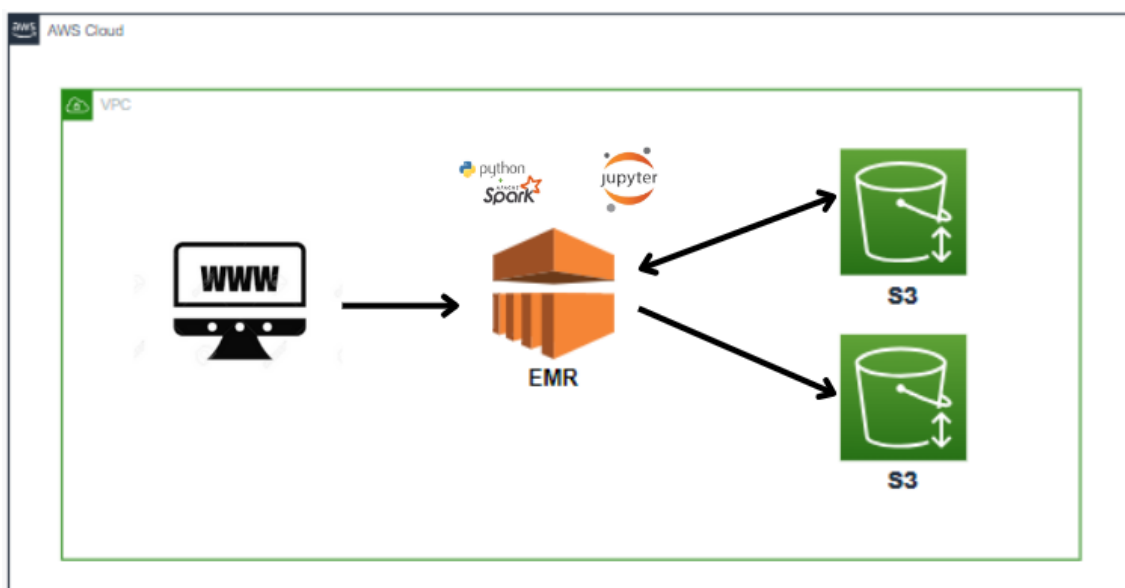


# Criação de Pipeline ETL em clusters EMR utilizando PySpark, Jupyter Notebook e S3 Buckets

## 1.Objetivo

Neste projeto construí um pipeline de dados na nuvem AWS, desde o zero, incluindo configurações de ambiente, criação de usuários, redes, subredes, EC2, grupos de segurança, clusters EMR, criação dos buckets S3, transformação e limpeza dos dados brutos, utilizando PySpark, até a disponibilização da base contendo os dados prontos para uso destinada ao usuário final. A fonte de dados é o site da Redfin, uma das maiores imobiliárias dos EUA e Canadá, que disponibiliza livremente dados do mercado imobiliário para áreas metropolitanas, cidades, bairros e códigos postais. Abaixo seguem descritos o passo a passo de cada etapa.

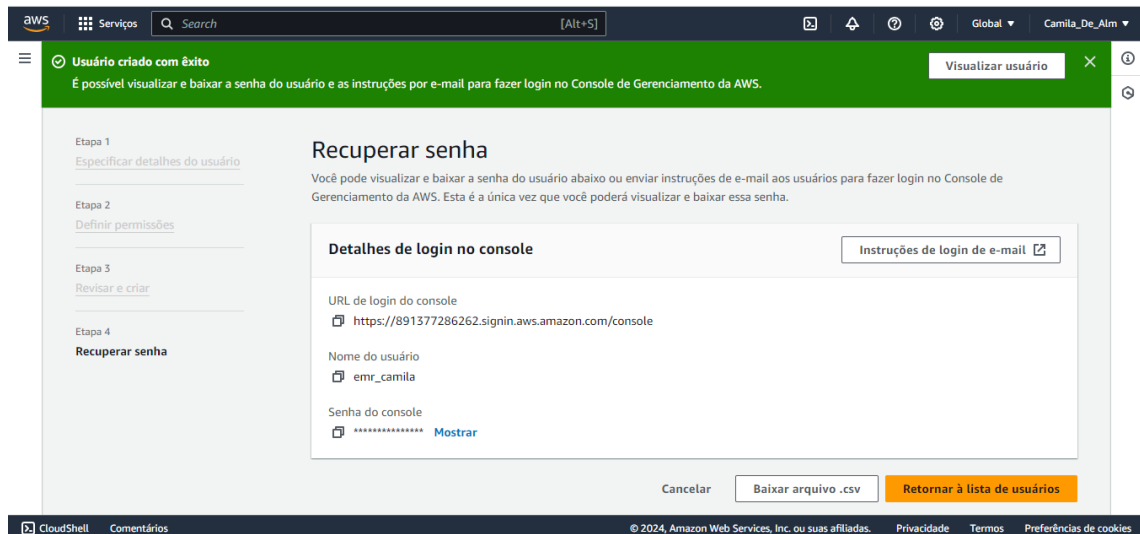
Demonstração do fluxograma final da arquitetura na AWS:



