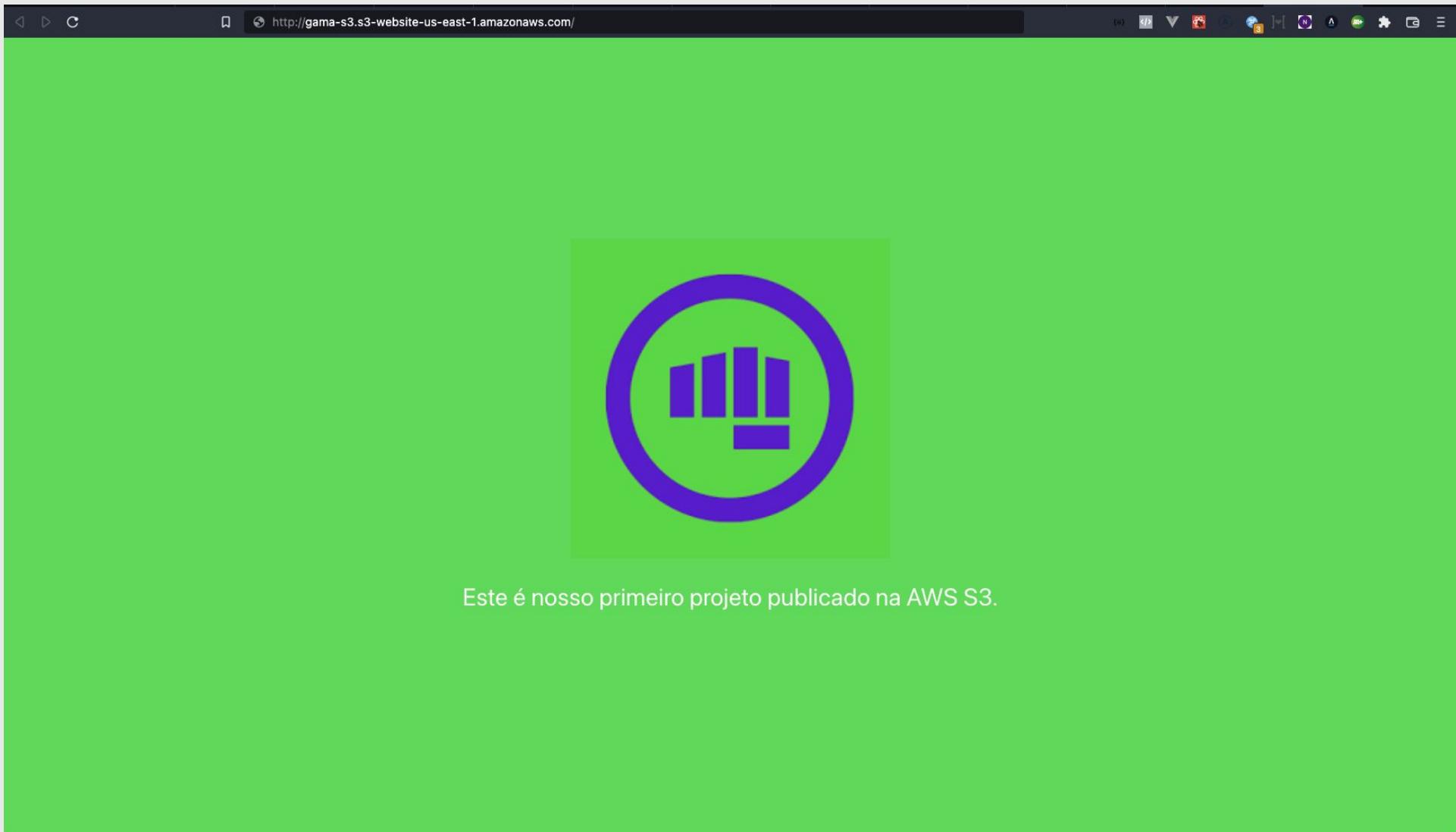
→ yarn build && yarn deploy



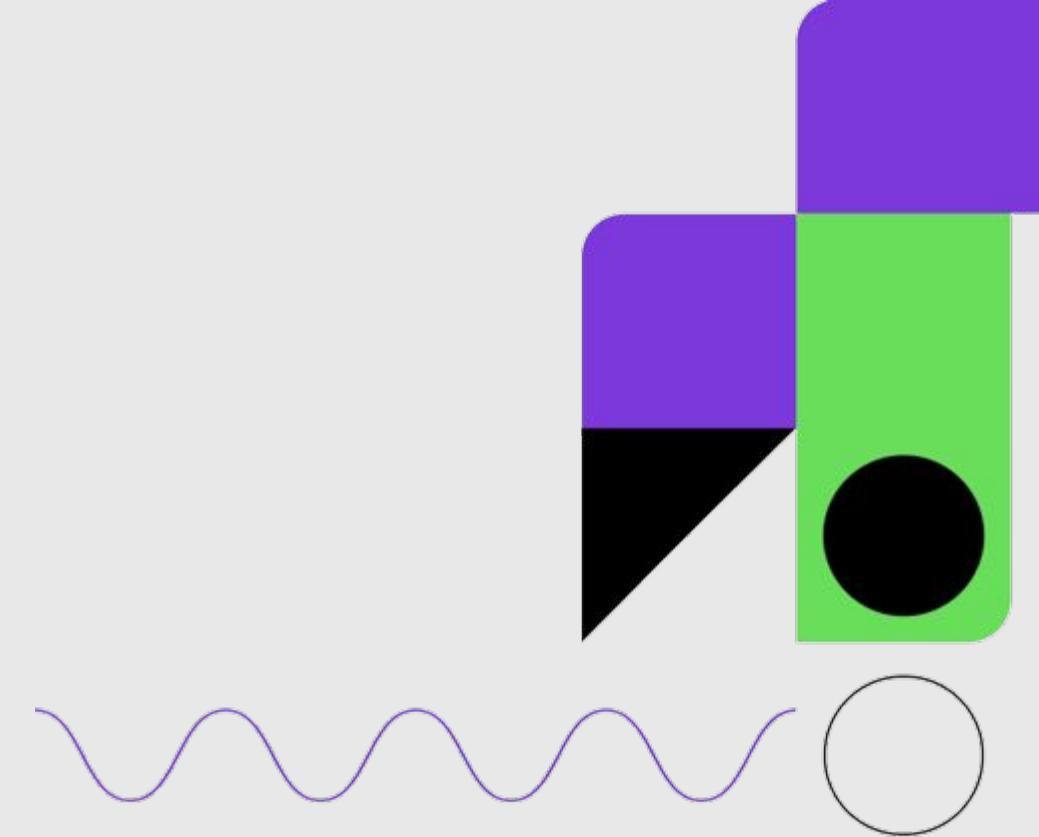
#PraCegoVer: VSCode com projeto em React aberto no arquivo package.json.

Resultado

E por fim, temos o nosso projeto publicado no novo bucket:



#PraCegoVer: Browser com nosso projeto publicado em nosso bucket (AWS S3)



