

Atividade prática na AWS

Objetivo

Implementar um cluster para processamento distribuído de dados utilizando o serviço AWS EMR com Hadoop MapReduce para contar palavras em um arquivo de texto armazenado no AWS S3 através de um algoritmo em Python. Para entender a lógica do MapReduce, recomendo a revisão do vídeo: <https://youtu.be/43fqzaSH0CQ>.

Material a ser entregue

Como não foi especificado qual material deve ser entregue neste desafio, optei por apresentar como produto deste desafio:

- Este tutorial, apresentando um passo a passo para replicação do mesmo
- Cópia do arquivo com a contagem do número de vezes que cada palavra aparece no texto

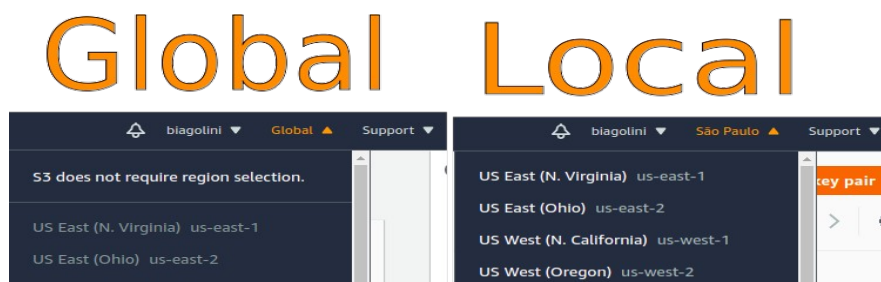
Requisitos

Para este tutorial, será preciso:

- Acesso a uma máquina com SO Linux (aqui utilizei o Ubuntu 20.04)
- Conta ativa na AWS. Para tal, você vai precisar ter um cartão de crédito e um número de telefone ativo para concluir o cadastro.

Sobre a AWS

Tenha em mente que a AWS possui servidores espalhados por todo mundo. Alguns serviços, exigem a especificação de local/servidor, outros não. Para este desafio, optei por sempre que requerido usar o servidor de São Paulo (que é identificado pela sigla de sa-east-1).



Passo-a-passo

Passo 1: Entre no AWS Management Console: <https://console.aws.amazon.com/>

Passo 2: Faça login como IAM user.