## 2.Criação do usuário Administrador

Será necessário ter uma conta ativa na AWS – Amazon World Services para execução dos itens a seguir. Devido a questões de segurança o usuário root, que é o usuário responsável pela criação da conta AWS, deverá criar um outro usuário administrador para que em caso de invasão da conta, ela se mantenha segura. Para este projeto o usuário administrador foi criado no AWS IAM (Identity and Access Management) que é um serviço da AWS que ajuda a controlar o acesso aos recursos da AWS de forma segura. Logada como usuário root, criei as credenciais e

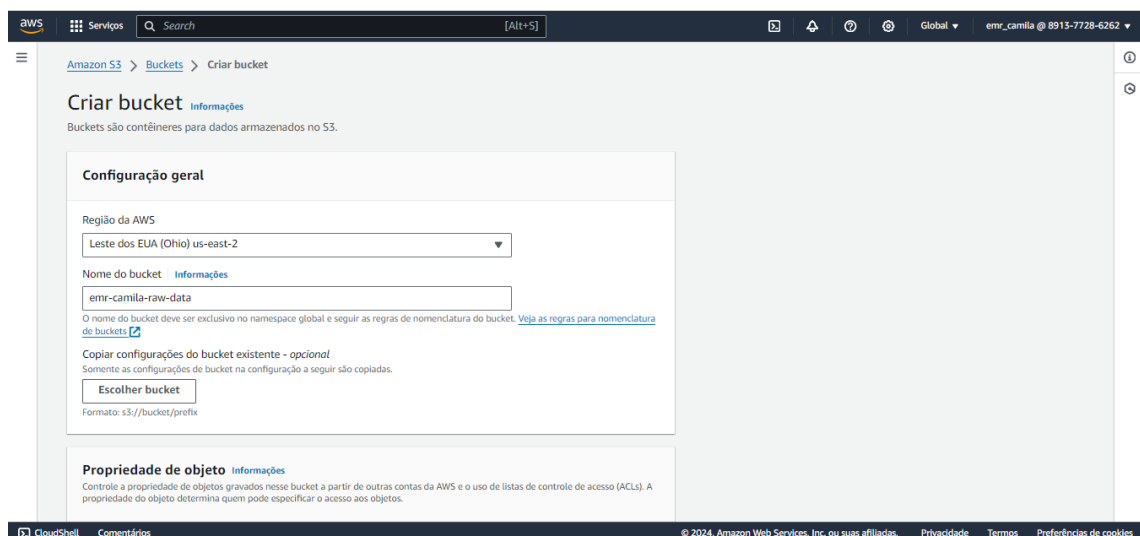
keypairs do usuário administrador, além disso habilitei as permissões necessárias para utilizar o EMR e os Buckets S3.



Por questões de segurança, também fiz o download das credenciais em csv, para poder recuperar no futuro, em caso de perda ou esquecimento de senha. Após seguir as práticas de segurança recomendadas pela AWS, fiz o log off do usuário Root e fiz log in no novo usuário Administrador.

### 3. Criação dos Buckets no AWS S3

Através do usuário Administrador fiz a criação de dois buckets S3, o primeiro para receber a extração dos dados da fonte localizada no site Redfin e o segundo bucket para receber os dados transformados e limpos. No console AWS, pesquise por S3, após criei o bucket. O nome do bucket deve ser exclusivo globalmente e seguir as regras de nomenclatura determinadas pela AWS, conforme imagem abaixo.



A região escolhida foi a us-east dos EUA, devido ao custo ser o mais recomendado, mas devido a latência, é recomendado que a escolha da Região seja a mesma

aonde o seu usuário está localizado. Além disso, é possível determinar propriedades, tags e versionamento, mas mantive as opções padrão sugeridas pela AWS. Para o segundo bucket, segui o mesmo procedimento acima.

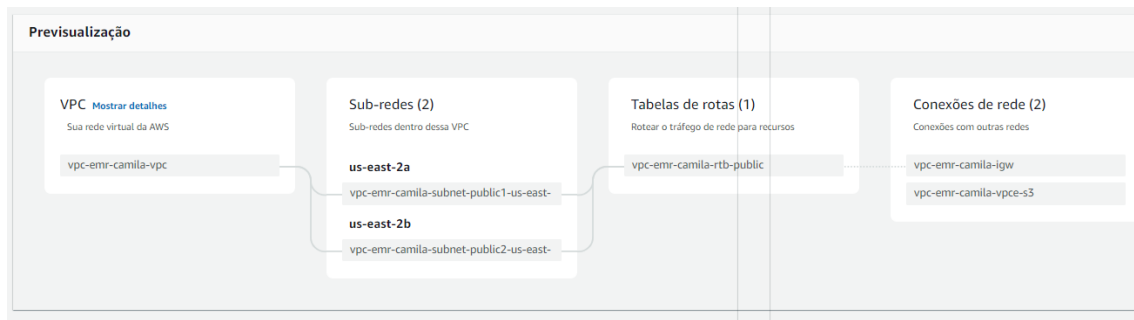
The screenshot shows the AWS 'Criar bucket' (Create bucket) page. The 'Configuração geral' (General configuration) section is active, showing the 'Região da AWS' (AWS Region) dropdown set to 'Leste dos EUA (Ohio) us-east-2'. The 'Nome do bucket' (Bucket name) field contains 'emr-camila-raw-data'. Below this, there is a link to 'Veja as regras para nomenclatura de buckets' and a button labeled 'Copiar configurações do bucket existente - opcional' (Copy bucket configuration - optional). The 'Propriedade de objeto' (Object property) section is partially visible at the bottom, with a note about ACLs.