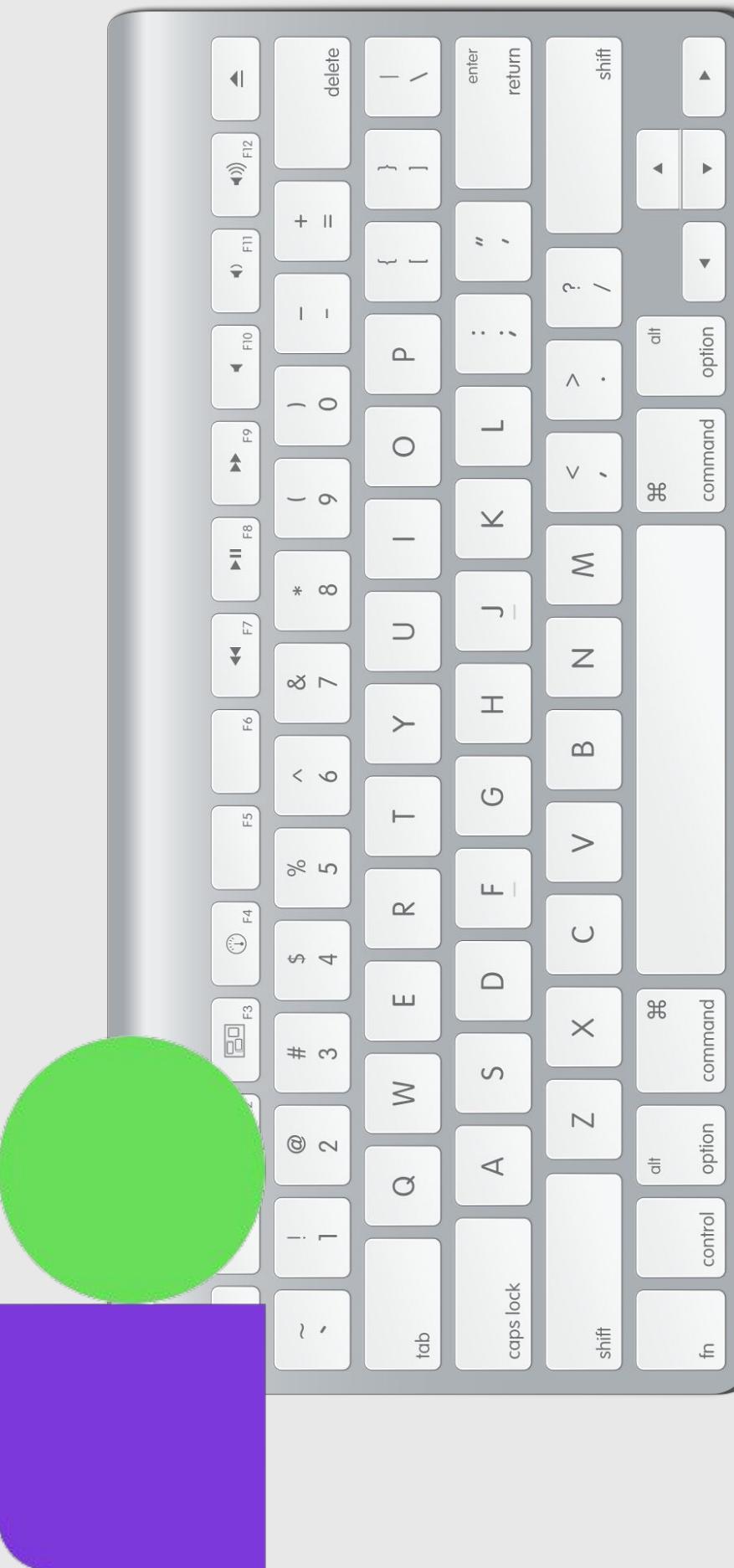
Caso ocorra algum erro em sua linha de comando, verifique o erro e confira se todos os passos foram implementados corretamente.

AWS S3

Neste módulo aprendemos a utilizar e configurar uma das muitas possibilidades de uso do **AWS S3** e tivemos o nosso primeiro contato com **AWS CLI** pontos importantes não mencionados durante o nosso exercício.

O AWS S3 é um serverless com custo muito baixo e para a publicação do mesmo em um domínio final ex: *myapp.io* é necessário integrar a outros serviços como **Route 53** ou até mesmo **CloudFront** (Gerencia e distribui o conteúdo do seu bucket através de CDN gerando um esquema de cachê). No entanto pode-se utilizar de outros serviços gratuitos para a publicação de sua aplicação.