Sign in

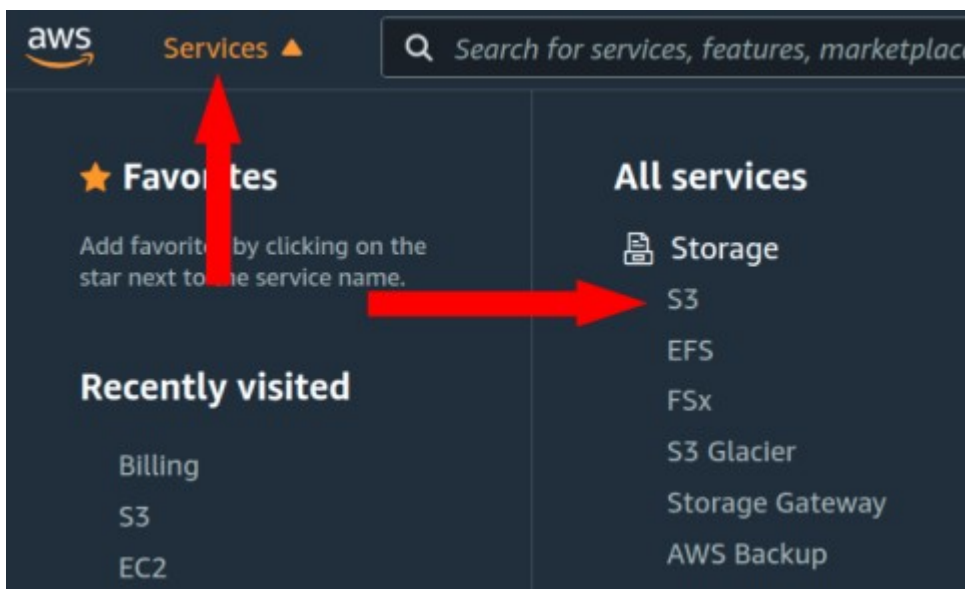
☐ Root user

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

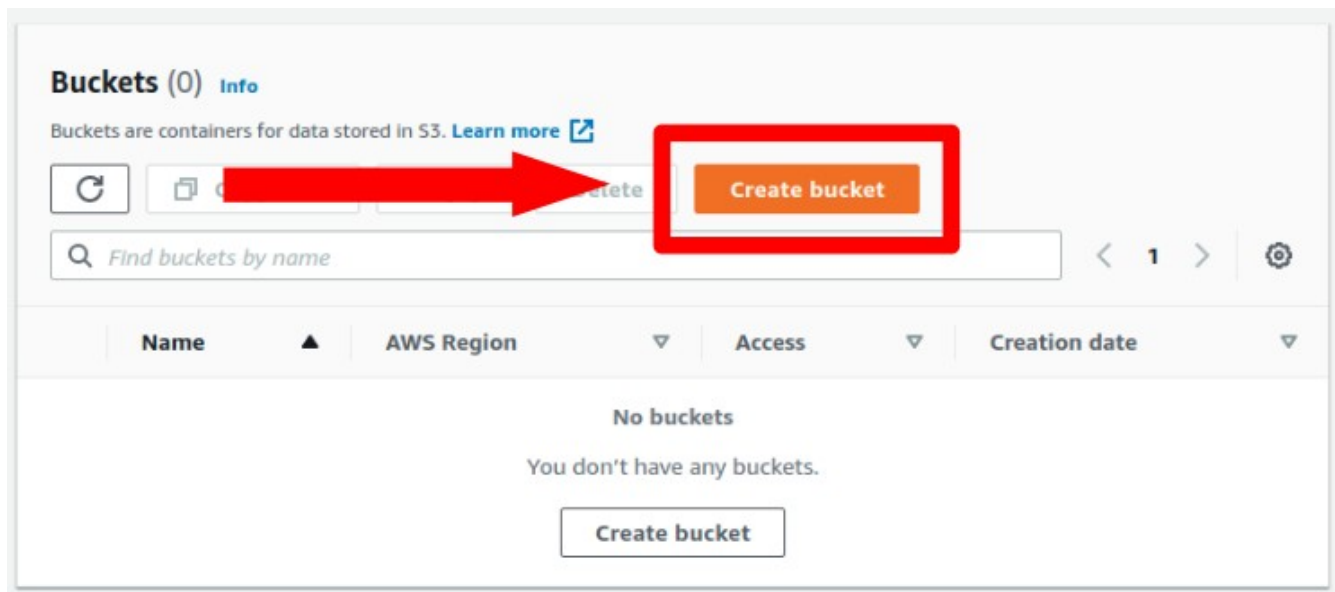
☒ IAM user

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

Passo 3: Clique em SERVICES, procure a secção STORAGE, e entre em S3.



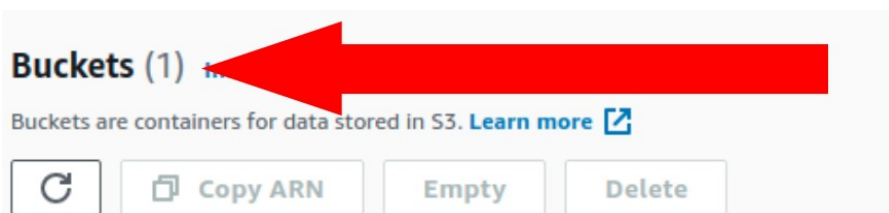
Passo 4: Clique em CREATE BUCKET



Passo 5: Escolha um nome para o seu Bucket (eu escolhi o nome de **desafio-dio-aws-sp**), confirme qual é a região que você escolheu para seu Bucker (optei por usar **South America (São Paulo) sa-east-1**). Depois vá lá no final da página e clique em CREATE BUCKET. OBS: o nome do Bucket é único (i.e. se um usuário criou um bucket com um nome, você não poderá criar outro com o mesmo), o nome do bucket deve ser composto de 3 à 63 caracteres, sendo proibido letras maiúsculas. Mais informações sobre regras para nomear buckets em: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>

