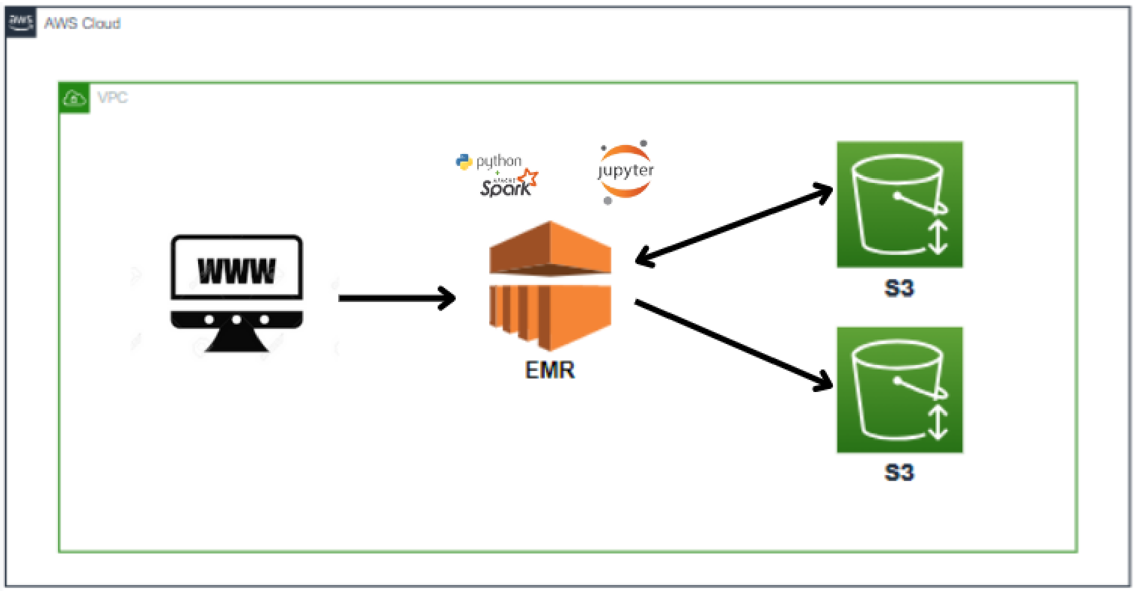
Creation of ETL Pipeline in EMR clusters using PySpark, Jupyter Notebook and S3 Buckets

1.Objective

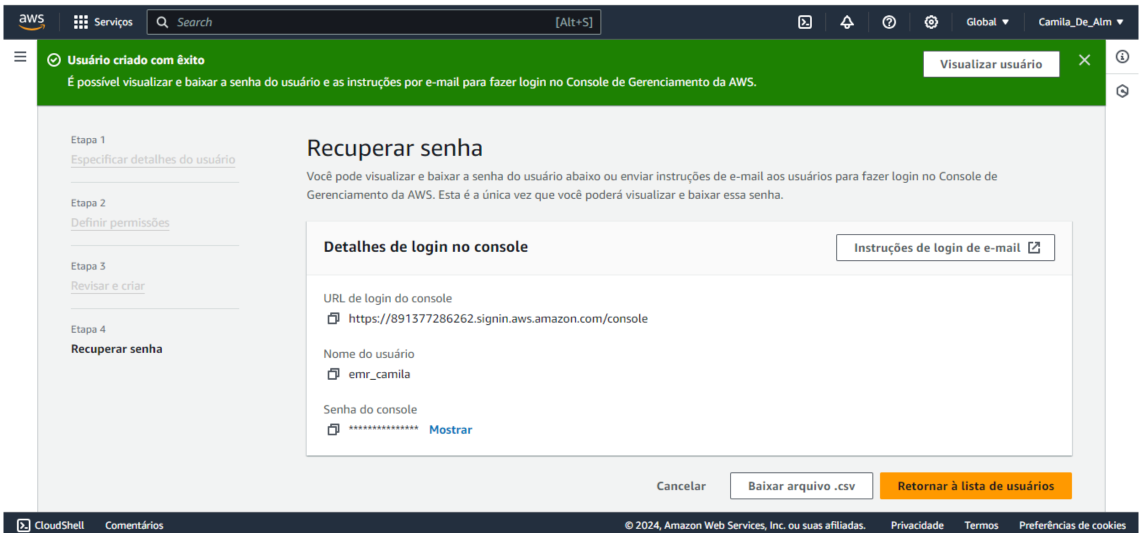
In this project I built a data pipeline in the AWS cloud, from scratch, including environment configurations, creation of users, networks, subnets, EC2, security groups, EMR clusters, creation of S3 buckets, transformation, and cleaning of raw data, using PySpark, until the database containing ready-to-use data is made available to the end user. The data source is the website of Redfin, one of the largest real estate agencies in the US and Canada, which freely makes real estate market data available for metropolitan areas, cities, neighborhoods and zip codes. Below are the step-by-step instructions for each step.

Demonstration of the final architecture flowchart on AWS:



2.Creation of the Administrator user

You will need to have an active AWS – Amazon World Services account to perform the following items. Due to security reasons, the root user, who is the user responsible for creating the AWS account, must create another administrator user so that in the event of an invasion of the account, it remains secure. For this project, the administrator user was created in AWS IAM (Identity and Access Management) which is an AWS service that helps you control access to AWS resources securely. Logged in as the root user, I created the credentials and keypairs for the administrator user, and enabled the necessary permissions to use the EMR and S3 Buckets.


Usuário criado com êxito
É possível visualizar e baixar a senha do usuário e as instruções por e-mail para fazer login no Console de Gerenciamento da AWS.

Etapa 1
Especificar detalhes do usuário
Etapa 2
Definir permissões
Etapa 3
Revisar e criar
Etapa 4
Recuperar senha

Recuperar senha
Você pode visualizar e baixar a senha do usuário abaixo ou enviar instruções de e-mail aos usuários para fazer login no Console de Gerenciamento da AWS. Esta é a única vez que você poderá visualizar e baixar essa senha.

Detalhes de login no console

URL de login do console
https://891377286262.signin.aws.amazon.com/console

Nome do usuário
emr_camila

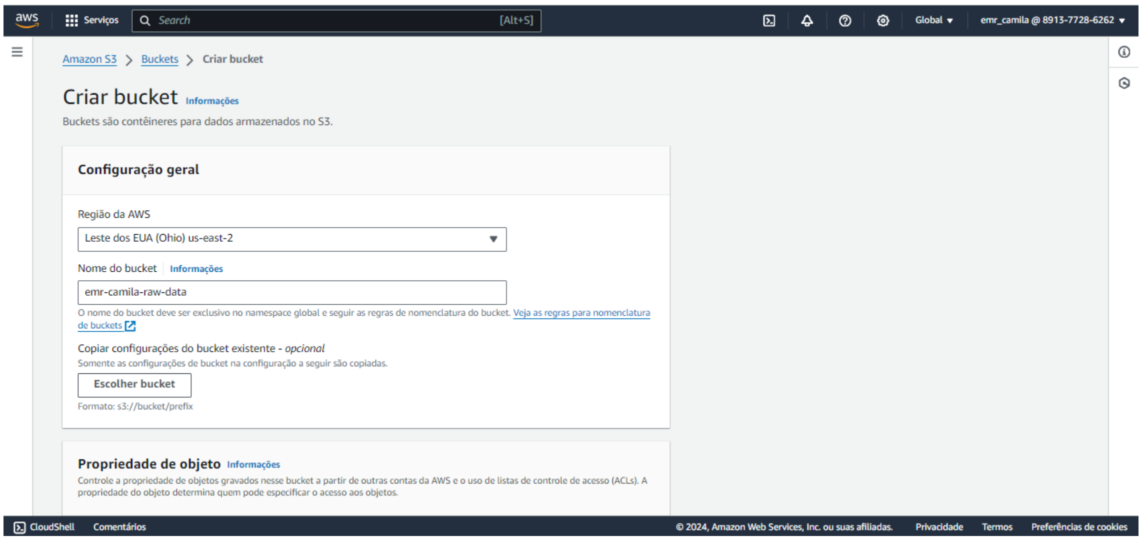
Senha do console
*************** Mostrar

© 2024, Amazon Web Services, Inc. ou suas afiliadas.

For security reasons, I also downloaded the credentials in csv, so I could recover them in the future, in case of lost or forgotten password. After following the security practices recommended by AWS, I logged out of the Root user and logged in with the new Administrator user.

3. Creating Buckets in AWS S3

Using the Administrator user, I created two S3 buckets, the first to receive the data extraction from the source located on the Redfin website and the second bucket to receive the transformed and cleaned data. In the AWS console, search for S3, then create the bucket. The bucket name must be globally unique and follow the naming rules determined by AWS, as shown in the image below.


Amazon S3 > Buckets > Criar bucket

Criar bucket
Buckets são contêineres para dados armazenados no S3.

Configuração geral

Região da AWS
Leste dos EUA (Ohio) us-east-2

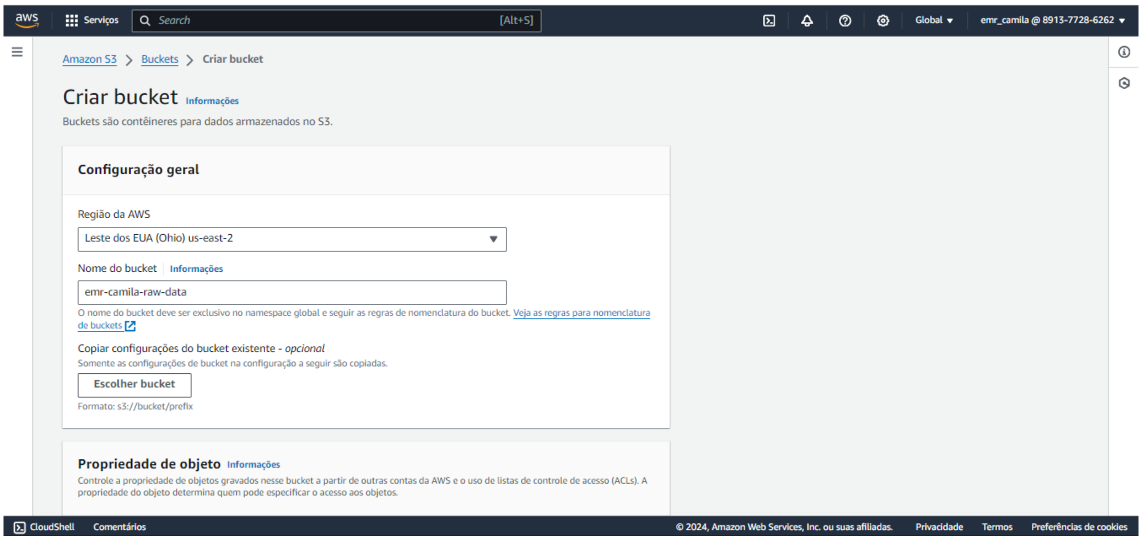
Nome do bucket
emr-camila-raw-data
O nome do bucket deve ser exclusivo no namespace global e seguir as regras de nomenclatura do bucket. Veja as regras para nomenclatura de buckets

Copiar configurações do bucket existente - *opcional*
Somente as configurações de bucket na configuração a seguir são copiadas.
Formato: s3://bucket/prefix

Propriedade de objeto
Controle a propriedade de objetos gravados nesse bucket a partir de outras contas da AWS e o uso de listas de controle de acesso (ACLs). A propriedade do objeto determina quem pode especificar o acesso aos objetos.

© 2024, Amazon Web Services, Inc. ou suas afiliadas.

The region chosen was us-east of the USA, due to the cost being the most recommended, but due to latency, it is recommended that the Region chosen to be the same as where your user is located. Additionally, it is possible to determine properties, tags, and versioning, but I kept the default options suggested by AWS. For the second bucket, I followed the same procedure as above.


aws
Serviços
Search [Alt+S]
Global
emr_camila @ 8913-7728-6262
Amazon S3 > Buckets > Criar bucket
CloudShell
Comentários
© 2024, Amazon Web Services, Inc. ou suas afiliadas.
Privacidade
Termos
Preferências de cookies

Criar bucket Informações

Buckets são contêineres para dados armazenados no S3.

Configuração geral

Região da AWS: Leste dos EUA (Ohio) us-east-2

Nome do bucket | Informações: emr-camila-raw-data

O nome do bucket deve ser exclusivo no namespace global e seguir as regras de nomenclatura do bucket. Veja as regras para nomenclatura de buckets

Copiar configurações do bucket existente - *opcional*

Somente as configurações de bucket na configuração a seguir são copiadas.

Escolher bucket

Formato: s3://bucket/prefix

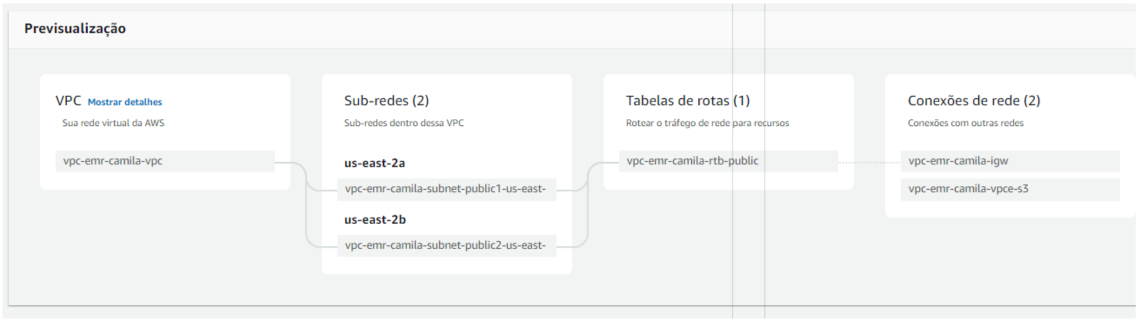
Propriedade de objeto Informações

Controle a propriedade de objetos gravados nesse bucket a partir de outras contas da AWS e o uso de listas de controle de acesso (ACLs). A propriedade do objeto determina quem pode especificar o acesso aos objetos.