Vale a pena dar aquela pesquisada no Google ;).





AWS RDS

Soluções em banco de dados escaláveis e completas com RDS.

#PraCegoVer: Dois notebooks
o primeiro apresentando o teclado
o segundo apresentando o
monitor.

AWS RDS

O que é RDS?

Amazon RDS (Relational Database Service) é um serviço que facilita a configuração e opera escalabilidade de dados relacionais na nuvem de maneira simples.

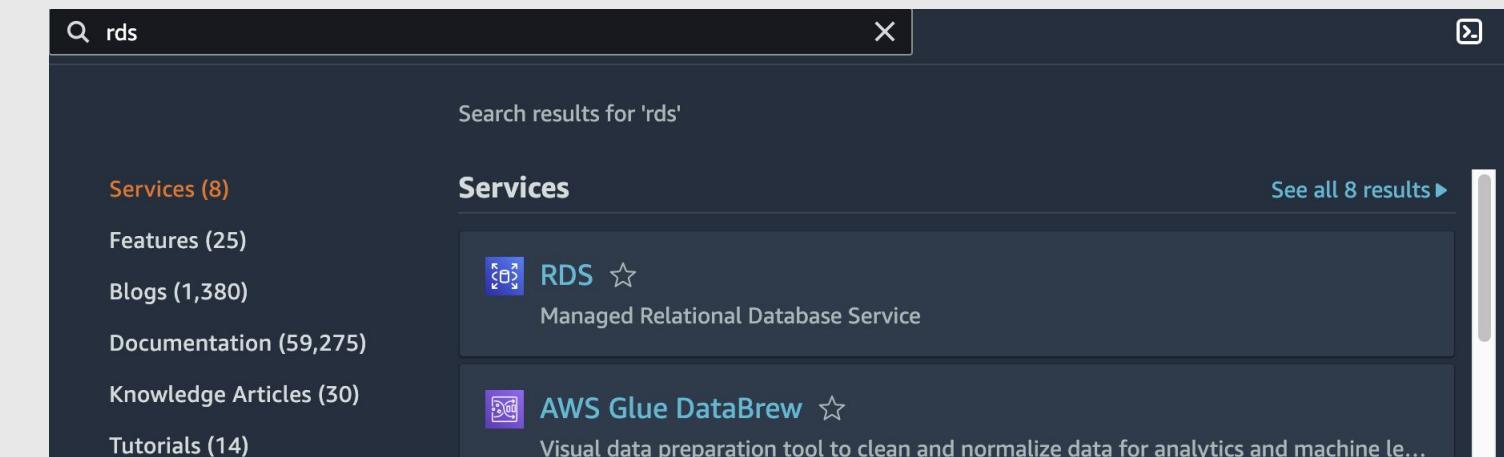
Fornece soluções de uso mais comum no mercado, como bancos: SQL Server, MySQL, PostgreSQL, Oracle e até mesmo soluções exclusivas da própria AWS como o Amazon Aurora disponíveis nos dialetos MySQL e PostgreSQL.