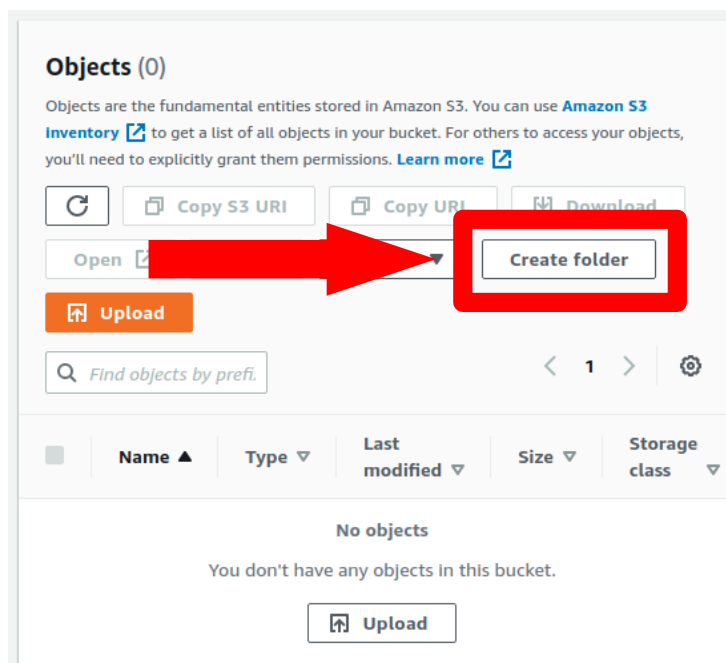
A screenshot of the 'Create bucket' configuration page in the AWS S3 console. The breadcrumb at the top is 'Amazon S3 > Create bucket'. The main heading is 'Create bucket Info'. Below it is the same text description as in the previous image. The 'General configuration' section contains two main fields: 'Bucket name' and 'AWS Region'. The 'Bucket name' field has the text 'desafio-dio-aws-sp' and a red arrow points to it from the right. Below this field is a note: 'Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming'. The 'AWS Region' field has the text 'South America (São Paulo) sa-east-1' and a red arrow points to it from the right. Below these fields is a section titled 'Copy settings from existing bucket - optional' with the subtext 'Only the bucket settings in the following configuration are copied.' and a 'Choose bucket' button.

Passo 6: Criado seu BUCKET, o sistema irá recarregar a página do serviço S3 (mesma página observada no passo 3). Note que agora o contador de Buckets passou a contar com mais 1 Bucket (se você não tinha nenhum Bucket previamente criado, você deve ter a indicação de 1 Bucket, como mostrado na figura abaixo). Na seção de lista de Buckets, você observará o Bucket que você acabou de criar, com o nome que você designou (no meu exemplo: “**desafio-dio-aws-sp**”), e a região que o Bucket está hospedado (no meu caso “**sa-east-1**”). Clique no nome do seu Bucket para acessar o mesmo, e siga para o próximo passo. Preste muita atenção no local onde você criou seu Bucket. Você vai precisar dessa informação mais adiante.



	Name ▾	AWS Region ▾	Access ▾	Creation date ▾
<input type="radio"/>	desafio-dio-aws-sp	South America (São Paulo) sa-east-1	Bucket and objects not public	September 2, 2021, 18:03:28 (UTC-03:00)

Passo 7: Neste passo, vamos criar 3 pastas que serão usadas para operacionalizar a exsução do nosso Script Python. Para tal, clique em CREATE FOLDER.



Passo 8: Crie 3 pastas, com os nomes de “data”, “output”, e “temp”. Repita o procedimento a seguir 3x. Indique o nome da pasta no campo FOLDER NAME, em seguida clique que CREATE FOLDER.

Folder

Folder name
data

Folder names can't contain "/"". [See rules for naming](#)

Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

Server-side encryption

☒ Disable

☐ Enable

Cancel Create folder

Passo 9: Criada as 3 pastas, acesse a pasta “data”, para que no próximo passo façamos o upload do arquivo de texto que será analisado.