4.Creation of VPC and Subnets

VPC is a Virtual Private Cloud, that is, a virtual private cloud, it creates an isolated virtual private network in AWS, protecting my EMR cluster and my data against unauthorized access and encrypts data at rest and in transit between Amazon S3 and Amazon EMR to ensure information confidentiality. In the console, search for VPC, then click “Create VPC”. I chose a name for the VPC, we must create the VPC, subnets, route tables and determine the availability zones, gateway and endpoint . I configured the availability zones as two, because if there is a disaster and one region becomes unavailable, data transit will be redirected to the second, I set the same value for public networks and none for private networks, because it will be internal data. The endpoint will be the S3 Gateway.

Next, visualization of the entire VPC flow.

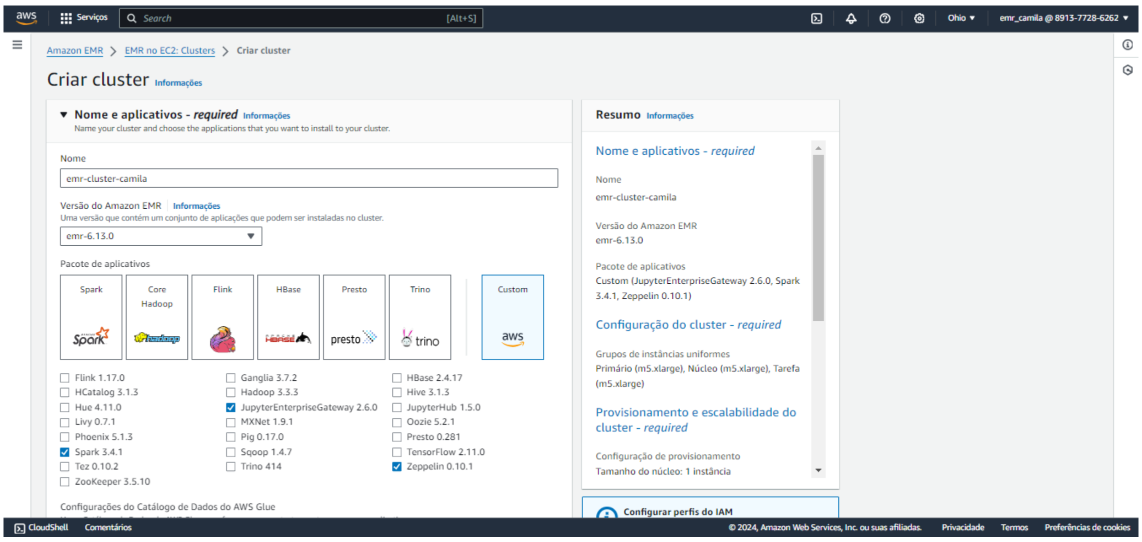

Previsualização
VPC Mostrar detalhes
Sua rede virtual da AWS
vpc-emr-camila-vpc
Sub-redes (2)
Sub-redes dentro dessa VPC
us-east-2a
vpc-emr-camila-subnet-public1-us-east-
us-east-2b
vpc-emr-camila-subnet-public2-us-east-
Tabelas de rotas (1)
Rotear o tráfego de rede para recursos
vpc-emr-camila-rtb-public
Conexões de rede (2)
Conexões com outras redes
vpc-emr-camila-igw
vpc-emr-camila-vpce-s3

5. EMR Clusters

After configuring the user, environment, and security, I created the AWS EMR (Elastic Map Reduce) cluster, which is an AWS service that facilitates the execution of big data frameworks, such as Apache Hadoop and Apache Spark, to process and analyze large volumes of data, in this project I used Apache Spark. In AWS EMR clusters, nodes are EC2 instances that come together to provide the processing and storage power needed to run your big data jobs. There are three main types of nodes in an EMR cluster:

- **Node or Primary Node:** It is responsible for coordinating all work in the cluster, managing tasks, monitoring the health of the cluster, and tracking the status of jobs.
- **Core Nodes:** They perform data processing tasks and there can be multiple core nodes to distribute and execute jobs in parallel.
- **Task Nodes (Optional):** Provide additional processing power to perform CPU-intensive tasks.

In this project, I created the Primary Node and the Secondary Node, due to the frequency and size of the load. The EMR already comes with main frameworks, so it can be configured directly in the console, which reduces the cost. The setup chosen was version 6.13.0, I didn't choose the most recent one, I selected the Spark, Jupyter and Zeppelin applications, as shown below.



Criar cluster Informações

Nome e aplicativos - *required* Informações
Name your cluster and choose the applications that you want to install to your cluster.

Nome: emr-cluster-camila

Versão do Amazon EMR Informações
Uma versão que contém um conjunto de aplicações que podem ser instaladas no cluster.
emr-6.13.0

Pacote de aplicativos: Custom

☐ Flink 1.17.0	☐ Ganglia 3.7.2	☐ HBase 2.4.17
☐ HCatalog 3.1.3	☐ Hadoop 3.3.3	☐ Hive 3.1.3
☐ Hue 4.11.0	✓ JupyterEnterpriseGateway 2.6.0	☐ JupyterHub 1.5.0
☐ Livy 0.7.1	☐ MXNet 1.9.1	☐ Oozie 5.2.1
☐ Phoenix 5.1.3	☐ Pig 0.17.0	☐ Presto 0.281
✓ Spark 3.4.1	☐ Sqoop 1.4.7	☐ TensorFlow 2.11.0
☐ Tez 0.10.2	☐ Trino 414	✓ Zeppelin 0.10.1
☐ ZooKeeper 3.5.10		

Configurações do Catálogo de Dados do AWS Glue

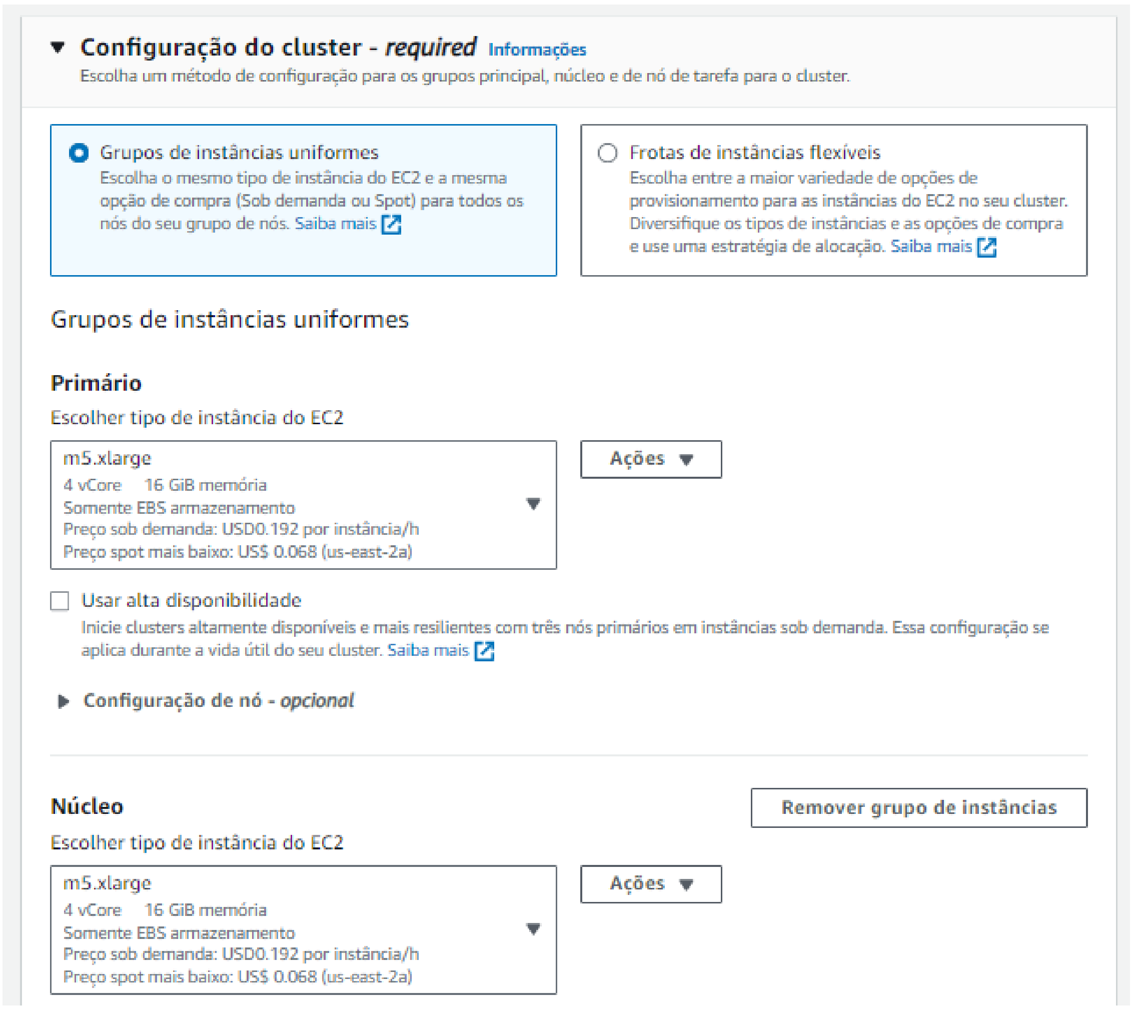
Resumo Informações

Nome e aplicativos - *required*
Nome
emr-cluster-camila
Versão do Amazon EMR
emr-6.13.0
Pacote de aplicativos
Custom (JupyterEnterpriseGateway 2.6.0, Spark 3.4.1, Zeppelin 0.10.1)

Configuração do cluster - *required*
Grupos de instâncias uniformes
Primário (m5.xlarge), Núcleo (m5.xlarge), Tarefa (m5.xlarge)

Provisionamento e escalabilidade do cluster - *required*
Configuração de provisionamento
Tamanho do núcleo: 1 instância

The group of instances chosen was uniform, to use AWS EC2 on demand or spot, due to economies of scale, the type chosen was M5.xlarge, which contains 4v core and 16Gib of memory, for the core I chose the same configurations and I removed the option to have a task cluster , because it is optional. Amazon EC2 M5 instances are powered by the cloud's fastest Intel Xeon Scalable processors with an all -core turbo frequency of up to 4.5 GHz. Additionally, M5zn instances feature 100 Gbps networking capacity, making them ideal for computing-intensive applications with high consumption of network resources, but the instance can be changed depending on the size of the load, for performance measures I chose the M5.Xlarge.



▼ **Configuração do cluster - *required*** Informações

Escolha um método de configuração para os grupos principal, núcleo e de nó de tarefa para o cluster.

[✓ Grupos de instâncias uniformes] Escolha o mesmo tipo de instância do EC2 e a mesma opção de compra (Sob demanda ou Spot) para todos os nós do seu grupo de nós. Saiba mais

Frotas de instâncias flexíveis — Escolha entre a maior variedade de opções de provisionamento para as instâncias do EC2 no seu cluster. Diversifique os tipos de instâncias e as opções de compra e use uma estratégia de alocação. Saiba mais

Grupos de instâncias uniformes

Primário

Escolher tipo de instância do EC2

m5.xlarge
4 vCore 16 GiB memória
Somente EBS armazenamento
Preço sob demanda: USD0.192 por instância/h
Preço spot mais baixo: US$ 0.068 (us-east-2a)

☐ Usar alta disponibilidade
Inicie clusters altamente disponíveis e mais resilientes com três nós primários em instâncias sob demanda. Essa configuração se aplica durante a vida útil do seu cluster. Saiba mais