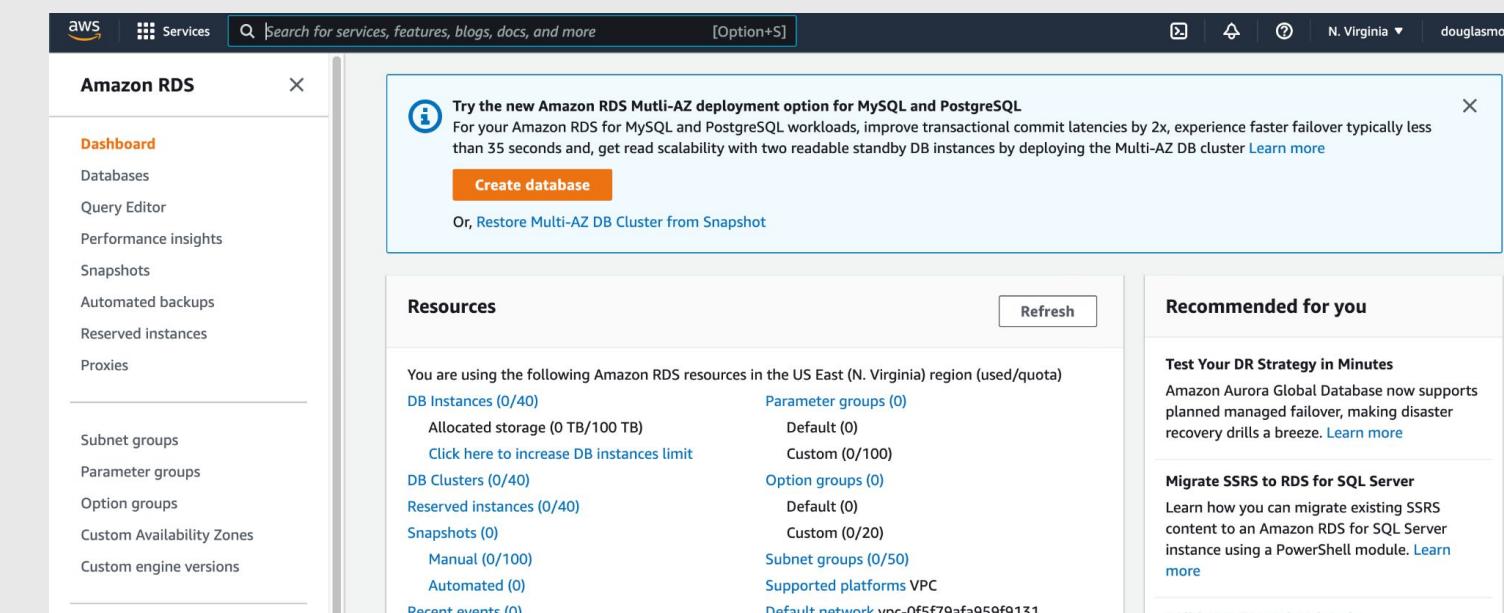
Criando um banco de dados RDS

Para este módulo vamos focar apenas na criação do nosso banco de dados, toda estrutura e querys você pode explorar nos módulos focados em dados.

Em nosso console **AWS** vamos pesquisar por **RDS**, primeira coisa que devemos fazer é criar um banco de dados, clique em **Create database**.



#PraCegoVer: Console AWS mecanismo de busca termo rds.



#PraCegoVer: Console AWS S3 página inicial dos serviços de RDS.

Criando um banco de dados RDS

A seguir podemos selecionar o dialeto do banco desejado, lembrando que a solução **AWS RDS** trata apenas de bancos relacionais.

Para este exercício vamos selecionar o **Amazon Aurora**, rolando a página do nosso console, podemos selecionar o Dialetos do nosso Banco de Dados (**Amazon Aurora** é uma solução otimizada da Amazon e não um dialeto), vamos selecionar o banco PostgreSQL.

The screenshot shows the 'Create database' page in the AWS RDS console. At the top, it says 'RDS > Create database'. Below that is the heading 'Create database'. Underneath, there's a section titled 'Choose a database creation method' with two options: 'Standard create' (selected) and 'Easy create'. The 'Standard create' option is described as allowing the user to set all configuration options, including ones for availability, security, backups, and maintenance. The 'Easy create' option is described as using recommended best-practice configurations, with some configuration options being changeable after the database is created. Below this, there's a section titled 'Engine options' with a heading 'Engine type'. It lists six database engines with their respective icons: Amazon Aurora (selected), MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. The Amazon Aurora icon features a cylinder with a starburst, while the others have their respective logos.

#PraCegoVer: Console AWS S3 página inicial dos serviços de RDS seleção de engine de Banco de Dados.