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

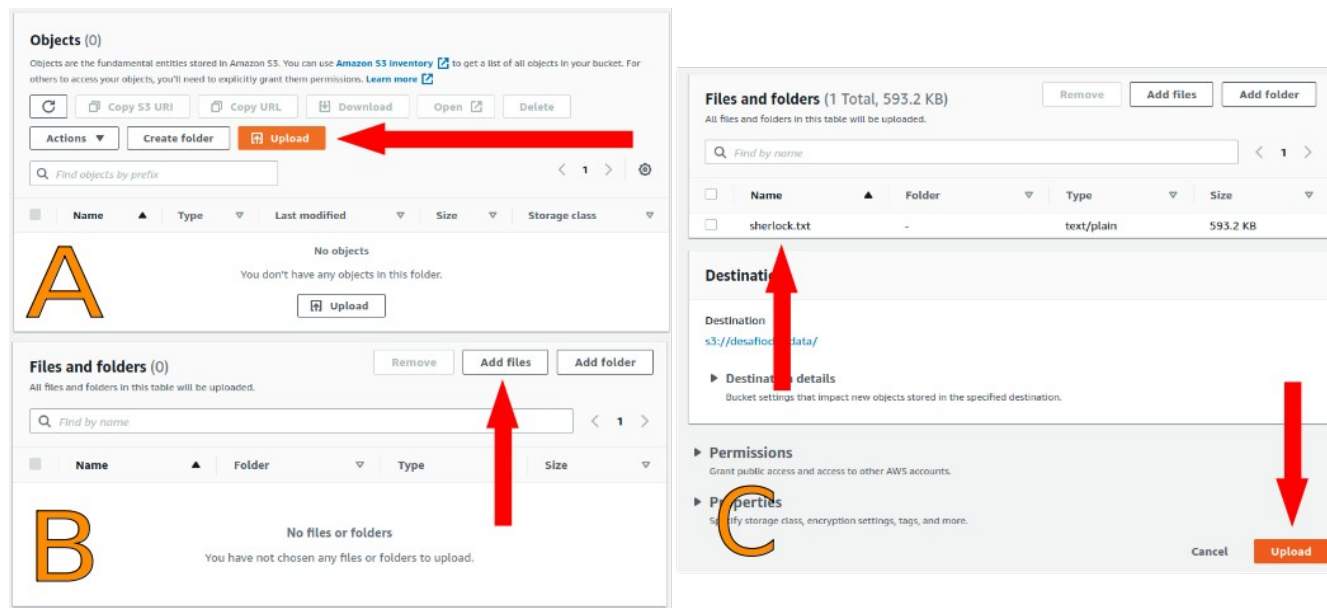
[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#)

[Delete](#) [Actions](#) [Create folder](#) [Upload](#)

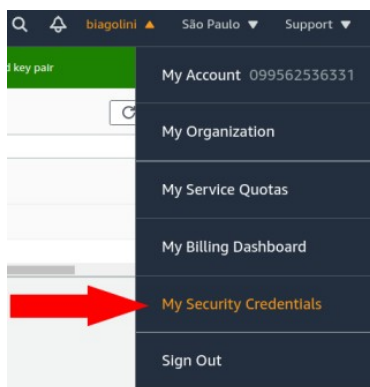
<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	data/				-
<input type="checkbox"/>	output/	Folder	-	-	-
<input type="checkbox"/>	temp/	Folder	-	-	-

Passo 10: Uma vez dentro da pasta “data”, clique em UPLOAD (secção A da figura a seguir). Em seguida, clique em ADD FILE (secção B da figura a seguir), e selecione o arquivo de texto. Este arquivo tem o nome de “sherlock.txt”, o mesmo está disponível em:

<https://github.com/cassianobrexbit/DIO-LiveCoding-AWS-BigData>. Ao final, você deve observar o arquivo “sherlock.txt” na lista de arquivos a serem copiados para pasta “data”, confirme o upload, clicando em UPLOAD no canto inferior a direita da janela (secção C da figura a seguir).