## 4.Criação da VPC e Subredes

A VPC é uma Virtual Private Cloud, ou seja, uma nuvem privada virtual, ela cria uma rede privada virtual isolada na AWS, protegendo meu cluster EMR e meus dados contra acesso não autorizado e criptografa os dados em repouso e em trânsito entre o Amazon S3 e o Amazon EMR para garantir a confidencialidade das informações. No console, pesquise por VPC, depois clique em “Criar VPC”. Escolhi um nome para VPC, devemos criar a VPC, subredes, tabelas de rotas e determinar as zonas de disponibilidade, gateway e o endpoint. As zonas de disponibilidade, configurei como duas, pois caso haja algum desastre e uma região se torne indisponível, o trânsito de dados será redirecionado para a segunda, setei o mesmo valor para as redes públicas e nenhuma para as redes privadas, porque serão dados internos. O endpoint será o Gateway S3.

This is an identical screenshot of the AWS 'Criar bucket' page as shown above. It displays the 'Configuração geral' section with the bucket name 'emr-camila-raw-data' and the region 'Leste dos EUA (Ohio) us-east-2'. The 'Propriedade de objeto' section is also visible at the bottom.

A seguir, visualização de todo o fluxo da VPC.



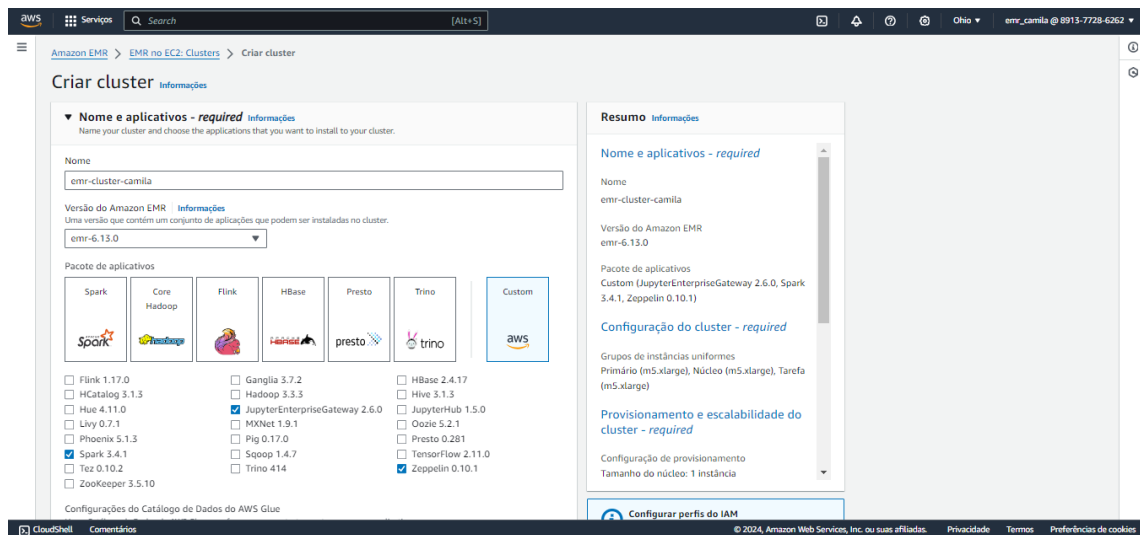
## 5. Clusters EMR

Após configuração de usuário, ambiente e segurança, criei o cluster AWS EMR (Elastic Map Reduce) que é um serviço da AWS que facilita a execução de frameworks de big data, como Apache Hadoop e Apache Spark, para processar e analisar grandes volumes de dados, neste projeto utilizei o Apache Spark.

Nos clusters EMR da AWS, os nós são instâncias do EC2 que se unem para fornecer a capacidade de processamento e armazenamento necessária para executar os jobs de big data. Existem três tipos principais de nós em um cluster EMR:

- **Node ou Nó primário (Primary Node):** É responsável por coordenar todo o trabalho no cluster, gerenciando tarefas, monitorando a saúde do cluster e rastreando o status dos jobs.
- **Nós centrais (Core Nodes):** Eles executam tarefas de processamento de dados e pode haver vários nós centrais para distribuir e executar jobs em paralelo.
- **Nós de tarefa (Task Nodes) (Opcional):** Fornecem capacidade de processamento adicional para executar tarefas intensivas em CPU.

Neste projeto, criei o Nó primário e o Nó secundário, devido a frequência e tamanho da carga. O EMR já vem com principais frameworks, por isso pode ser configurado, diretamente no console, o que diminui o custo. O setup escolhido foi a versão 6.13.0, não escolhi a mais recente, selecionei os aplicativos Spark, Jupyter e Zeppelin, conforme abaixo.