Criando um banco de dados RDS

A seguir temos algumas possibilidade de escolher o formato do nosso banco, Provisionando (*um banco provisionado é um servidor reservado e dedicado para a sua solução*) ou Serverless (*é uma solução de baixo custo porém com tempo de latência maior que recurso dedicado*). em nosso caso vamos selecionar o banco provisionado.

A seguir podemos selecionar um template para o nosso banco, ou seja um banco para ambiente de desenvolvimento (*ambiente de desenvolvimento não tem necessidade de performance alta*) ou um banco para ambiente produtivo (*Ambiente o cliente de fato acessa as informações os recursos contam otimizações de performance aprimoradas*).

The screenshot shows the 'Edition' section with 'Amazon Aurora PostgreSQL-Compatible Edition' selected. It also shows the 'Capacity type' section with 'Provisioned' selected, indicating managed server instance sizes. Below these, the 'Engine version' section lists 'Aurora PostgreSQL (Compatible with PostgreSQL 13.4)'.

Edition

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition

Capacity type [Info](#)

- Provisioned**
You provision and manage the server instance sizes.
- Serverless**
You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

Engine version [Info](#)
View the engine versions that support the following database features.
▶ [Show filters](#)

Available versions (21/21) [Info](#)

Aurora PostgreSQL (Compatible with PostgreSQL 13.4)

Templates
Choose a sample template to meet your use case.

Production
Use defaults for high availability and fast, consistent performance.

Dev/Test
This instance is intended for development use outside of a production environment.

Settings

#PraCegoVer: Console AWS S3 página inicial dos serviços de RDS seleção de tipo e configurações de serviços.

Criando um banco de dados RDS

Na sequência temos configurações como nome, senhas de acesso e afins.

Vamos selecionar uma senha gerada pela própria **AWS** e o nome do nosso banco vai ser *gamadb*.

Nos passos seguintes precisamos selecionar o provisionamento de recursos do nosso banco ou seja teremos nosso banco funcionando em cima de uma instância vamos manter selecionada a opção padrão.

Settings

DB cluster identifier [Info](#)
Type a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter.
 Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password [Info](#)

DB instance class

#PraCegoVer: Console AWS S3 página inicial dos serviços de RDS seleção de tipo e configurações de serviços.

Criando um banco de dados RDS

A seguir precisamos selecionar as opções de Viabilidade e durabilidade, neste caso manteremos as opções padrão selecionadas, para VPC as configurações padrão e para acesso ao nosso banco vamos tornar público.

Temos a opção de manter ele acessado apenas por nossos serviços como protocolos de segurança.

Nas configurações de VPC vamos manter padrão e database authentication também.

E por fim clique em **Create database**.

The screenshot shows the 'Availability & durability' section with the 'Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)' option selected. In the 'Connectivity' section, the 'Default VPC (vpc-0f5f79afa959f9131)' is chosen. A note states: 'After a database is created, you can't change its VPC.' In the 'Subnet group' section, 'default' is selected. Under 'Public access', the 'No' option is chosen, with a note: 'RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.'