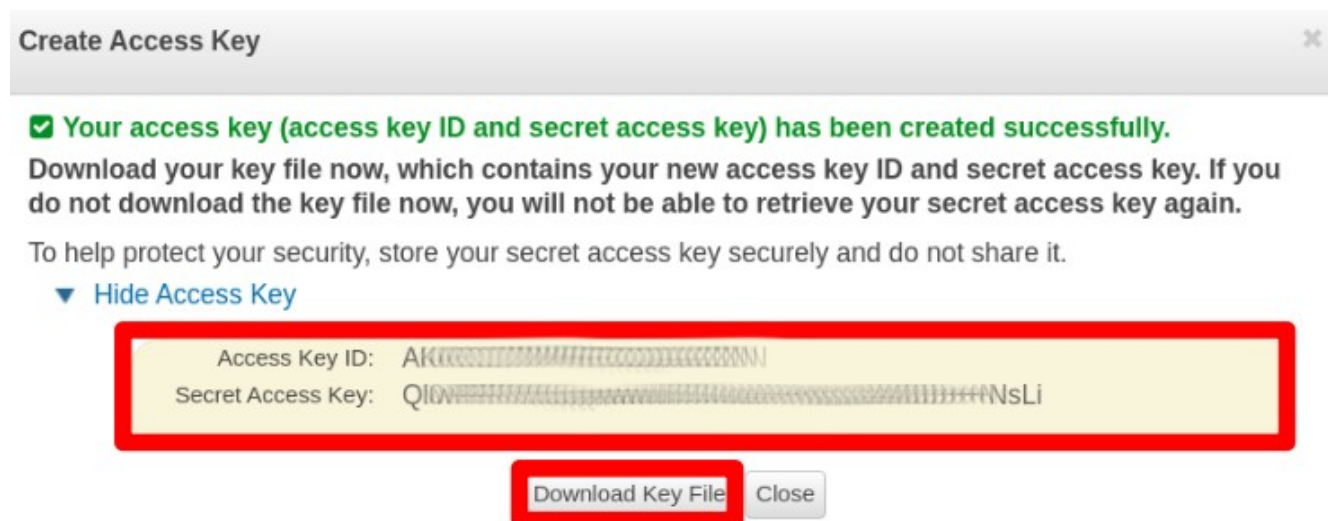
Passo 11: Criar uma credencial, que vai conter seu ID e uma chave secreta para acesso à sua conta AWS. Essa chave será fornecida para o pacote mrjob do Python, que fará o acesso a AWS. Que por sua vez, criará um cluster e executar o serviço. Para tal, clique no seu nome da sua conta (canto superior a direita no painel da AWS), depois clique em MY SECURITY CREDENTIALS.