O grupo de instâncias escolhido foi o uniforme, para usar AWS EC2 sob demanda ou spot, devido a economia de escala, o tipo escolhido foi o M5.xlarge, que contém 4v core e 16Gib de memória, para o núcleo escolhi as mesmas configurações e removi a opção de ter um cluster de task, porque ele é opcional. As instâncias M5 do Amazon EC2 são alimentadas pelos processadores Intel Xeon Scalable mais rápidos da nuvem, com uma frequência turbo "all-core" de até 4,5 GHz. Além disso, as instâncias M5zn apresentam capacidade de rede de 100 Gbps, sendo ideais para aplicativos de computação intensiva e de alto consumo de recursos de rede, mas a instância pode ser alterada a depender do tamanho da carga, por medidas de desempenho escolhi a M5.Xlarge.

### Informações

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



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](#)

○

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)

Ações ▼

☐ 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)

### Remover grupo de instâncias

Ações ▼

Para a seção “Provisionamento e escalabilidade do cluster”, sobre a escalabilidade escolhi a opção gerenciado pelo EMR, porque assim o EMR se adapta conforme a variação do fluxo de dados. O Tamanho mínimo de cluster(número de instâncias) foram três, porque uma fica para o Nó Principal e a segunda e terceira ficam para o Nó Central. Quando a demanda estiver alta o número máximo do cluster será de 10 instâncias EC2, já para o Nó Central serão 8. Configuração de provisionamento ficou para duas, que seria a capacidade inicial da instância do núcleo.

▼ **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  instância(s)      Tamanho máximo do cluster  instância(s)

Máximo de nós centrais no cluster  
Limite o número de nós centrais em seu cluster.  
 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.  
 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	<input type="text" value="2"/>	<input type="checkbox"/>

Na seção, “Redes”, selecionei a VPC que havia criado anteriormente, pesquisei em navegar, selecionei a VPC e automaticamente o AWS preenche com as informações, da VPC e das Subredes.

▼ **Redes - *required*** [Informações](#)  
Choose the network settings that determine how you and other entities communicate with your cluster.

Nuvem privada virtual (VPC) [Informações](#)  
 [Navegar](#) [Criar VPC](#)

Sub-rede [Informações](#)  
 [Navegar](#) [Criar sub-rede](#)

► Grupos de segurança do EC2 (firewall)

No tópico “Encerramento de cluster e substituição de nó”, selecionei “Encerrar automaticamente o cluster após o tempo de inatividade”, porque é uma forma de reduzir custo, assim ele será encerrado em momentos de inatividade. Selecionei a opção “Usar proteção contra encerramento” se refere as EC2, assim deleta os clusters, mas mantém os dados armazenados.

▼ Cluster termination and node replacement [Informações](#)

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.

Na seção, “Cluster logs”, permiti que a AWS criasse um bucket para armazenamento dos logs dos clusters EMR.

▼ Cluster logs [Informações](#)

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

✕

Exibir [🔗](#)

Navegar no S3

Formato: usar s3://bucket/prefixo

☐ Criptografar logs específicos do cluster

Sobre a “Configuração de segurança e par de chaves do EC2” a prática recomendada é criar uma keypair para acessar via SSH, no entanto eu criei diretamente pelo console. Clique no botão criar par de chaves, abriu outra tab, dei um nome, escolhi o par RSA e o formato .pem, salvei o arquivo gerado.



Após selecione, a keypair, na configuração de segurança conforme abaixo.

Na seção “Perfis do IAM”, selecionei a opção escolha um perfil de serviço, por isso criei a VPC anteriormente, selecionei o grupo de segurança default cada vez que criamos um VPC um grupo de segurança é criado automaticamente.

### ▼ 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 X  
vpc-0a78b076a8755ae72

##### Sub-rede

Escolha uma ou mais sub-redes

vpc-emr-camila-subnet-public1-us-east-2a X  
subnet-0d794e8e8058361f9

##### Grupo de segurança

Escolha um ou mais grupos de segurança

default X  
sg-0a3ac6005ed529794

Após na seção “Perfil de instância”, escolhi um perfil de instância, dei acesso a todos os buckets para leitura e gravação, por isso os buckets foram criados antes dos 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



[Criar perfil do IAM](#)

Após cliquei em “Criar cluster” o que pode levar até 7 minutos para finalizar.