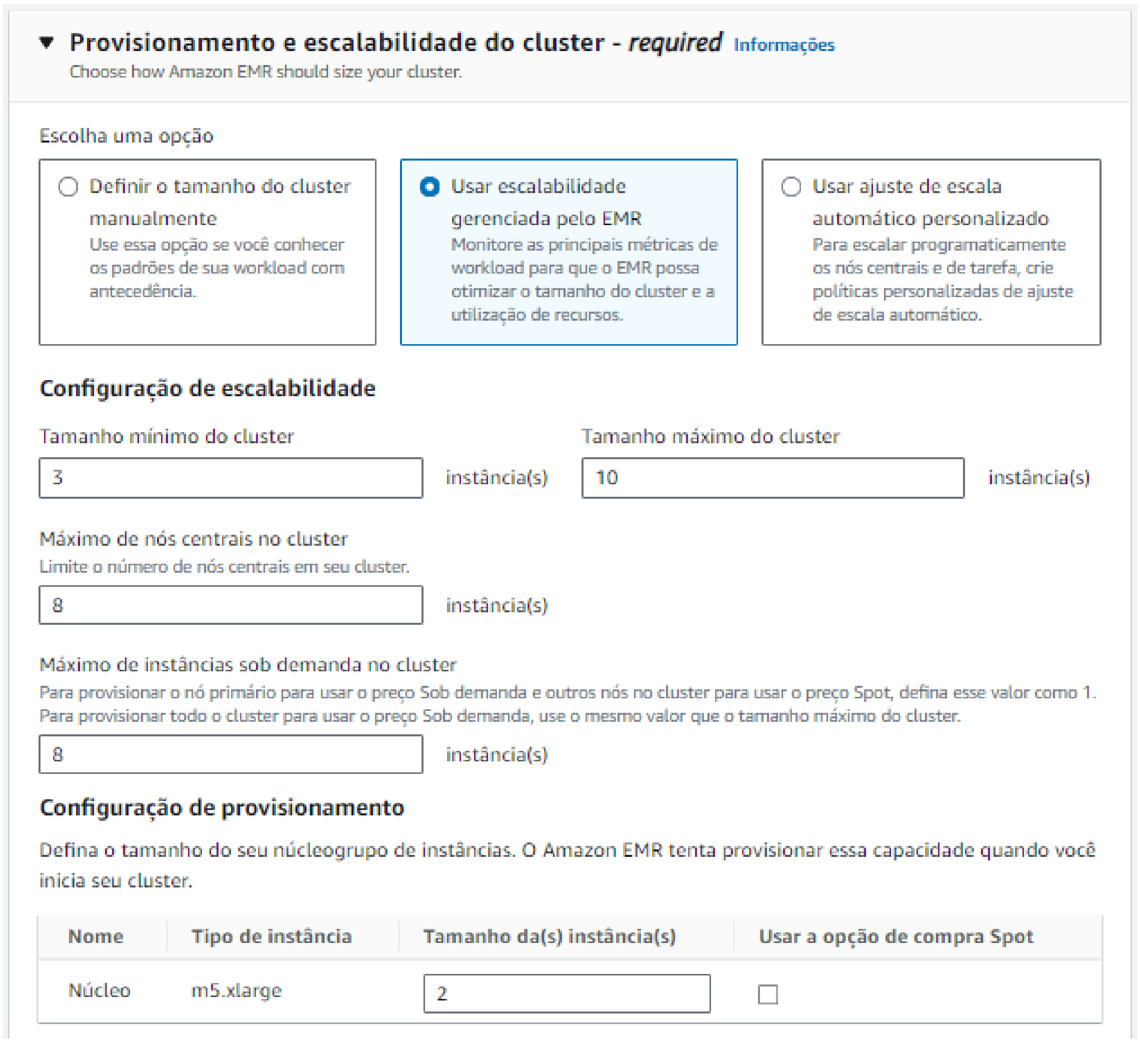
▶ Configuração de nó - *opcional*

Núcleo

Escolher tipo de instância do EC2

m5.xlarge
4 vCore 16 GiB memória
Somente EBS armazenamento
Preço sob demanda: USD0.192 por instância/h
Preço spot mais baixo: US$ 0.068 (us-east-2a)

For the “Cluster provisioning and scalability” section, regarding scalability I chose the option managed by EMR, because this way the EMR adapts according to the variation in the data flow. The minimum cluster size (number of instances) was three, because one is for the Main Node and the second and third are for the Central Node. When demand is high, the maximum number of the cluster will be 10 EC2 instances, while for the Central Node it will be 8. Provisioning configuration is set to two, which would be the initial capacity of the core instance.



Provisionamento e escalabilidade do cluster - *required* Informações

Choose how Amazon EMR should size your cluster.

Escolha uma opção

- ○ Definir o tamanho do cluster manualmente — Use essa opção se você conhecer os padrões de sua workload com antecedência.
- ● Usar escalabilidade gerenciada pelo EMR — Monitore as principais métricas de workload para que o EMR possa otimizar o tamanho do cluster e a utilização de recursos.
- ○ Usar ajuste de escala automático personalizado — Para escalar programaticamente os nós centrais e de tarefa, crie políticas personalizadas de ajuste de escala automático.

Configuração de escalabilidade

Tamanho mínimo do cluster: 3 instância(s)

Tamanho máximo do cluster: 10 instância(s)

Máximo de nós centrais no cluster

Limite o número de nós centrais em seu cluster.

8 instância(s)

Máximo de instâncias sob demanda no cluster

Para provisionar o nó primário para usar o preço Sob demanda e outros nós no cluster para usar o preço Spot, defina esse valor como 1. Para provisionar todo o cluster para usar o preço Sob demanda, use o mesmo valor que o tamanho máximo do cluster.

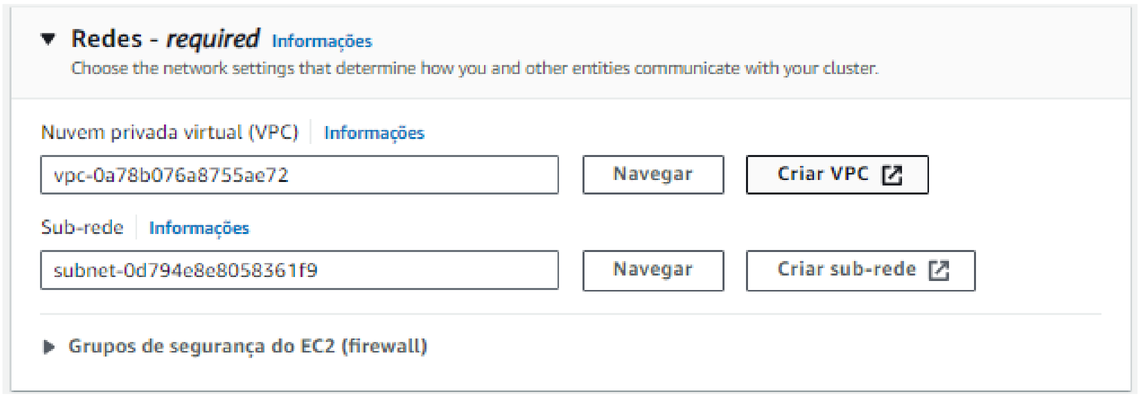
8 instância(s)

Configuração de provisionamento

Defina o tamanho do seu núcleogrupo de instâncias. O Amazon EMR tenta provisionar essa capacidade quando você inicia seu cluster.

Nome	Tipo de instância	Tamanho da(s) instância(s)	Usar a opção de compra Spot
Núcleo	m5.xlarge	2	☐

In the “Networks” section, I selected the VPC that I had previously created, searched, selected the VPC and AWS automatically filled in the information about the VPC and Subnets.



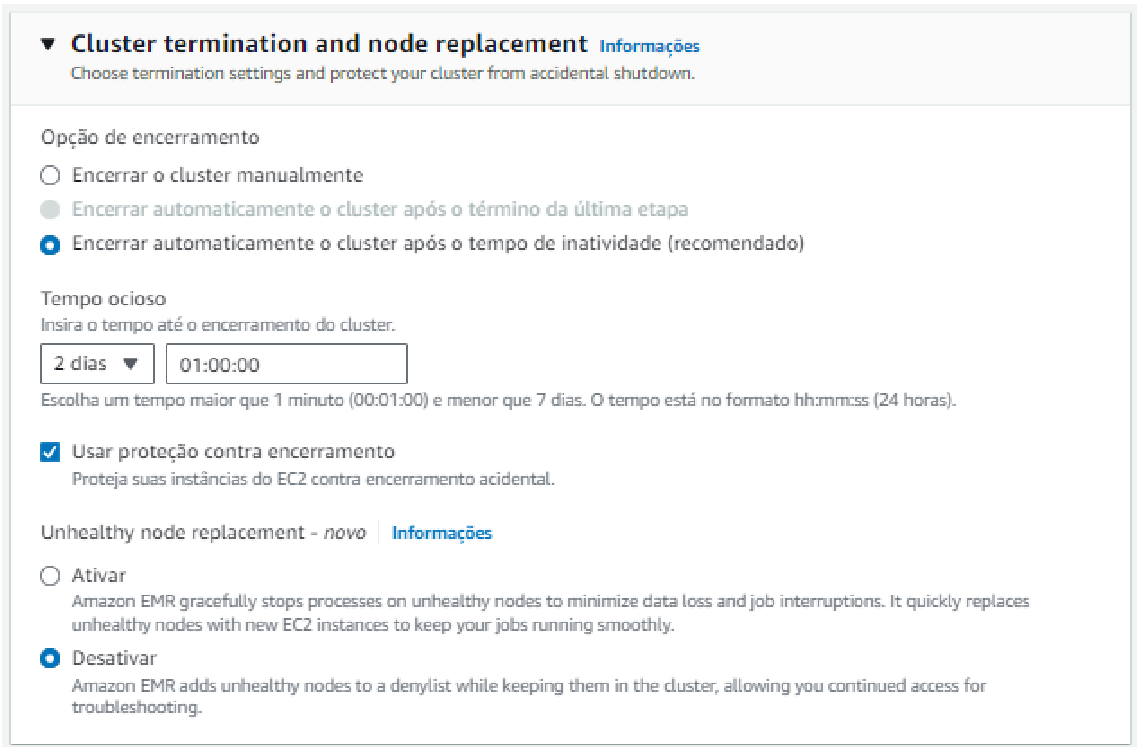
Redes - *required* Informações

Choose the network settings that determine how you and other entities communicate with your cluster.

Nuvem privada virtual (VPC) Informações: vpc-0a78b076a8755ae72

Sub-rede Informações: subnet-0d794e8e8058361f9

In the topic “Cluster termination and node replacement”, I selected “Automatically terminate the cluster after downtime”, because it is a way to reduce cost, so it will be terminated in times of downtime. I selected the option “Use termination protection” which refers to EC2, thus deleting the clusters, but keeping the stored data.



Choose termination settings and protect your cluster from accidental shutdown.

Opção de encerramento

- ○ Encerrar o cluster manualmente
- ○ Encerrar automaticamente o cluster após o término da última etapa
- ● Encerrar automaticamente o cluster após o tempo de inatividade (recomendado)

Tempo ocioso

Insira o tempo até o encerramento do cluster.

2 dias | 01:00:00

Escolha um tempo maior que 1 minuto (00:01:00) e menor que 7 dias. O tempo está no formato hh:mm:ss (24 horas).

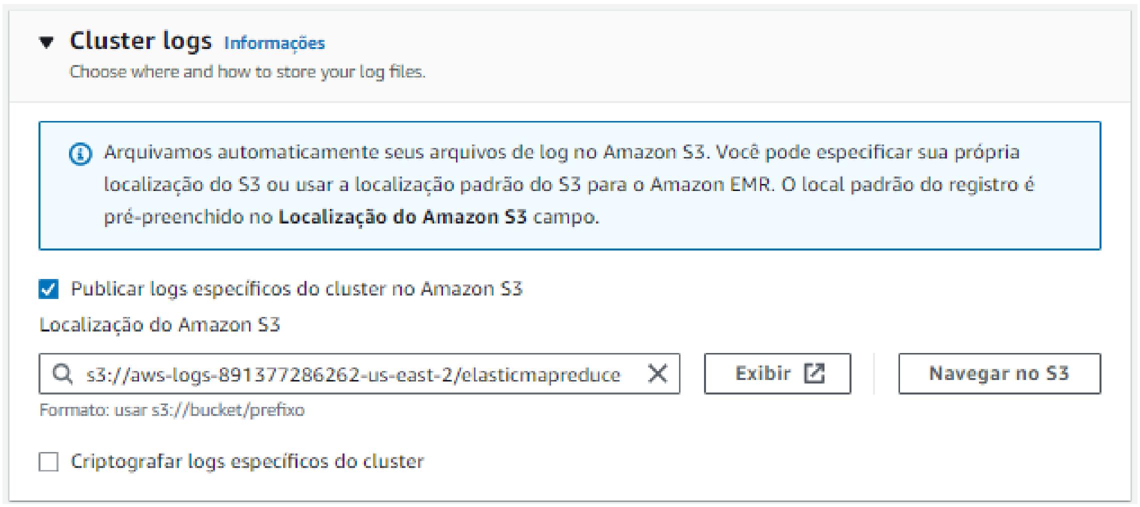
✓ Usar proteção contra encerramento

Proteja suas instâncias do EC2 contra encerramento acidental.

Unhealthy node replacement - *novo* | Informações

- ○ Ativar
  Amazon EMR gracefully stops processes on unhealthy nodes to minimize data loss and job interruptions. It quickly replaces unhealthy nodes with new EC2 instances to keep your jobs running smoothly.
- ● Desativar
  Amazon EMR adds unhealthy nodes to a denylist while keeping them in the cluster, allowing you continued access for troubleshooting.

In the section, “Cluster logs”, I allowed AWS to create a bucket to store the EMR cluster logs.



Choose where and how to store your log files.

Arquivamos automaticamente seus arquivos de log no Amazon S3. Você pode especificar sua própria localização do S3 ou usar a localização padrão do S3 para o Amazon EMR. O local padrão do registro é pré-preenchido no **Localização do Amazon S3** campo.