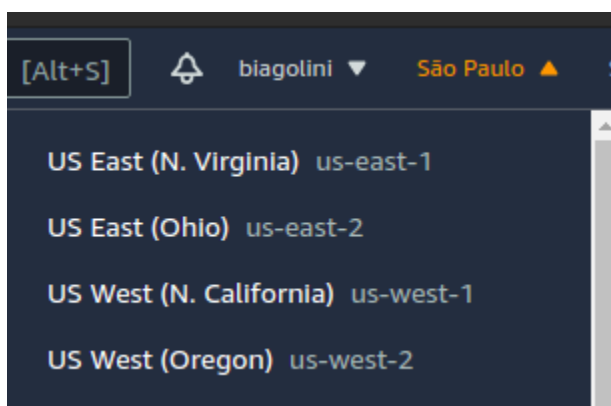
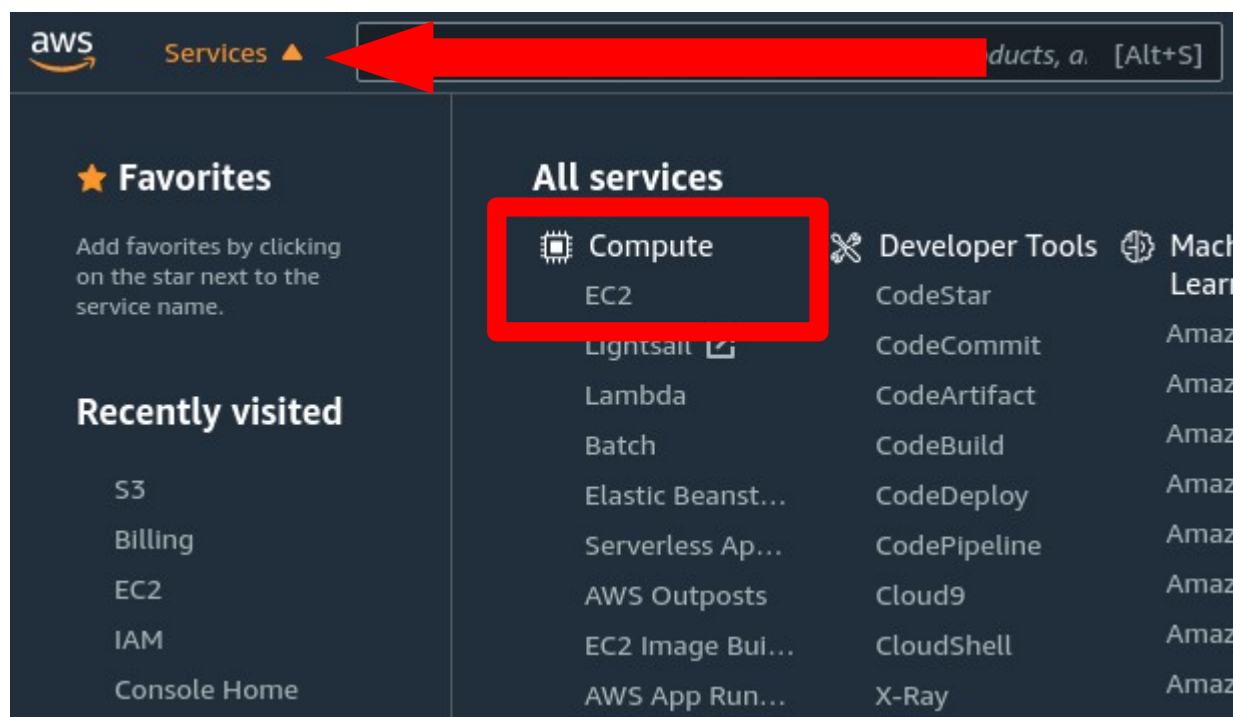
Passo 12: Localize a seção ACCESS KEYS (ACCESS KEY ID AND SECRET ACCESS KEY), e clique em CREATE NEW ACESS KEY. Se você já usa a AWS, e já tem 2 chaves, você não poderá criar novas chaves. Portanto, reutilize uma das suas chaves, ou apague uma das suas chaves e crie uma nova.



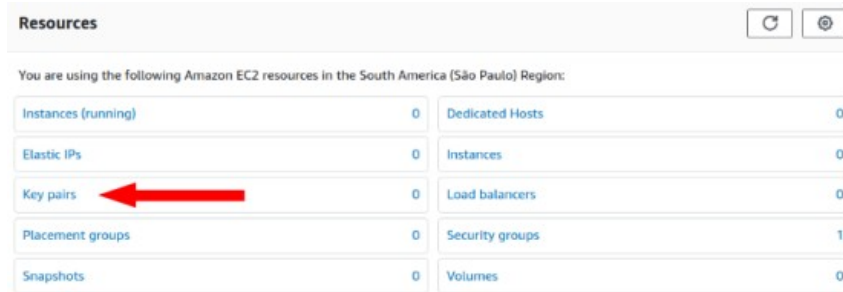
Passo 13: Uma vez que sua chave foi gerada, você pode visualizar a mesma, e/ou fazer download desta. Contudo, se você perder essa chave, a AWS não vai lhe fornecer essa mesma chave novamente. Então você terá que apagar essa chave terá que criar outra. Por padrão, quando você faz download da chave o arquivo enviado é chamado de “rootkey.csv”, este conterá 2 linhas de texto, com os mesmos dados que foram apresentados a você. OBS: essas chaves são muito importantes para sua segurança. Cuidado para não deixar elas perdidas. Assim, após esse tutorial, recomenda-se que você apague sua chave (assim como eu farei).



Passo 14: Agora, vamos criar chaves SSH para acessar as instâncias. Para tal, clicando em SERVICES, Localize a secção COMPUTE, e clique em EC2. O serviço EC2 é ligado à gestão de máquinas virtuais. Este serviço está restrito a região/servidor de onde você está fazendo suas operações. Portanto, após abrir este serviço, confira se no canto superior a direita, está selecionada a mesma região da qual você criou seu bucket (no passo 05). Caso não esteja, faça a modificação, para o local desejado.