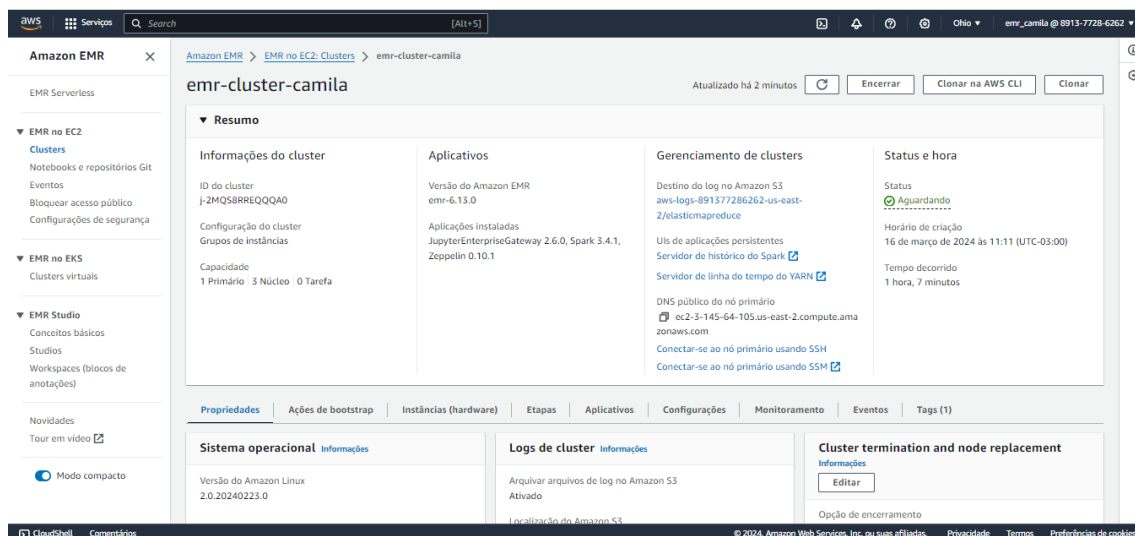
The screenshot shows the AWS Management Console for an Amazon EMR cluster named 'emr-cluster-camila'. The cluster is in the 'Starting' state. The console displays various tabs like 'Resumo', 'Propriedades', 'Ações de bootstrap', 'Instâncias (hardware)', 'Etapas', 'Aplicativos', 'Configurações', 'Monitoramento', 'Eventos', and 'Tags (1)'. The 'Resumo' tab is active, showing cluster information, applications, and logs.

Informações do cluster	Aplicativos	Gerenciamento de clusters	Status e hora
<b>ID do cluster</b> j-2MQS8RREQQQA0	<b>Versão do Amazon EMR</b> emr-6.13.0	<b>Destino do log no Amazon S3</b> aws-logs-891377286262-us-east-2/elasticmapreduce	<b>Status</b> Iniciando
<b>Configuração do cluster</b> Grupos de instâncias	<b>Aplicações instaladas</b> JupyterEnterpriseGateway 2.6.0, Spark 3.4.1, Zeppelin 0.10.1	<b>DNS público do nó primário</b> -	<b>Horário de criação</b> 16 de março de 2024 às 11:11 (UTC-03:00)
<b>Capacidade</b> 1 Primário 2 Núcleo 0 Tarefa			<b>Tempo decorrido</b> 2 minutos, 6 segundos

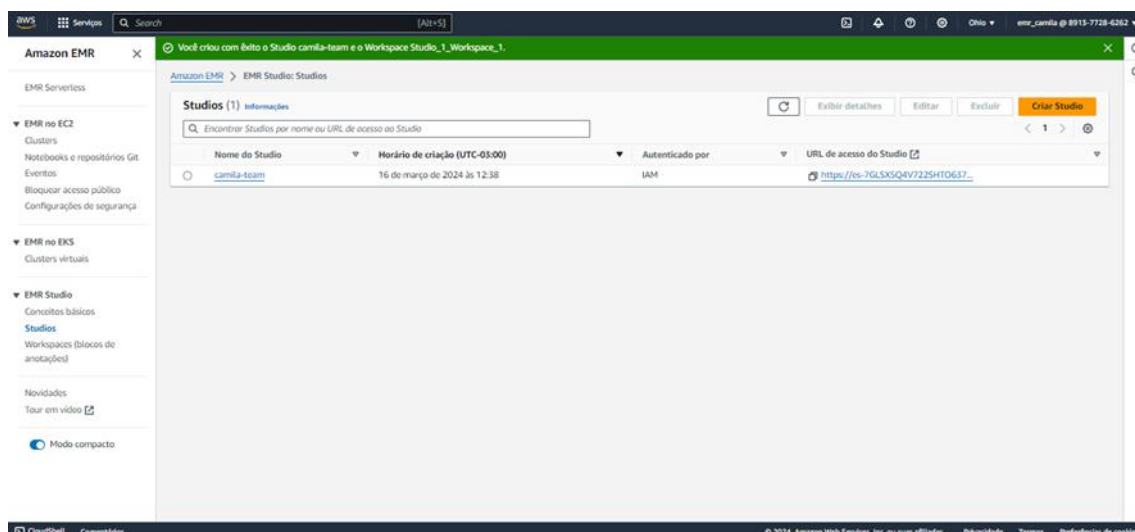
The console also shows a 'Logs de cluster' section with a table of logs and a 'Cluster termination and node replacement' section with a table of settings.

## 6. Configuração do Jupyter Notebook para utilizar o PySpark

Agora é possível configurar o Jupyter Notebook para utilizar o PySpark, para isso temos que configurar um Studio. À esquerda, em seção EMR Studio, clique em conceitos básicos, imagem abaixo. Depois em criar studio, um bucket foi criado para poder anexar o Jupyter ao EMR, assim os códigos escritos no notebook, serão executados no EMR e no bucket S3 via programação. Para isso, precisei criar um bucket personalizado e atribuir para o meu usuário, primeiro dei um nome, após criei um bucket só para o meu studio e associei ao meu usuário administrador.