✓ Publicar logs específicos do cluster no Amazon S3

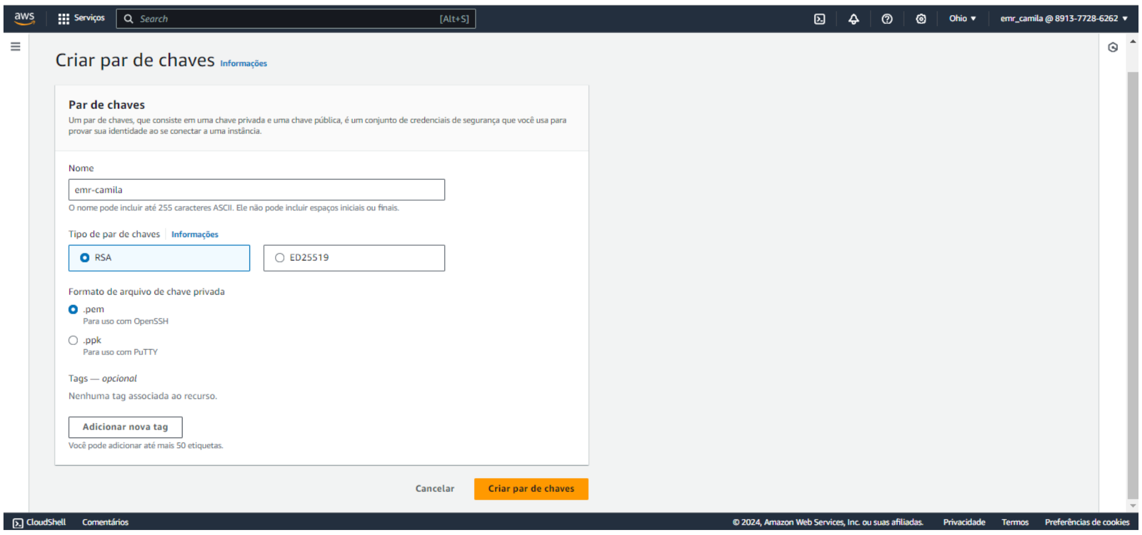
Localização do Amazon S3

s3://aws-logs-891377286262-us-east-2/elasticmapreduce

Formato: usar s3://bucket/prefixo

☐ Criptografar logs específicos do cluster

Regarding “EC2 security and key pair configuration”, the recommended practice is to create a keypair to access via SSH, however I created it directly through the console. Click on the create key pair button, opened another tab , gave it a name, chose the RSA pair and the format. pem , I saved the generated file.



Criar par de chaves Informações

Par de chaves

Um par de chaves, que consiste em uma chave privada e uma chave pública, é um conjunto de credenciais de segurança que você usa para provar sua identidade ao se conectar a uma instância.

Nome: emr-camila

O nome pode incluir até 255 caracteres ASCII. Ele não pode incluir espaços iniciais ou finais.

Tipo de par de chaves: [✓ RSA] ED25519

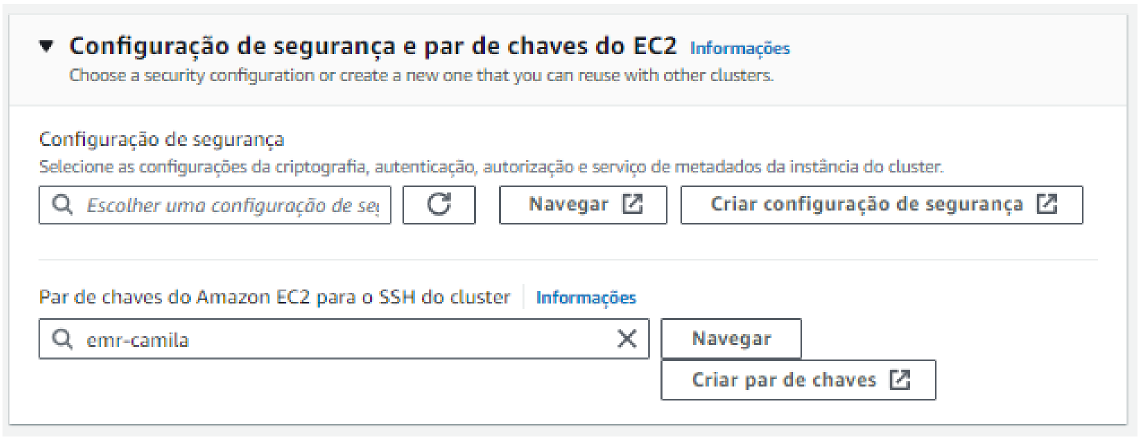
Formato de arquivo de chave privada:
- [✓ .pem] Para uso com OpenSSH
- .ppk Para uso com PuTTY

Tags — *opcional*

Nenhuma tag associada ao recurso.

Você pode adicionar até mais 50 etiquetas.

Then select the keypair in the security configuration as below.



Configuração de segurança e par de chaves do EC2 Informações

Choose a security configuration or create a new one that you can reuse with other clusters.

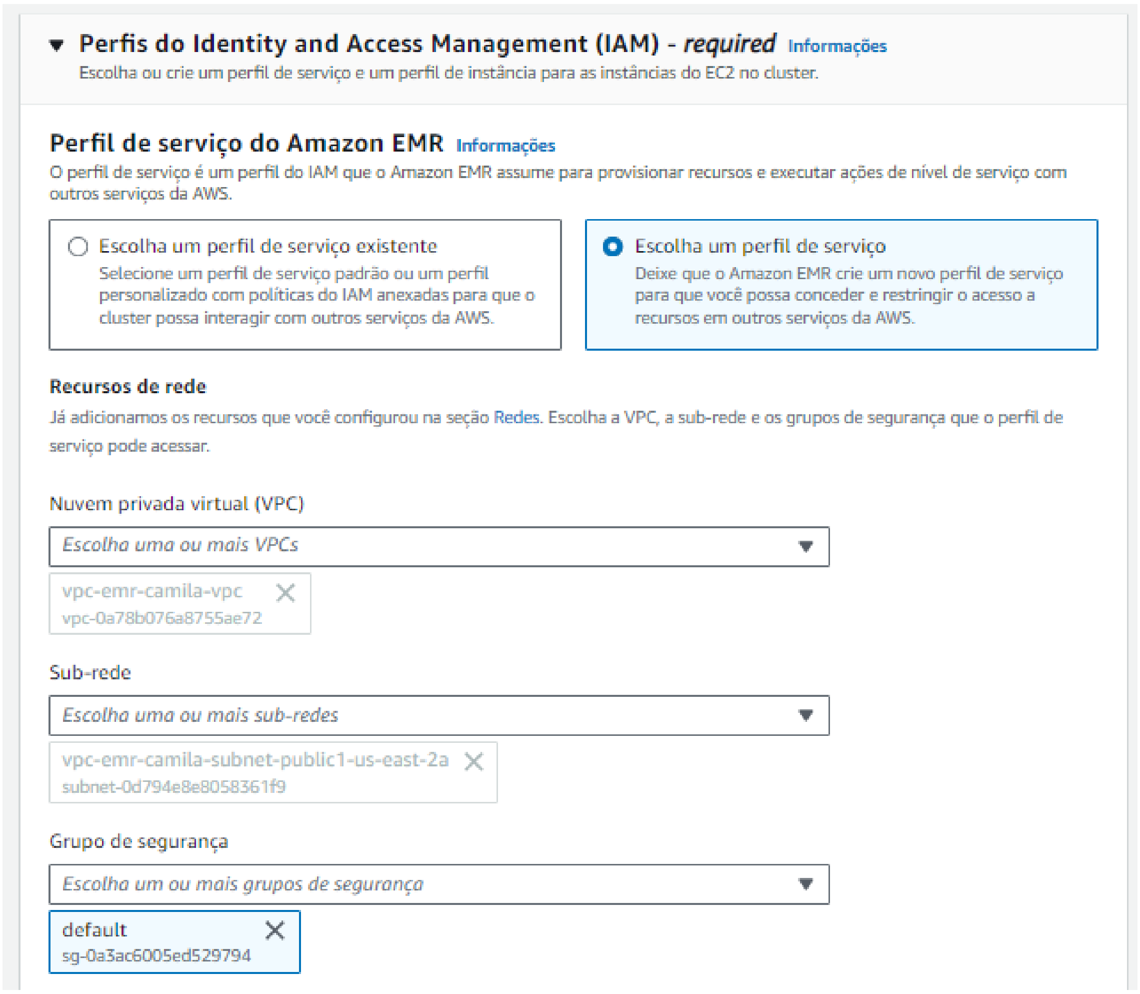
Configuração de segurança

Selecione as configurações da criptografia, autenticação, autorização e serviço de metadados da instância do cluster.

Par de chaves do Amazon EC2 para o SSH do cluster: emr-camila

Escolher uma configuração de se
Navegar
Criar configuração de segurança
Informações
Navegar
Criar par de chaves

In the “IAM Profiles” section, I selected the option choose a service profile, that's why I created the VPC previously, I selected the default security group each time we create a VPC a security group is created automatically.



▼ Perfis do Identity and Access Management (IAM) - *required* Informações

Escolha ou crie um perfil de serviço e um perfil de instância para as instâncias do EC2 no cluster.

Perfil de serviço do Amazon EMR Informações

O perfil de serviço é um perfil do IAM que o Amazon EMR assume para provisionar recursos e executar ações de nível de serviço com outros serviços da AWS.

○ Escolha um perfil de serviço existente
Selecione um perfil de serviço padrão ou um perfil personalizado com políticas do IAM anexadas para que o cluster possa interagir com outros serviços da AWS.

● Escolha um perfil de serviço
Deixe que o Amazon EMR crie um novo perfil de serviço para que você possa conceder e restringir o acesso a recursos em outros serviços da AWS.

Recursos de rede

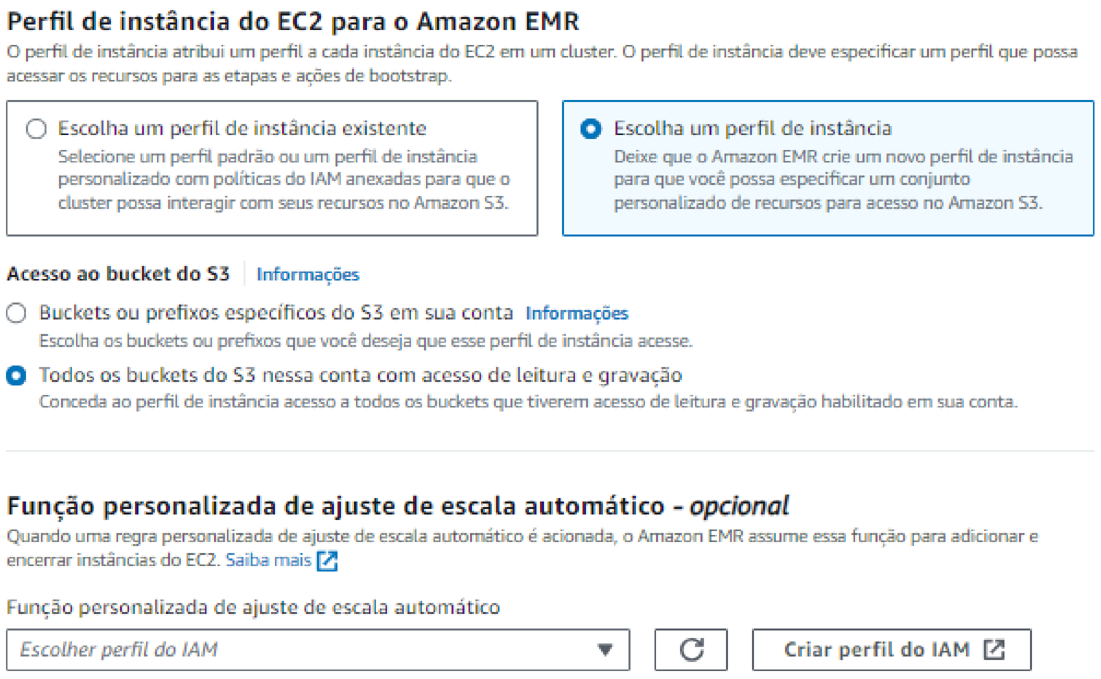
Já adicionamos os recursos que você configurou na seção Redes. Escolha a VPC, a sub-rede e os grupos de segurança que o perfil de serviço pode acessar.

Nuvem privada virtual (VPC)
Escolha uma ou mais VPCs
vpc-emr-camila-vpc
vpc-0a78b076a8755ae72

Sub-rede
Escolha uma ou mais sub-redes
vpc-emr-camila-subnet-public1-us-east-2a
subnet-0d794e8e8058361f9

Grupo de segurança
Escolha um ou mais grupos de segurança
default
sg-0a3ac6005ed529794

After in the “Instance Profile” section, I chose an instance profile, gave access to all buckets for reading and writing, which is why the buckets were created before the clusters.



Perfil de instância do EC2 para o Amazon EMR

O perfil de instância atribui um perfil a cada instância do EC2 em um cluster. O perfil de instância deve especificar um perfil que possa acessar os recursos para as etapas e ações de bootstrap.

○ Escolha um perfil de instância existente
Selecione um perfil padrão ou um perfil de instância personalizado com políticas do IAM anexadas para que o cluster possa interagir com seus recursos no Amazon S3.

● Escolha um perfil de instância
Deixe que o Amazon EMR crie um novo perfil de instância para que você possa especificar um conjunto personalizado de recursos para acesso no Amazon S3.

Acesso ao bucket do S3 Informações

○ Buckets ou prefixos específicos do S3 em sua conta Informações
Escolha os buckets ou prefixos que você deseja que esse perfil de instância acesse.