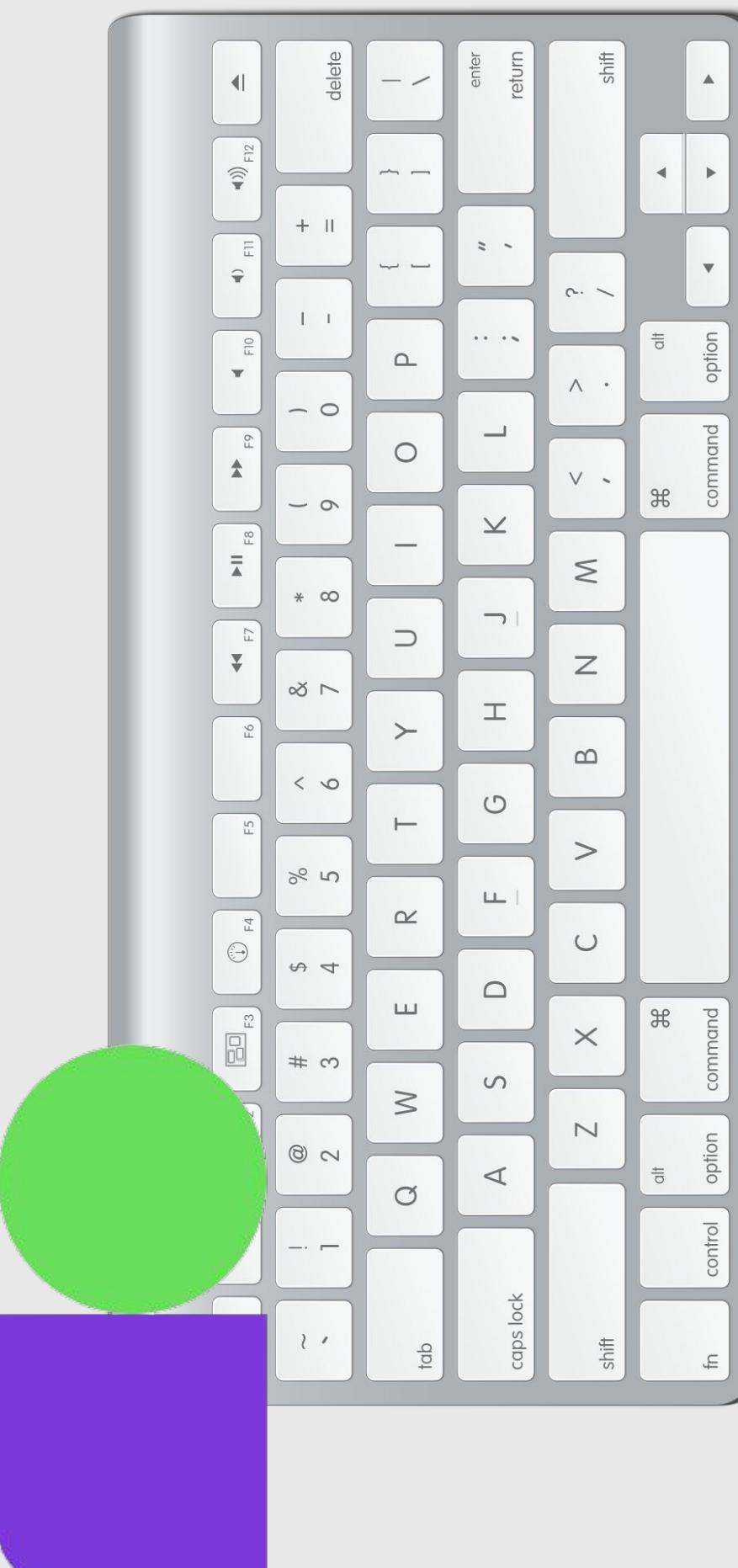
#PraCegoVer: Console AWS S3 página inicial dos serviços de RDS seleção de tipo e configurações de serviços.

AWS RDS

Aprendemos a criar nosso primeiro banco de dados fazendo uso da **Amazon RDS**, esta solução nos permite criar e gerenciar banco de dados de maneira escalável e completa.

Aproveite este aprendizado para se conectar ao banco e iniciar suas queries ou até mesmo começar a popular o banco.

Nos módulos a seguir você vai aprender a criar serviços de **API's** estruturar e modelar dados e até mesmo atuar com devops, por tanto adicione todos estes conhecimentos a soluções **AWS** para colocar em prática todo seu aprendizado e não se esqueça de tomar cuidado com o tamanho dos serviços selecionados procure por soluções free tier (camada gratuita).





AWS EKS

Kubernetes auto scaling com com EKS.