Após, na seção de “Redes e segurança” selecionei a VPC e a Subrede que já havia criado, lembrando que o usuário administrador deve ter a permissão de full acesso ao AWS S3 e ser Usuário Administrador, conforme mencionado no passo dois, após clique em “Criar Studio”. Agora podemos criar o Jupyter notebook, clique na URL do Studio criado, conforme abaixo.



Cliquei em “Criar espaço de trabalho” escolhi um nome, as demais configurações seguem as mesmas para rede e armazenamento.

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, hífens 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 que já havia criado.

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

vpc-0a78b076a8755ae72

Sub-redes

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)

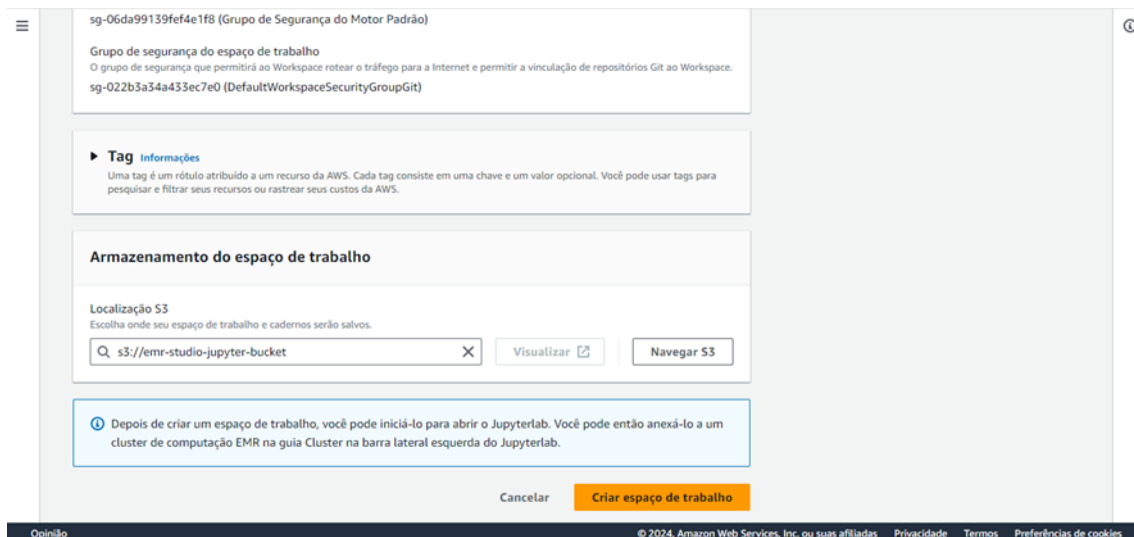
► Tag

Informações

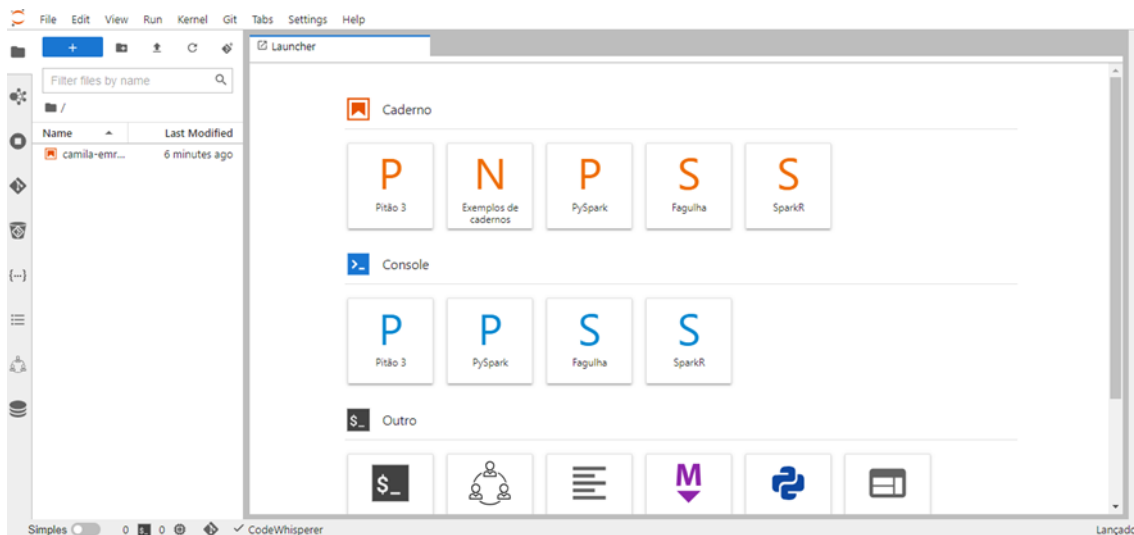
Uma tag é um rótulo atribuído a um recurso da AWS. Cada tag consiste em uma chave e um valor opcional. Você pode usar tags para pesquisar e filtrar seus recursos ou rastrear seus custos da AWS.