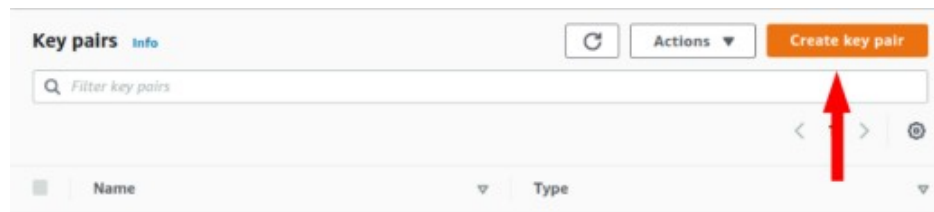


Passo 15: Entre em Key pairs

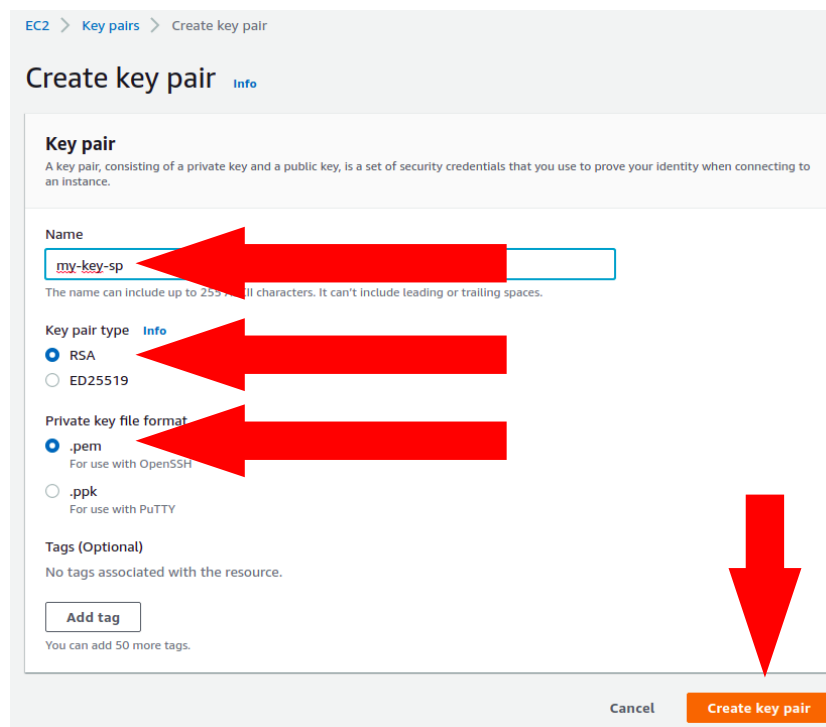


Resources	
You are using the following Amazon EC2 resources in the South America (São Paulo) Region:	
Instances (running)	0
Elastic IPs	0
Key pairs	0
Placement groups	0
Snapshots	0
Dedicated Hosts	0
Instances	0
Load balancers	0
Security groups	1
Volumes	0

Passo 16: Clique em CREATE KEY PAIR.



Passo 17: Crie uma chave com qualquer nome (escolhi “my-key-sp”), selecione o formato RSA (exclusivos para instâncias Linux e Mac), e formato .pem em segui confirme a criação dessa chave clicando em CREATE KEY PAIR. Automaticamente vai abrir uma janela para download dessa chave. Salve a chave em uma pasta segura do seu computador. Se você perder esse arquivo, vai precisar criar outro, mas pode criar quantos novos arquivos você quiser.



EC2 > Key pairs > Create key pair

Create key pair Info

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

The name can include up to 255 alphanumeric characters. It can't include leading or trailing spaces.