#PraCegoVer: Dois notebooks
o primeiro apresentando o teclado
o segundo apresentando o
monitor.

AWS EKS

O que é EKS (Elastic Kubernetes Service)?

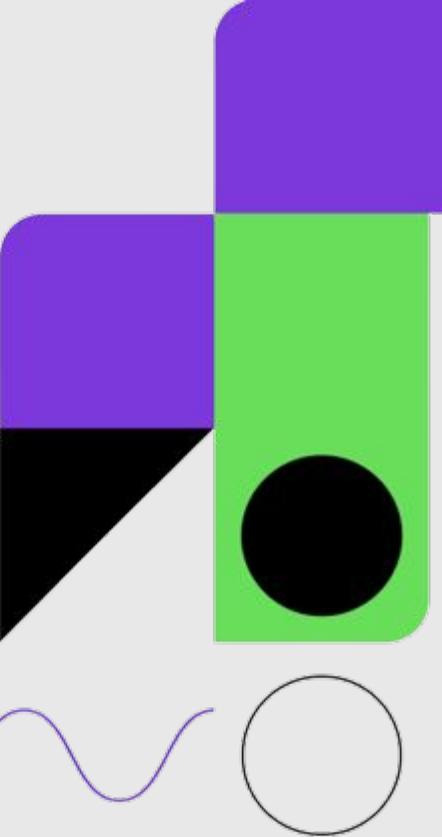
O Amazon Elastic Kubernetes (Amazon EKS) é um serviço gerenciado que é utilizado com o Kubernetes e se você ainda não sabe o que é o Kubernetes, clique [aqui](#) para ver mais sobre esta tecnologia.

Quando é utilizado este tipo de solução?

Em aplicações grandes que necessitam de ser auto gerenciadas aumentando ou diminuindo a capacidade computacional de acordo com a utilização, reduzindo riscos e custos operacionais.



AWS EKS



Antes de qualquer coisa vamos precisar ter uma compreensão sobre kubernetes e seus superpoderes para uma compreensão mais ampla sobre esta solução open source clique [aqui](#) para ler a documentação oficial.

E antes de qualquer coisa vamos precisar dos seguintes itens:

Ter instalado em seu equipamento o **Kubernetes** clique [aqui](#) para um passo a passo.

Clonar projeto que estou disponibilizando do **GitHub**, clique [aqui](#).

Instalar o **AWS CLI** selecione o seu sistema operacional na documentação, para acompanhar passo a passo da instalação clique [aqui](#).

*IMPORTANTE: Devemos ter o nosso [IAM](#) configurado.

CONSIDERAÇÕES FINAIS

Sobre o nosso aprendizado.

Aprendemos aqui a atuar com algumas das soluções da AWS, lembrando que existem muitas possibilidades em torno destas soluções e que nosso aprendizado foi uma degustação ou um convite para expandir o seu conhecimento.

Para saber mais sobre as soluções **AWS** em breve disponibilizaremos um conteúdo muito massa com foco nos **AWS** produtos criando soluções para dores reais.

Siga-nos no Linkedin.



Fechamento

Sobre o nosso aprendizado.

Aprendemos aqui a atuar com algumas das soluções da AWS, lembrando que existem muitas possibilidades em torno destas soluções e que nosso aprendizado foi uma degustação ou um convite para expandir o seu conhecimento.

Para saber mais sobre as soluções **AWS** em breve disponibilizaremos um conteúdo muito massa com foco nos **AWS** produtos criando soluções para dores reais.

Siga-nos no Linkedin.



E os estudos não param por aqui.

Aproveite ao máximo para praticar lembrando que...

"a repetição sem exaustão leva a perfeição" - Autor desconhecido.

Prof Douglas Moraes

Referência Bibliográfica

AMAZON AWS documentação oficial.

WikiPedia.

Serverless Framework documentação oficial.

Kubernetes documentação oficial.