Armazenamento do espaço de trabalho

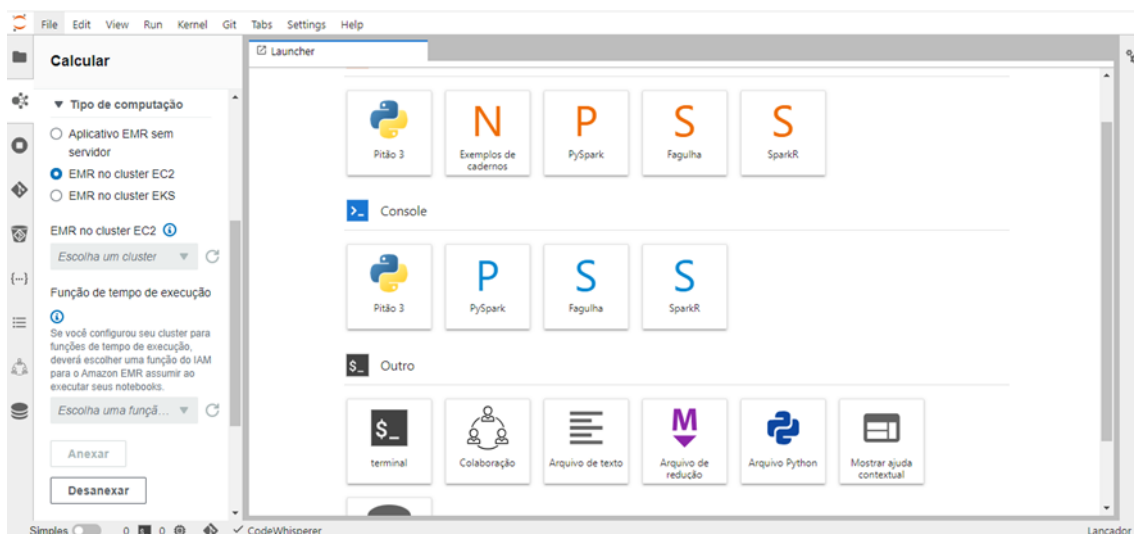
Bucket S3 criado especificamente para o Jupyter.



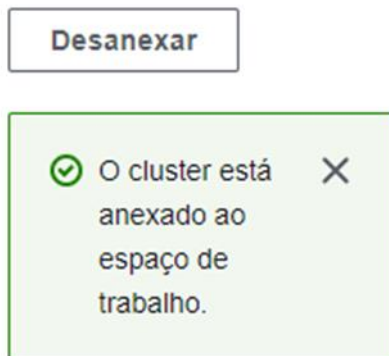
Cliquei no Studio abaixo, e ele abriu o Jupyter Notebook em uma nova aba.



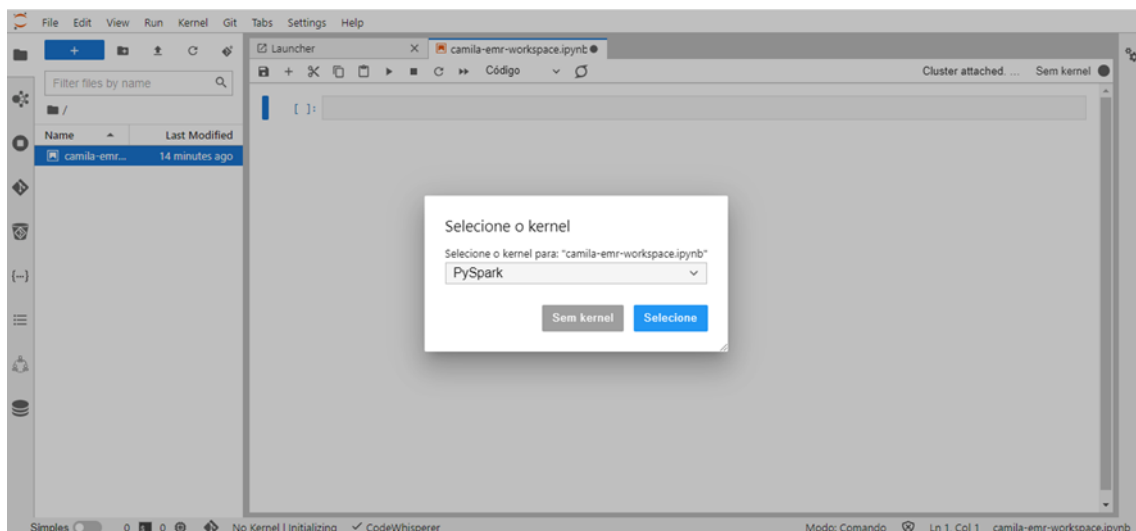
Agora é necessário anexar o Jupyter ao cluster que criamos e devemos selecionar EC2.



Atualizei a página e apareceu a mensagem abaixo em “Computação EMR”.