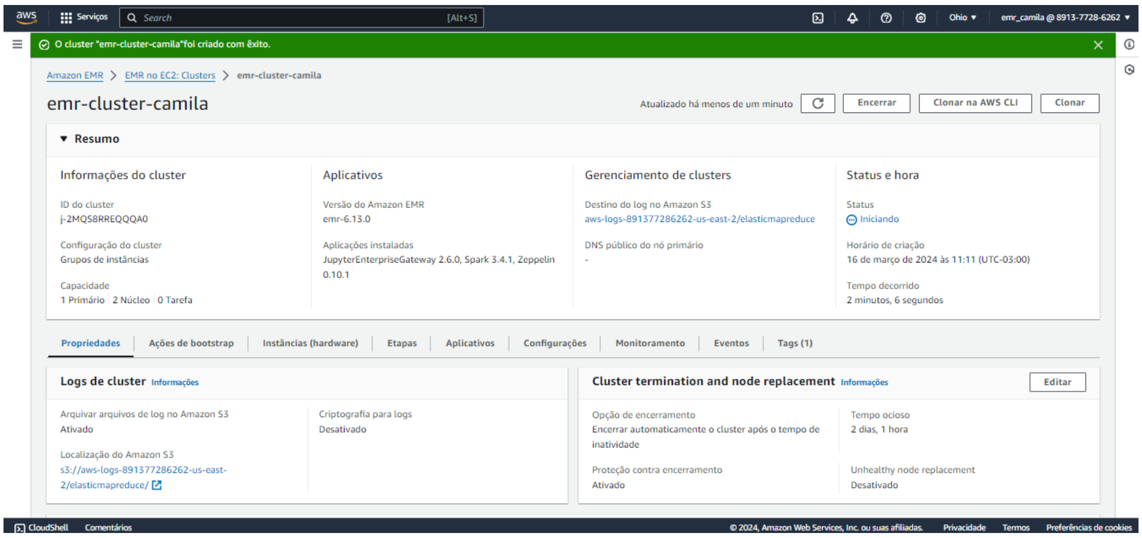
● Todos os buckets do S3 nessa conta com acesso de leitura e gravação
Conceda ao perfil de instância acesso a todos os buckets que tiverem acesso de leitura e gravação habilitado em sua conta.

Função personalizada de ajuste de escala automático - *opcional*

Quando uma regra personalizada de ajuste de escala automático é acionada, o Amazon EMR assume essa função para adicionar e encerrar instâncias do EC2. Saiba mais

Função personalizada de ajuste de escala automático
Escolher perfil do IAM

Afterwards, I clicked on “Create cluster”, which can take up to 7 minutes to complete.



O cluster "emr-cluster-camila"foi criado com êxito.

Amazon EMR > EMR no EC2: Clusters > emr-cluster-camila

emr-cluster-camila

Atualizado há menos de um minuto

Resumo

Informações do cluster
ID do cluster
j-2MQS8RREQQQA0
Configuração do cluster
Grupos de instâncias
Capacidade
1 Primário | 2 Núcleo | 0 Tarefa

Aplicativos
Versão do Amazon EMR
emr-6.13.0
Aplicações instaladas
JupyterEnterpriseGateway 2.6.0, Spark 3.4.1, Zeppelin 0.10.1

Gerenciamento de clusters
Destino do log no Amazon S3
aws-logs-891377286262-us-east-2/elasticmapreduce
DNS público do nó primário
-

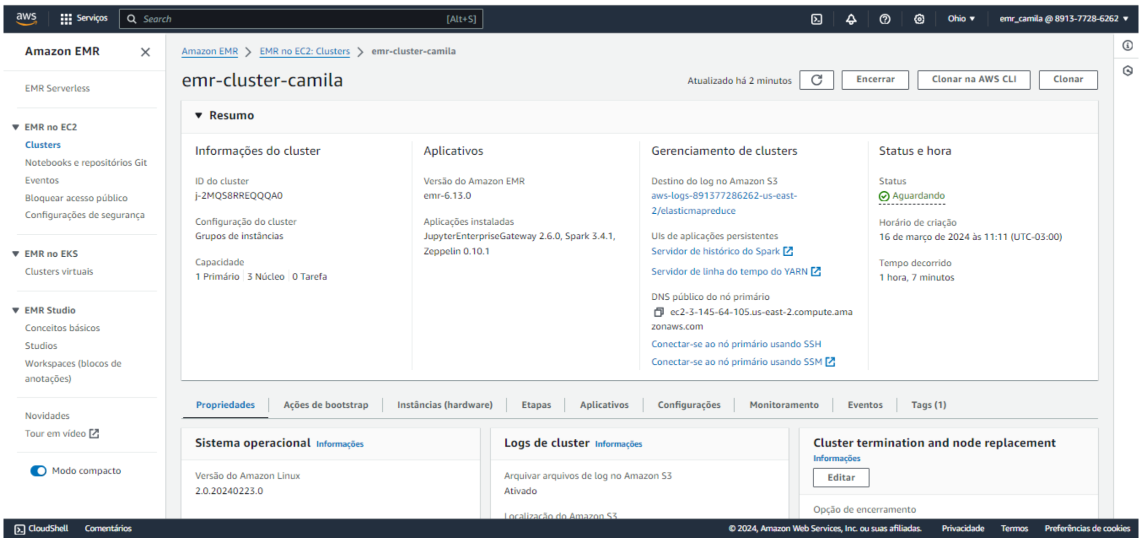
Status e hora
Status
Iniciando
Horário de criação
16 de março de 2024 às 11:11 (UTC-03:00)
Tempo decorrido
2 minutos, 6 segundos

Logs de cluster Informações
Arquivar arquivos de log no Amazon S3
Ativado
Localização do Amazon S3
s3://aws-logs-891377286262-us-east-2/elasticmapreduce/
Criptografia para logs
Desativado

Cluster termination and node replacement Informações
Opção de encerramento
Encerrar automaticamente o cluster após o tempo de inatividade
Tempo ocioso
2 dias, 1 hora
Proteção contra encerramento
Ativado
Unhealthy node replacement
Desativado

6. Configuring Jupyter Notebook to use PySpark

Now it is possible to configure Jupyter Notebook to use PySpark, for this we have to configure a Studio. On the left, in the EMR Studio section, click on basic concepts, image below. After creating studio, a bucket was created to be able to attach Jupyter to the EMR, so the codes written in the notebook will be executed in the EMR and in the S3 bucket via programming. To do this, I needed to create a custom bucket and assign it to my user, first I gave it a name, then I created a bucket just for my studio and associated it with my administrator user.


emr-cluster-camila

Atualizado há 2 minutos

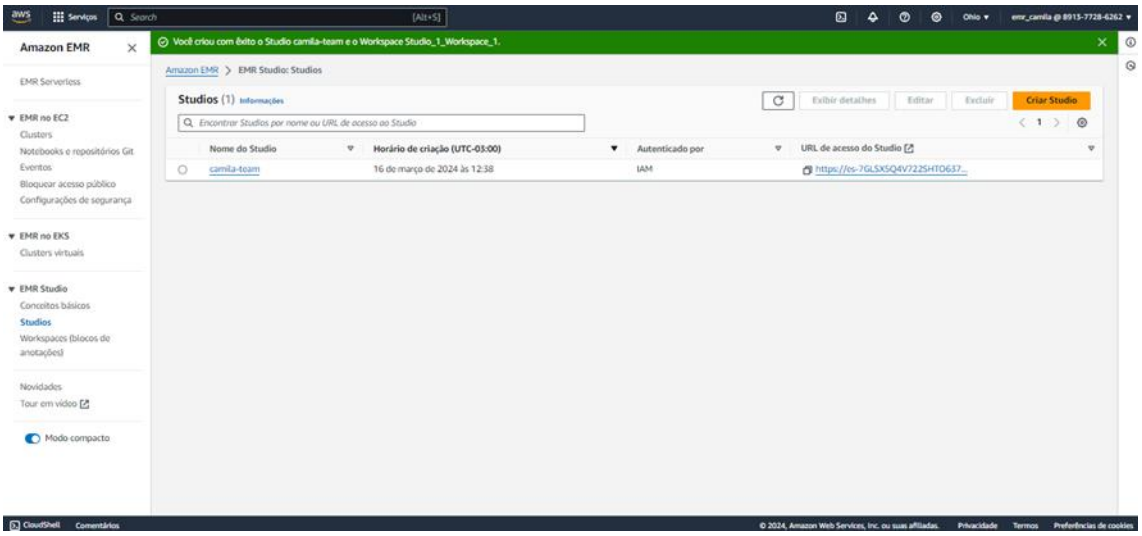
Resumo

Informações do cluster	Aplicativos	Gerenciamento de clusters	Status e hora
ID do cluster: j-2MQS8RREQQQA0	Versão do Amazon EMR: emr-6.13.0	Destino do log no Amazon S3: aws-logs-891377286262-us-east-2/elasticmapreduce	Status: Aguardando
Configuração do cluster: Grupos de instâncias	Aplicações instaladas: JupyterEnterpriseGateway 2.6.0, Spark 3.4.1, Zeppelin 0.10.1	UIs de aplicações persistentes: Servidor de histórico do Spark; Servidor de linha do tempo do YARN	Horário de criação: 16 de março de 2024 às 11:11 (UTC-03:00)
Capacidade: 1 Primário, 3 Núcleo, 0 Tarefa		DNS público do nó primário: ec2-3-145-64-105.us-east-2.compute.amazonaws.com	Tempo decorrido: 1 hora, 7 minutos

Sistema operacional: Versão do Amazon Linux 2.0.20240223.0

Logs de cluster: Arquivar arquivos de log no Amazon S3: Ativado

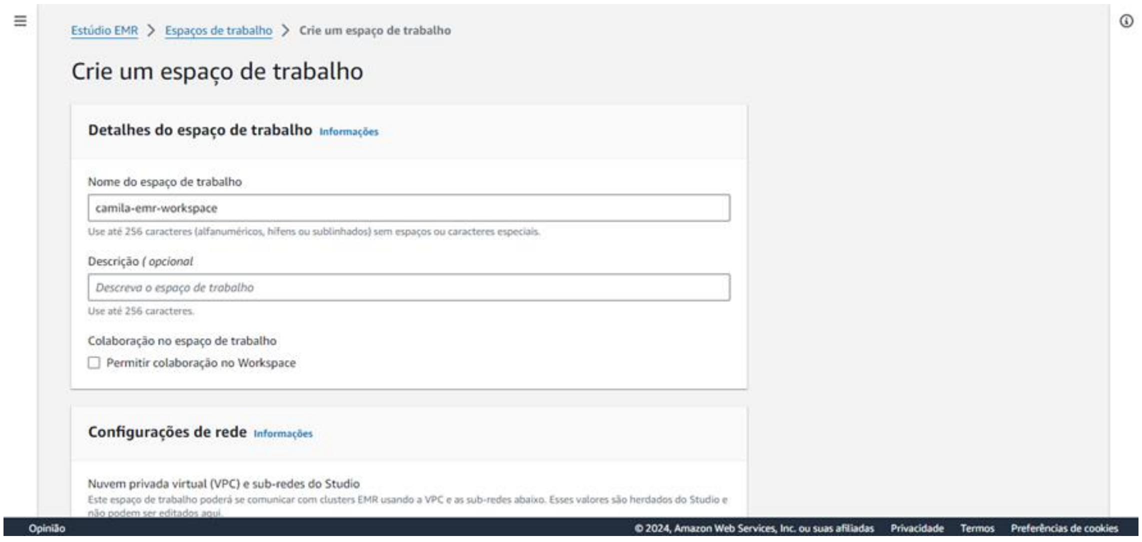
Afterwards, in the “Networks and security” section, I selected the VPC and Subnet that I had already created, remembering that the administrator user must have full access permission to AWS S3 and be an Administrator User, as mentioned in step two, after clicking “Create Studio”. Now we can create the Jupyter notebook, click on the URL of the created Studio, as shown below.


Você criou com êxito o Studio camila-team e o Workspace Studio_1_Workspace_1.

Studios (1)

Nome do Studio	Horário de criação (UTC-03:00)	Autenticado por	URL de acesso do Studio
camila-team	16 de março de 2024 às 12:38	IAM	https://es-7GLSXSQ4V722SHTO637...

I clicked on “Create workspace” chose a name, the other settings remain the same for network and storage.



Estúdio EMR > Espaços de trabalho > Crie um espaço de trabalho

Crie um espaço de trabalho

Detalhes do espaço de trabalho Informações

Nome do espaço de trabalho

camila-emr-workspace

Use até 256 caracteres (alfanuméricos, hifens ou sublinhados) sem espaços ou caracteres especiais.

Descrição *(opcional*

Descreva o espaço de trabalho

Use até 256 caracteres.