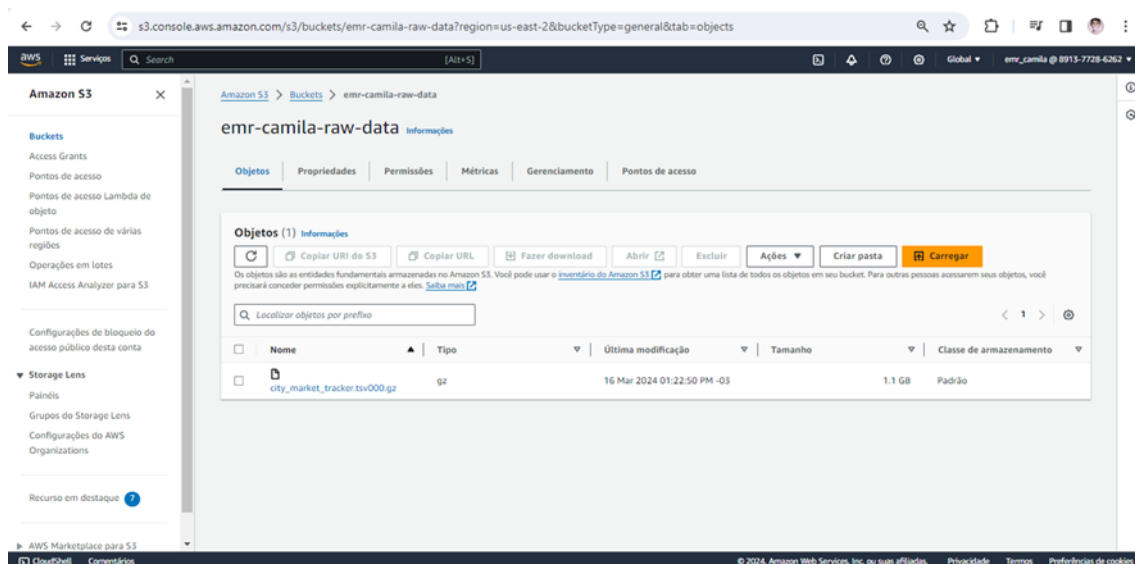


Após, voltei no meu workspace, cliquei duas vezes no caderno e escolhi o kernel como Pyspark.

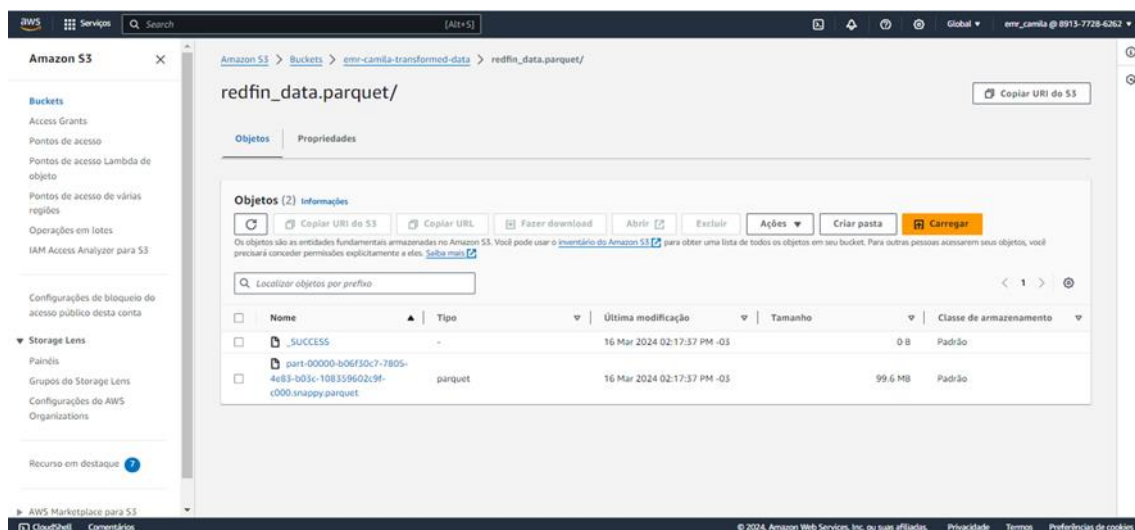


## 7. Explicação do código

Na primeira célula, importamos o Spark, na segunda, realizei o carregamento direto da página da Redfin e armazenei no primeiro bucket criado, o emr-camila-raw-data, o bucket que estava vazio, já recebeu a carga, conforme abaixo.



Após processar e limpar, salvei os dados transformados no segundo bucket que criei, conforme abaixo, temos o arquivo e o log de sucesso. O arquivo foi salvo em Parquet por causa do tamanho, mas poderia ter sido em csv, mas visando o melhor armazenamento e menor custo, escolhi a extensão Parquet.



## 8. Jupyter Notebook completo

A seguir, anexe o Jupyter Notebook com todos os códigos, transformações e alterações, as células possuem código comentado. Importante lembrar que a base disponibilizada pela RedFin, estava em sua maior parte limpa e organizada, no entanto, ajuste de colunas, verificação de valores NA, Null e mudança de nomes de variáveis foram aplicadas. Todo o embasamento foi retirado do site da AWS e a ideia para projeto foi retirada do canal Telespectra.