Colaboração no espaço de trabalho

☐ Permitir colaboração no Workspace

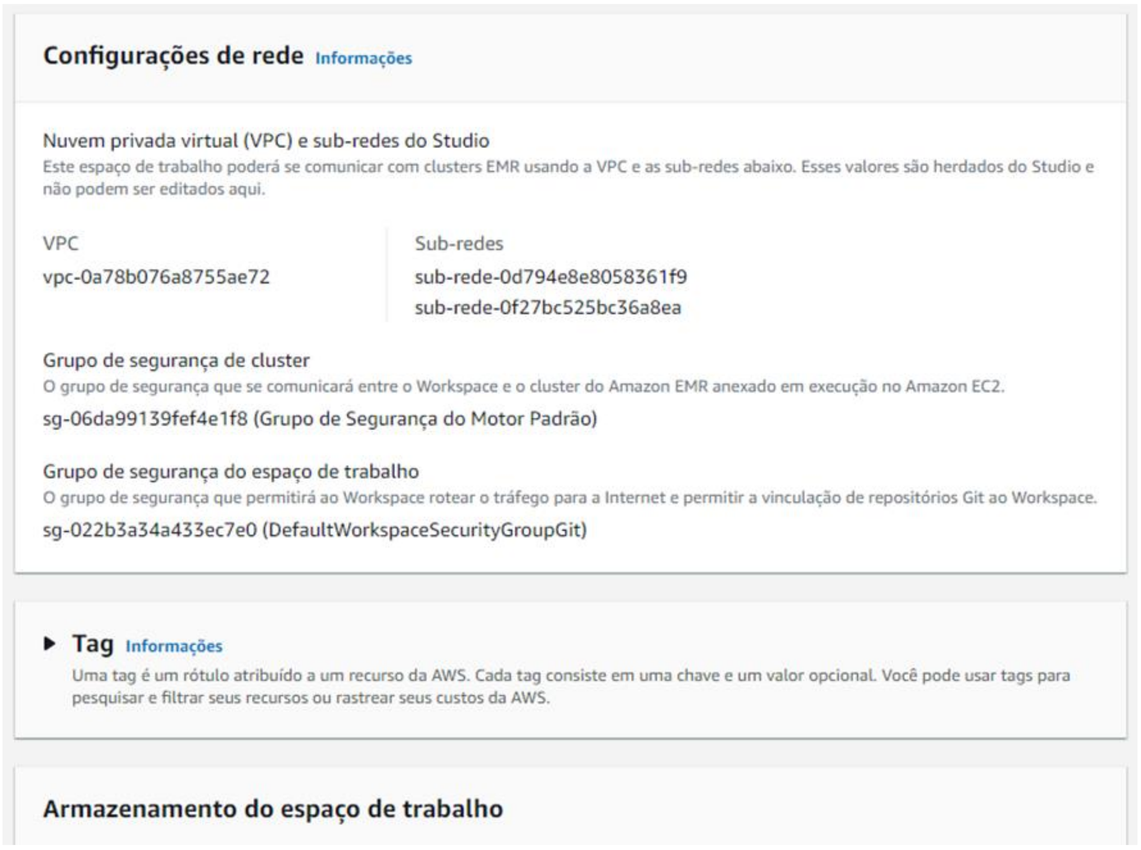
Configurações de rede Informações

Nuvem privada virtual (VPC) e sub-redes do Studio

Este espaço de trabalho poderá se comunicar com clusters EMR usando a VPC e as sub-redes abaixo. Esses valores são herdados do Studio e não podem ser editados aqui.

Opinião © 2024, Amazon Web Services, Inc. ou suas afiliadas Privacidade Termos Preferências de cookies

VPC that you had already created.



Configurações de rede Informações

Nuvem privada virtual (VPC) e sub-redes do Studio

Este espaço de trabalho poderá se comunicar com clusters EMR usando a VPC e as sub-redes abaixo. Esses valores são herdados do Studio e não podem ser editados aqui.

VPC	Sub-redes
vpc-0a78b076a8755ae72	sub-rede-0d794e8e8058361f9
	sub-rede-0f27bc525bc36a8ea

Grupo de segurança de cluster

O grupo de segurança que se comunicará entre o Workspace e o cluster do Amazon EMR anexado em execução no Amazon EC2.

sg-06da99139fef4e1f8 (Grupo de Segurança do Motor Padrão)

Grupo de segurança do espaço de trabalho

O grupo de segurança que permitirá ao Workspace rotear o tráfego para a Internet e permitir a vinculação de repositórios Git ao Workspace.

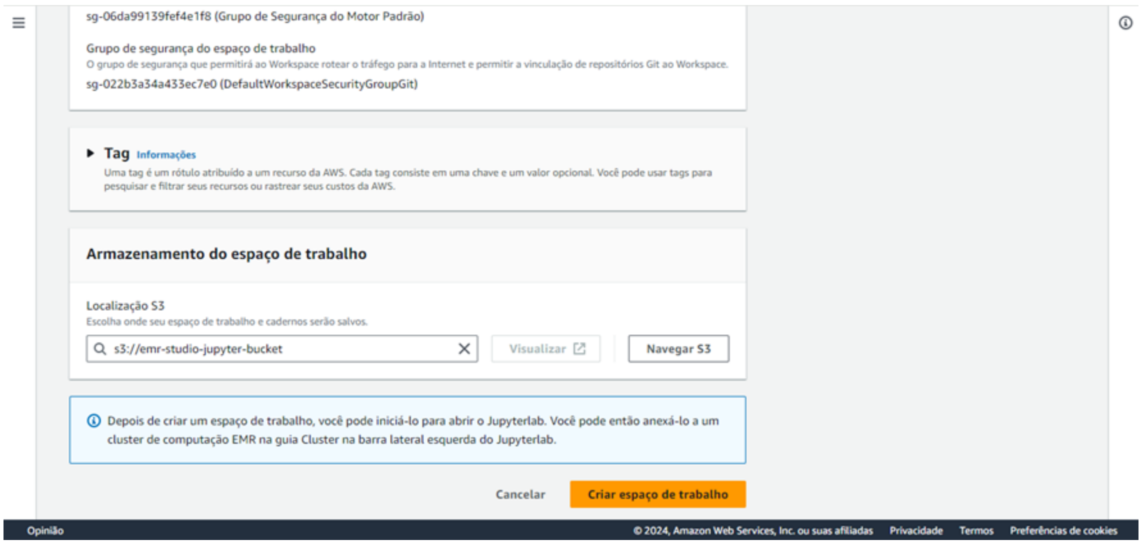
sg-022b3a34a433ec7e0 (DefaultWorkspaceSecurityGroupGit)

▶ **Tag** Informações

Uma tag é um rótulo atribuído a um recurso da AWS. Cada tag consiste em uma chave e um valor opcional. Você pode usar tags para pesquisar e filtrar seus recursos ou rastrear seus custos da AWS.

Armazenamento do espaço de trabalho

S3 bucket created specifically for Jupyter.


sg-06da99139fef4e1f8 (Grupo de Segurança do Motor Padrão)

Grupo de segurança do espaço de trabalho
O grupo de segurança que permitirá ao Workspace rotear o tráfego para a Internet e permitir a vinculação de repositórios Git ao Workspace.
sg-022b3a34a433ec7e0 (DefaultWorkspaceSecurityGroupGit)

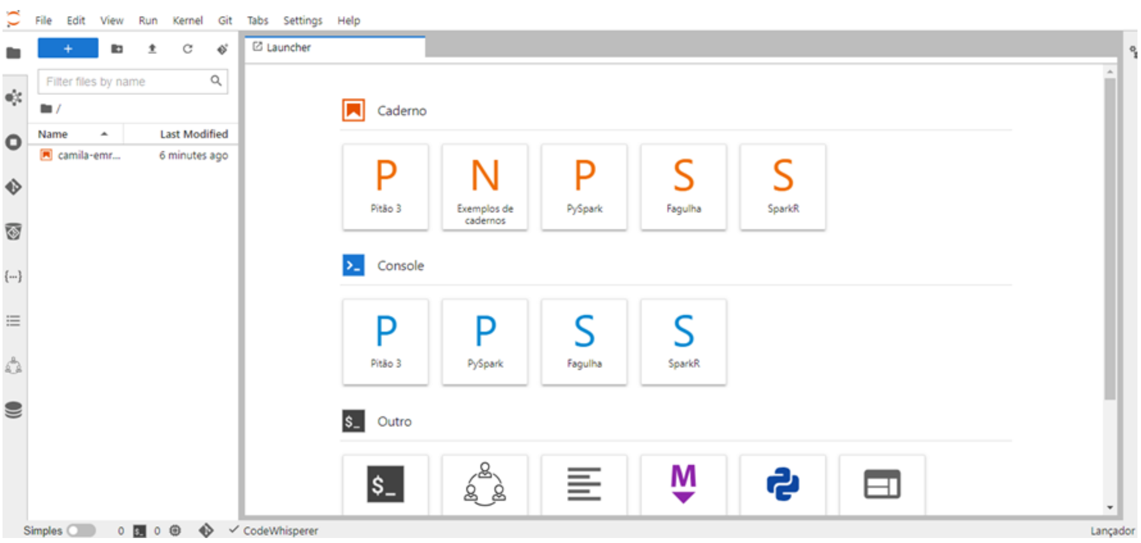
Tag Informações
Uma tag é um rótulo atribuído a um recurso da AWS. Cada tag consiste em uma chave e um valor opcional. Você pode usar tags para pesquisar e filtrar seus recursos ou rastrear seus custos da AWS.

Armazenamento do espaço de trabalho

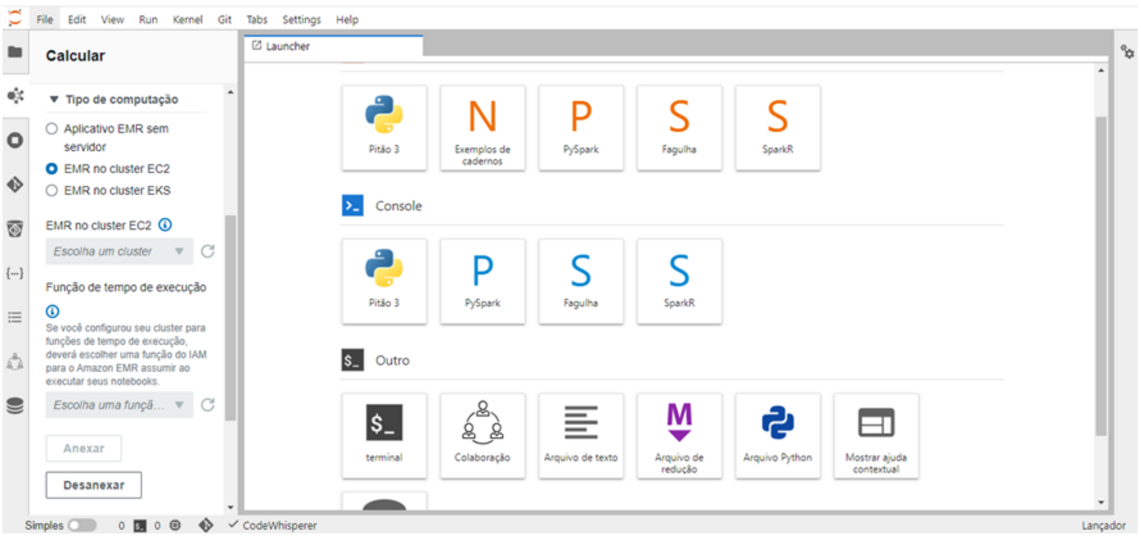
Localização S3
Escolha onde seu espaço de trabalho e cadernos serão salvos.
s3://emr-studio-jupyter-bucket

Depois de criar um espaço de trabalho, você pode iniciá-lo para abrir o Jupyterlab. Você pode então anexá-lo a um cluster de computação EMR na guia Cluster na barra lateral esquerda do Jupyterlab.

I clicked on Studio below, and it opened Jupyter Notebook in a new tab.



Now it is necessary to attach Jupyter to the cluster that we created, and we must select the EC2 option.

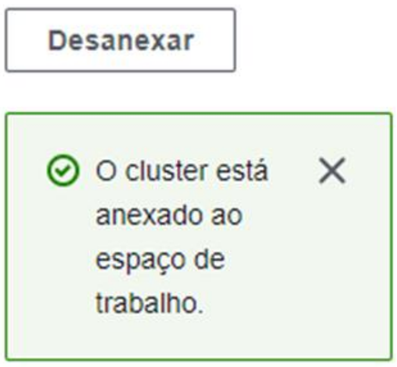

Calcular

Tipo de computação
- Aplicativo EMR sem servidor
- [✓ EMR no cluster EC2]
- EMR no cluster EKS

EMR no cluster EC2
Escolha um cluster

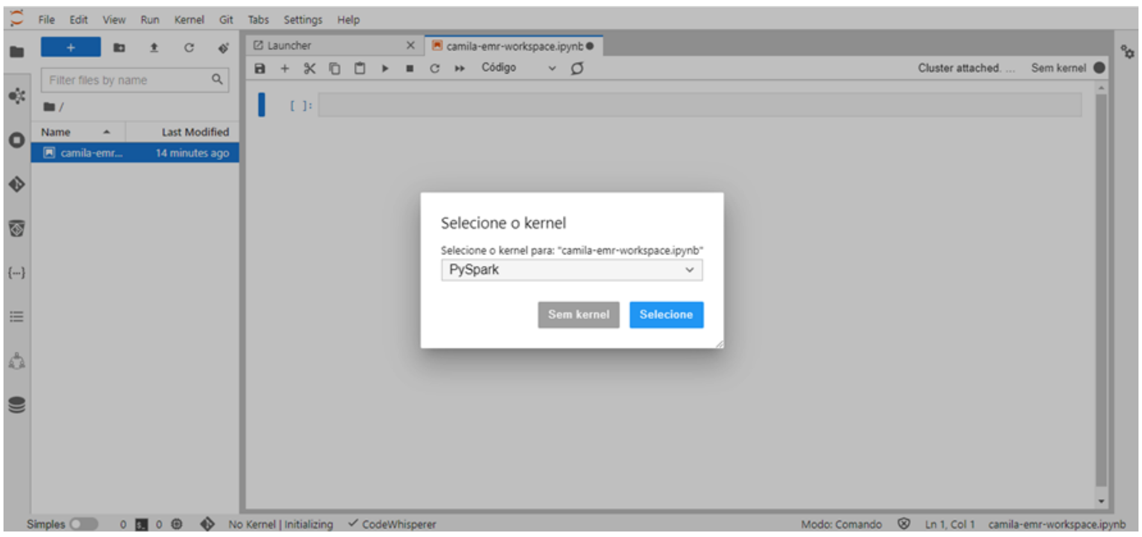
Função de tempo de execução
Se você configurou seu cluster para funções de tempo de execução, deverá escolher uma função do IAM para o Amazon EMR assumir ao executar seus notebooks.
Escolha uma funçã...

I refreshed the page and the message below appeared under “EMR Computing”.


Desanexar

O cluster está anexado ao espaço de trabalho.

Afterwards, I went back to my workspace, double-clicked on the notebook and chose the kernel as Pyspark .



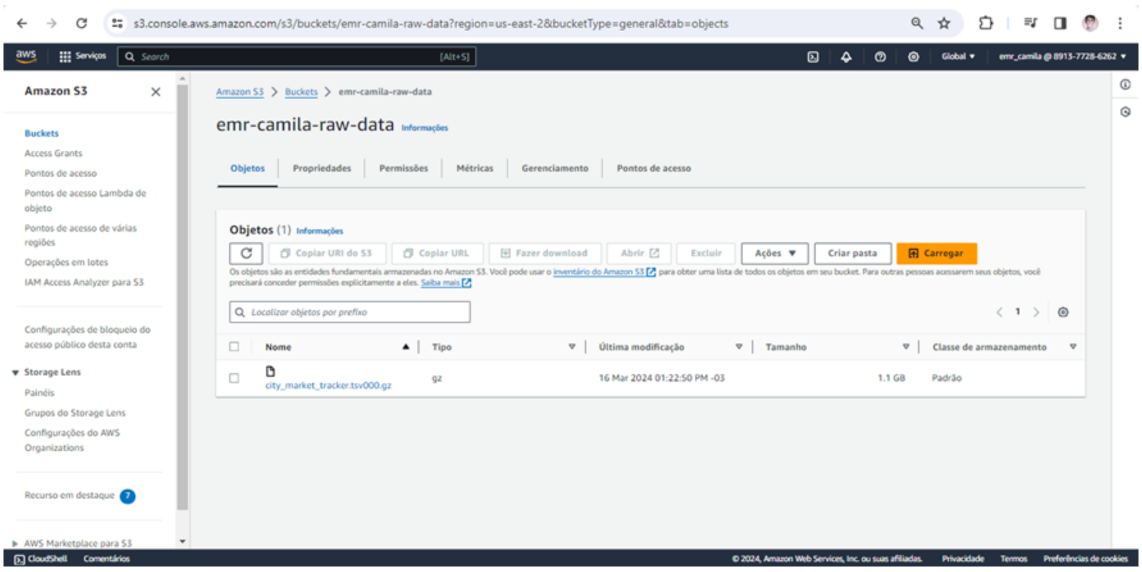
Selecione o kernel

Selecione o kernel para: "camila-emr-workspace.ipynb"

PySpark

7. Code Explanation

In the first cell, we imported Spark, in the second, I loaded directly from the Redfin page and stored it in the first bucket created, emr -camila- raw -data, the bucket that was empty, already received the load, as shown below.


s3.console.aws.amazon.com/s3/buckets/emr-camila-raw-data?region=us-east-2&bucketType=general&tab=objects
Serviços
Search
[Alt+S]
Global
emr_camila @ 8913-7728-6262
Amazon S3
Buckets
Access Grants
Pontos de acesso
Pontos de acesso Lambda de objeto
Pontos de acesso de várias regiões
Operações em lotes
IAM Access Analyzer para S3
Configurações de bloqueio do acesso público desta conta
Storage Lens
Painéis
Grupos do Storage Lens
Configurações do AWS Organizations
Recurso em destaque
AWS Marketplace para S3
Amazon S3 > Buckets > emr-camila-raw-data
Objetos
Propriedades
Permissões
Métricas
Gerenciamento
Pontos de acesso
Copiar URI do S3
Copiar URL
Fazer download
Abrir
Excluir
Ações
Criar pasta
Carregar
Localizar objetos por prefixo
CloudShell
Comentários
© 2024, Amazon Web Services, Inc. ou suas afiliadas.
Privacidade
Termos
Preferências de cookies

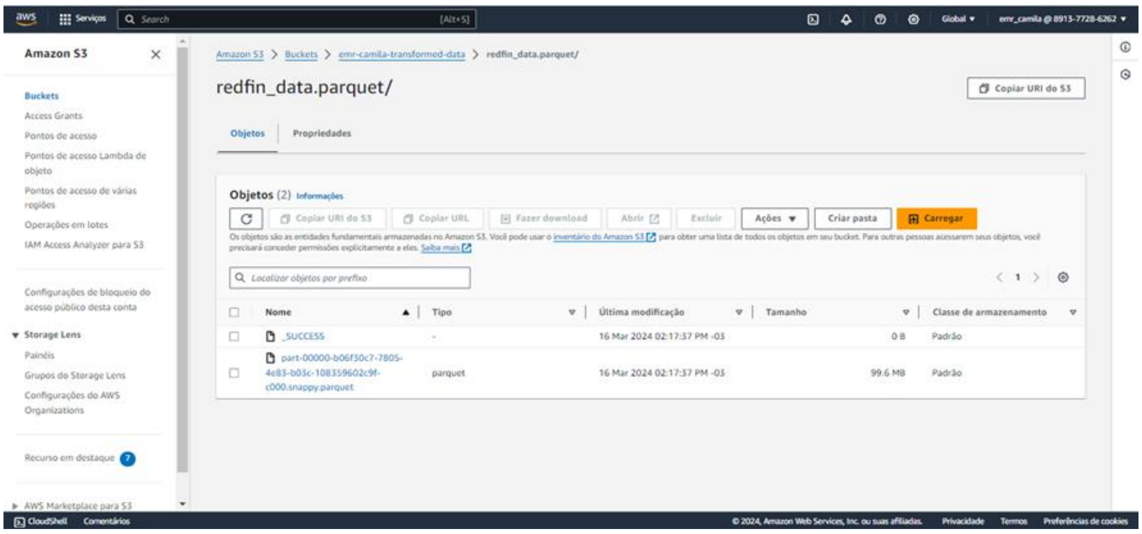
emr-camila-raw-data Informações

Objetos (1) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o inventário do Amazon S3 para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. Saiba mais

Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
city_market_tracker.tsv000.gz	gz	16 Mar 2024 01:22:50 PM -03	1.1 GB	Padrão

After processing and cleaning, I saved the transformed data in the second bucket I created, as shown below, we have the file and the success log. The file was saved in Parquet because of the size, but it could have been in csv, but aiming for better storage and lower cost, I chose the Parquet extension.


Serviços
Search
[Alt+S]
Global
emr_camila @ 8913-7728-6262
Amazon S3
Buckets
Access Grants
Pontos de acesso
Pontos de acesso Lambda de objeto
Pontos de acesso de várias regiões
Operações em lotes
IAM Access Analyzer para S3
Configurações de bloqueio do acesso público desta conta
Storage Lens
Painéis
Grupos do Storage Lens
Configurações do AWS Organizations
Recurso em destaque
AWS Marketplace para S3
Amazon S3 > Buckets > emr-camila-transformed-data > redfin_data.parquet/
Copiar URI do S3
Objetos
Propriedades
Copiar URI do S3
Copiar URL
Fazer download
Abrir
Excluir
Ações
Criar pasta
Carregar
Localizar objetos por prefixo
CloudShell
Comentários
© 2024, Amazon Web Services, Inc. ou suas afiliadas.
Privacidade
Termos
Preferências de cookies

redfin_data.parquet/

Objetos (2) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o inventário do Amazon S3 para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. Saiba mais

Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
_SUCCESS	-	16 Mar 2024 02:17:37 PM -03	0 B	Padrão
part-00000-b06f30c7-7805-4e83-b03c-108359602c9f-c000.snappy.parquet	parquet	16 Mar 2024 02:17:37 PM -03	99.6 MB	Padrão

8. Complete Jupyter Notebook

Next, I attached the Jupyter Notebook with all the codes, transformations and changes, the cells have commented code. It is important to remember that the database provided by RedFin was mostly clean and organized, however, column adjustments, checking NA, Null values and changing variable names were applied. All the basis was taken from the AWS website and the idea for the project was taken from the Telespectra channel.

camila-emr-workspace

March 16, 2024

[1]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
```

```
VBox()

Starting Spark application

<IPython.core.display.HTML object>

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…

SparkSession available as 'spark'.

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[3]:
```
spark = SparkSession.builder.appName("CamilaRedfinDataAnalysis").getOrCreate()
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[4]:
```
# extrai do bucket emr-camila-raw-data para o início das análises
redfin_data = spark.read.csv("s3://emr-camila-raw-data/city_market_tracker.
↪tsv000.gz", header=True, inferSchema=True, sep= "\t")
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[5]:
```
redfin_data.show(3)
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…

+------------+----------+---------------+----------+--------------+--------+---
-------------------+--------------------+-----------------+-------------+--------
---+--------------------+----------------+-----------------+--------------------
```

```
-+---------------------+-----------------+---------------------+----------------
-----+------------------+--------------------+--------------------+-------------
-----+--------------------+--------------------+----------+-------------------+-
-----------------+-------------+--------------------+------------------+--------
----+------------------+-------------------+---------+-------------------+------
--------------+----------------+--------------------+--------------------+------
----+--------------+--------------+------------------+--------------------+-----
---------------+-------------------+--------------------+--------------------+--
-----------------+--------------------+--------------------+--------------------
---+---------------------------+---------------------------+-------------------+
------------------------------+-------------------+
|period_begin|period_end|period_duration|region_type|region_type_id|table_id|is_
seasonally_adjusted|              region|             city|
state|state_code|       property_type|property_type_id|median_sale_price|median_
sale_price_mom|median_sale_price_yoy|median_list_price|median_list_price_mom|med
ian_list_price_yoy|       median_ppsf|     median_ppsf_mom|     median_ppsf_yoy|
median_list_ppsf|median_list_ppsf_mom|median_list_ppsf_yoy|homes_sold|
homes_sold_mom|    homes_sold_yoy|pending_sales|   pending_sales_mom|
pending_sales_yoy|new_listings|  new_listings_mom|   new_listings_yoy|inventory|
inventory_mom|       inventory_yoy|months_of_supply|months_of_supply_mom|months_
of_supply_yoy|median_dom|median_dom_mom|median_dom_yoy|
avg_sale_to_list|avg_sale_to_list_mom|avg_sale_to_list_yoy|    sold_above_list|
sold_above_list_mom| sold_above_list_yoy|        price_drops|
price_drops_mom|     price_drops_yoy|off_market_in_two_weeks|off_market_in_two_w
eeks_mom|off_market_in_two_weeks_yoy|parent_metro_region|parent_metro_region_met
ro_code|       last_updated|
+------------+----------+---------------+-----------+--------------+--------+---
-------------------+--------------------+-----------------+-------------+-------
---+--------------------+----------------+-----------------+--------------------
-+---------------------+-----------------+---------------------+----------------
-----+------------------+--------------------+--------------------+-------------
-----+--------------------+--------------------+----------+-------------------+-
-----------------+-------------+--------------------+------------------+--------
----+------------------+-------------------+---------+-------------------+------
--------------+----------------+--------------------+--------------------+------
----+--------------+--------------+------------------+--------------------+-----
---------------+-------------------+--------------------+--------------------+--
-----------------+--------------------+--------------------+--------------------
---+---------------------------+---------------------------+-------------------+
------------------------------+-------------------+
|  2021-02-01|2021-02-28|             30|      place|             6|    5779|
f|       Fair Lawn, NJ|        Fair Lawn|   New Jersey|        NJ|Single Family
Res…|               6|         542500.0|  0.21378230227094752|
0.35964912280701755|         469000.0| -0.14711765775595564|
0.04454342984409809| 278.1372476167421|-0.03125719269388…|-0.00459122296016…
|291.19741100323625|-0.02531486920188…|-0.02420798775354…|
20|-0.4117647058823529|0.4285714285714286|
14|-0.26315789473684215|0.2727272727272727|
25|0.7857142857142858|-0.3902439024390244|       69|
0.0|-0.34905660377358494|            3.5|                 1.5|
-4.1|        91|             8|            -2|
1.0|-0.01214315981126557|0.020401919432001314|               0.0|-0.20588235294
117646|-0.14285714285714285|0.08695652173913043|-0.07246376811594202|
0.03978671041837571|    0.21428571428571427|        0.10902255639097744|
-0.05844155844155843|       New York, NY|
35614|2024-03-10 14:36:40|
|  2019-04-01|2019-04-30|             30|      place|             6|    6504|
f|         Elyria, OH|           Elyria|         Ohio|        OH|Multi-Family
(2-4…|               4|          30000.0|  -0.7014925373134329|
-0.6557165399512266|         110000.0| -0.04264577893820709|
-0.06542056074766356|18.610421836228287| -0.6026777101693752|
-0.5572373301855638|
81.52173913043478|0.004286982260352179|0.005327375837772763|         3|
0.5|              0.5|            2|                 null|               1.0|
4|               3.0|0.33333333333333326|        7|                 0.0|
-0.5|              2.3| -1.2000000000000002|                -4.7|       130|
-149|           33|
0.860358532597008|-0.02600566975054…|0.007997142263547308|                0.0|
0.0|                0.0|              null|                 null|
null|                   1.0|                      null|
1.0|      Cleveland, OH|                        17460|2024-03-10 14:36:40|
|  2020-08-01|2020-08-31|             30|      place|             6|   24459|
f|Northwest Harwich…|Northwest Harwich|Massachusetts|        MA|Single Family
Res…|              6|         625000.0|   0.3789299503585217|
-0.14617486338797814|         649000.0|  0.26019417475728157|
0.42668718399648276|340.65934065934067| 0.12348060103162162|
0.31284153005464477| 393.0817610062893|  0.2546073635603787|
0.18042570872759556|         11|0.22222222222222232|2.6666666666666665|
8| -0.4285714285714286|              1.0|
14|0.5555555555555556|0.07692307692307687|       14|0.07692307692307687|
-0.7407407407407407|              1.3|-0.09999999999999987|              -16.7|
26|          -51|            12|0.9771452130091354|0.022599268740304268|-0.0270
9829717127…|0.18181818181818182| 0.07070707070707072| -0.4848484848484848|
0.2857142857142857|-0.09890109890109894|0.007936507936507908|
0.5|        0.4285714285714286|                       0.5|Barnstable Town, MA|
12700|2024-03-10 14:36:40|
+------------+----------+--------------+----------+-------------+--------+---
-------------------+--------------------+-----------------+-------------+-------
---+--------------------+----------------+------------------+-------------------
-+---------------------+----------------+--------------------+----------------
-----+-----------------+--------------------+--------------------+-------------
-----+--------------------+--------------------+----------+-------------------+-
-----------------+-------------+--------------------+-----------------+--------
----+------------------+-------------------+---------+-------------------+------
--------------+----------------+--------------------+--------------------+------
----+-------------+--------------+------------------+-------------------+-----
```

```
---------------+-------------------+--------------------+--------------------+--
-----------------+--------------------+--------------------+--------------------
---+---------------------------+---------------------------+-------------------+
----------------------------+-------------------+
only showing top 3 rows
```

[6]:
```
#visualiza o schema
redfin_data.printSchema()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
root
 |-- period_begin: date (nullable = true)
 |-- period_end: date (nullable = true)
 |-- period_duration: integer (nullable = true)
 |-- region_type: string (nullable = true)
 |-- region_type_id: integer (nullable = true)
 |-- table_id: integer (nullable = true)
 |-- is_seasonally_adjusted: string (nullable = true)
 |-- region: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- state_code: string (nullable = true)
 |-- property_type: string (nullable = true)
 |-- property_type_id: integer (nullable = true)
 |-- median_sale_price: double (nullable = true)
 |-- median_sale_price_mom: double (nullable = true)
 |-- median_sale_price_yoy: double (nullable = true)
 |-- median_list_price: double (nullable = true)
 |-- median_list_price_mom: double (nullable = true)
 |-- median_list_price_yoy: double (nullable = true)
 |-- median_ppsf: double (nullable = true)
 |-- median_ppsf_mom: double (nullable = true)
 |-- median_ppsf_yoy: double (nullable = true)
 |-- median_list_ppsf: double (nullable = true)
 |-- median_list_ppsf_mom: double (nullable = true)
 |-- median_list_ppsf_yoy: double (nullable = true)
 |-- homes_sold: integer (nullable = true)
 |-- homes_sold_mom: double (nullable = true)
 |-- homes_sold_yoy: double (nullable = true)
 |-- pending_sales: integer (nullable = true)
 |-- pending_sales_mom: double (nullable = true)
 |-- pending_sales_yoy: double (nullable = true)
 |-- new_listings: integer (nullable = true)
 |-- new_listings_mom: double (nullable = true)
 |-- new_listings_yoy: double (nullable = true)
 |-- inventory: integer (nullable = true)
 |-- inventory_mom: double (nullable = true)
 |-- inventory_yoy: double (nullable = true)
 |-- months_of_supply: double (nullable = true)
 |-- months_of_supply_mom: double (nullable = true)
 |-- months_of_supply_yoy: double (nullable = true)
 |-- median_dom: integer (nullable = true)
 |-- median_dom_mom: integer (nullable = true)
 |-- median_dom_yoy: integer (nullable = true)
 |-- avg_sale_to_list: double (nullable = true)
 |-- avg_sale_to_list_mom: double (nullable = true)
 |-- avg_sale_to_list_yoy: double (nullable = true)
 |-- sold_above_list: double (nullable = true)
 |-- sold_above_list_mom: double (nullable = true)
 |-- sold_above_list_yoy: double (nullable = true)
 |-- price_drops: double (nullable = true)
 |-- price_drops_mom: double (nullable = true)
 |-- price_drops_yoy: double (nullable = true)
 |-- off_market_in_two_weeks: double (nullable = true)
 |-- off_market_in_two_weeks_mom: double (nullable = true)
 |-- off_market_in_two_weeks_yoy: double (nullable = true)
 |-- parent_metro_region: string (nullable = true)
 |-- parent_metro_region_metro_code: integer (nullable = true)
 |-- last_updated: timestamp (nullable = true)
```

[7]:
```
#print os nomes das colunas
redfin_data.columns
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
['period_begin', 'period_end', 'period_duration', 'region_type',
'region_type_id', 'table_id', 'is_seasonally_adjusted', 'region', 'city',
'state', 'state_code', 'property_type', 'property_type_id', 'median_sale_price',
'median_sale_price_mom', 'median_sale_price_yoy', 'median_list_price',
'median_list_price_mom', 'median_list_price_yoy', 'median_ppsf',
'median_ppsf_mom', 'median_ppsf_yoy', 'median_list_ppsf',
'median_list_ppsf_mom', 'median_list_ppsf_yoy', 'homes_sold', 'homes_sold_mom',
'homes_sold_yoy', 'pending_sales', 'pending_sales_mom', 'pending_sales_yoy',
'new_listings', 'new_listings_mom', 'new_listings_yoy', 'inventory',
'inventory_mom', 'inventory_yoy', 'months_of_supply', 'months_of_supply_mom',
'months_of_supply_yoy', 'median_dom', 'median_dom_mom', 'median_dom_yoy',
'avg_sale_to_list', 'avg_sale_to_list_mom', 'avg_sale_to_list_yoy',
'sold_above_list', 'sold_above_list_mom', 'sold_above_list_yoy', 'price_drops',
'price_drops_mom', 'price_drops_yoy', 'off_market_in_two_weeks',
'off_market_in_two_weeks_mom', 'off_market_in_two_weeks_yoy',
'parent_metro_region', 'parent_metro_region_metro_code', 'last_updated']
```

[8]:
```
df_redfin = redfin_data.select(['period_end','period_duration', 'city',␣
↪'state', 'property_type',
     'median_sale_price', 'median_ppsf', 'homes_sold', 'inventory',␣
↪'months_of_supply', 'median_dom', 'sold_above_list', 'last_updated'])
df_redfin.show(3)
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…

+----------+---------------+-----------------+-------------+--------------------
+-----------------+------------------+----------+---------+----------------+----
------+--------------------+-------------------+
|period_end|period_duration|             city|        state|
property_type|median_sale_price|
median_ppsf|homes_sold|inventory|months_of_supply|median_dom|
sold_above_list|       last_updated|
+----------+---------------+-----------------+-------------+--------------------
+-----------------+------------------+----------+---------+----------------+----
------+--------------------+-------------------+
|2021-02-28|             30|        Fair Lawn|   New Jersey|Single Family
Res…|         542500.0| 278.1372476167421|        20|       69|
3.5|        91|                 0.0|2024-03-10 14:36:40|
|2019-04-30|             30|           Elyria|         Ohio|Multi-Family
(2-4…|          30000.0|18.610421836228287|         3|        7|
2.3|       130|                 0.0|2024-03-10 14:36:40|
|2020-08-31|             30|Northwest Harwich|Massachusetts|Single Family
Res…|         625000.0|340.65934065934067|        11|       14|
1.3|        26|0.18181818181818182|2024-03-10 14:36:40|
+----------+---------------+-----------------+-------------+--------------------
+-----------------+------------------+----------+---------+----------------+----
------+--------------------+-------------------+
only showing top 3 rows
```

[9]:
```
#verifica o número total de linhas
print(f"Total number of rows: {df_redfin.count()}")
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…

Total number of rows: 5245047
```

[10]:
```
from pyspark.sql.functions import isnull
#Conta valores nulos em cada coluna
#usamos uma list comprehension
```

```
null_counts = [df_redfin.where(isnull(col_name)).count() for col_name in␣
 ↪df_redfin.columns]
null_counts
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
[0, 0, 0, 0, 0, 6066, 69125, 5646, 413699, 334643, 69227, 36370, 0]
```

[11]:
```
#Exibe os resultados
for i, col_name in enumerate(df_redfin.columns):
    print(f"{col_name}: {null_counts[i]} null values")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
period_end: 0 null values
period_duration: 0 null values
city: 0 null values
state: 0 null values
property_type: 0 null values
median_sale_price: 6066 null values
median_ppsf: 69125 null values
homes_sold: 5646 null values
inventory: 413699 null values
months_of_supply: 334643 null values
median_dom: 69227 null values
sold_above_list: 36370 null values
last_updated: 0 null values
```

[12]:
```
# Verifica se há valores faltantes em todo o DataFrame
remaining_count = df_redfin.na.drop().count()

print(f"Number of missing rows: {df_redfin.count() - remaining_count}")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
Number of missing rows: 501546
```

[13]:
```
print(f"Total number of remaining rows: {remaining_count}")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
Total number of remaining rows: 4743501
```

[14]:
```
#remove NA e conta o número total de linhas restantes
df_redfin = df_redfin.na.drop()
print(f"Total number of rows: {df_redfin.count()}")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
Total number of rows: 4743501
```

[15]:
```
# Conta os valores nulos em cada coluna para confirmar se removemos todos os␣
↪na(tem q ficar tudo zero)
null_counts = [df_redfin.where(isnull(col_name)).count() for col_name in␣
↪df_redfin.columns]
null_counts
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

[16]:
```
from pyspark.sql.functions import year, month

#Extrai o ano de period_end e salva em uma nova coluna "period_end_yr"
df_redfin = df_redfin.withColumn("period_end_yr", year(col("period_end")))

#Extrai mês de period_end e salva em uma nova coluna "period_end_month"
df_redfin = df_redfin.withColumn("period_end_month", month(col("period_end")))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[17]:
```
# Elimine as colunas period_end e last_updated
df_redfin = df_redfin.drop("period_end", "last_updated")
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[18]:
```
df_redfin.show(3)
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…

+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
|period_duration|             city|        state|
property_type|median_sale_price|
median_ppsf|homes_sold|inventory|months_of_supply|median_dom|
sold_above_list|period_end_yr|period_end_month|
+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
|             30|        Fair Lawn|   New Jersey|Single Family Res…|
542500.0| 278.1372476167421|        20|       69|             3.5|        91|
0.0|         2021|               2|
|             30|           Elyria|         Ohio|Multi-Family (2-4…|
30000.0|18.610421836228287|         3|        7|             2.3|       130|
0.0|         2019|               4|
|             30|Northwest Harwich|Massachusetts|Single Family Res…|
625000.0|340.65934065934067|        11|       14|             1.3|
26|0.18181818181818182|         2020|               8|
+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
only showing top 3 rows
```

[19]:
```python
from pyspark.sql.functions import when

#altera o número do mês para o respectivo nome do mês.

df_redfin = df_redfin.withColumn("period_end_month",
                   when(col("period_end_month") == 1, "January")
                    .when(col("period_end_month") == 2, "February")
                    .when(col("period_end_month") == 3, "March")
                    .when(col("period_end_month") == 4, "April")
                    .when(col("period_end_month") == 5, "May")
                    .when(col("period_end_month") == 6, "June")
                    .when(col("period_end_month") == 7, "July")
                    .when(col("period_end_month") == 8, "August")
                    .when(col("period_end_month") == 9, "September")
                    .when(col("period_end_month") == 10, "October")
                    .when(col("period_end_month") == 11, "November")
                    .when(col("period_end_month") == 12, "December")
                    .otherwise("Unknown")
                  )
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[20]:
```python
df_redfin.show(3)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
|period_duration|             city|        state|
property_type|median_sale_price|
median_ppsf|homes_sold|inventory|months_of_supply|median_dom|
sold_above_list|period_end_yr|period_end_month|
+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
|             30|        Fair Lawn|   New Jersey|Single Family Res…|
542500.0| 278.1372476167421|        20|       69|             3.5|        91|
0.0|         2021|        February|
|             30|           Elyria|         Ohio|Multi-Family (2-4…|
30000.0|18.610421836228287|         3|        7|             2.3|       130|
0.0|         2019|           April|
|             30|Northwest Harwich|Massachusetts|Single Family Res…|
625000.0|340.65934065934067|        11|       14|             1.3|
26|0.18181818181818182|         2020|          August|
+---------------+-----------------+-------------+--------------------+----------
-------+------------------+----------+---------+----------------+----------+----
---------------+-------------+----------------+
only showing top 3 rows
```

[22]:
```python
#salva dataframe final transformado no outro bucket s3 como um arquivo parquet.
s3_bucket = "s3://emr-camila-transformed-data/redfin_data.parquet"
df_redfin.write.mode("overwrite").parquet(s3_bucket)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```