Key pair type Info
☒ RSA
☐ ED25519

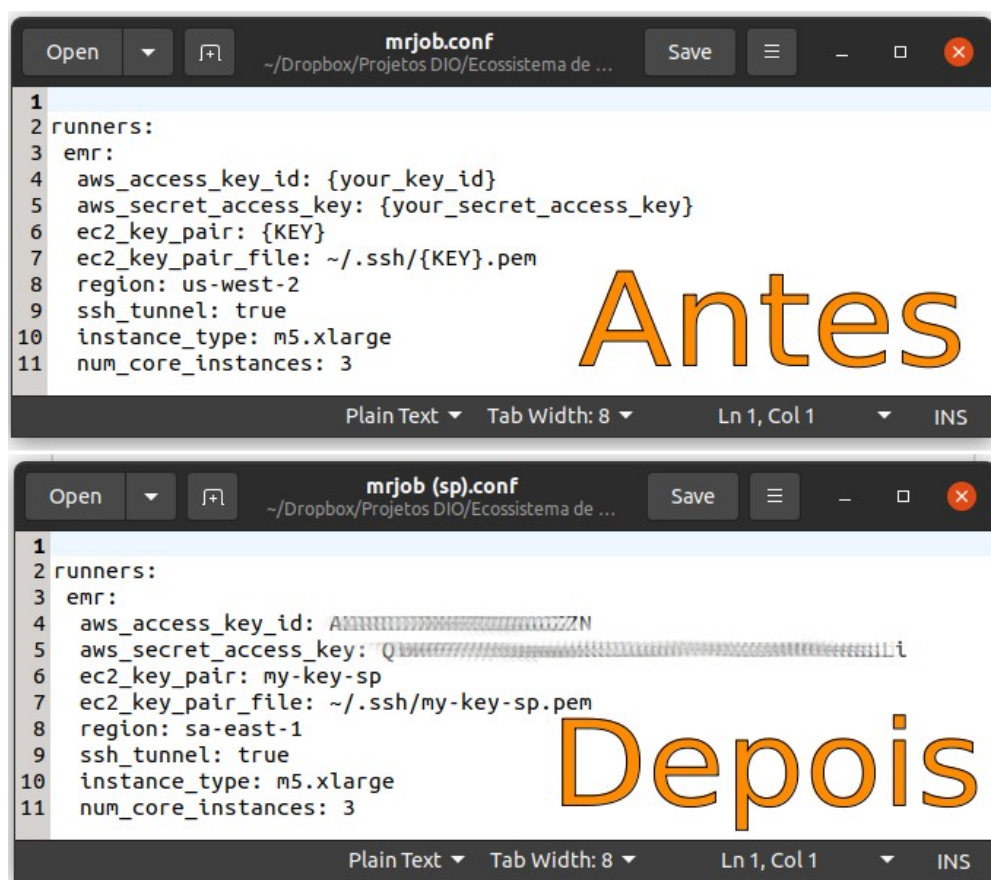
Private key file format
☒ .pem
For use with OpenSSH
☐ .ppk
For use with PuTTY

Tags (Optional)
No tags associated with the resource.

You can add 50 more tags.

Passo 18: Pegue o arquivo “mrjob.conf” (download também disponível no link <https://github.com/cassianobrexbit/DIO-LiveCoding-AWS-BigData>) e atualize as seções de acordo com a lista abaixo. Salve esse arquivo em uma pasta qualquer do seu computador. Aqui vamos usar esse arquivo apenas para ter tudo organizado, para num passo mais adiante copiar e colar o conteúdo deste documento em outro arquivo.

- *aws_access_key_id*: sua chave de acesso, gerada no passo 13. Se você fez download e não mudou nome do arquivo da chave, esta, estará salva em um arquivo chamado “rootkey.csv”
- *aws_secret_access_key*: sua chave de acesso, gerada no passo 13. Se você fez download e não mudou nome do arquivo da chave, esta, estará salva em um arquivo chamado “rootkey.csv”
- *ec2_key_pair*: deve receber o nome da sua chave gerada no passo 17 (no meu caso “minhachave”)
- *ec2_key_pair_file*: indicação do caminho para sua chave. Esse arquivo, nós vamos configurar isso mais a frente, por hora, entenda que você vai copiar sua chave para um arquivo oculto dentro da pasta .ssh no diretório do usuário da sua conta no Ubuntu. Sendo assim, no meu caso o caminho foi “~/.ssh/my-key-sp.pem”
- *region*: Essa parte é MUITO importante! As chaves ssh são sensíveis a região, portanto, aqui tudo deve estar batendo. No meu caso, hospedei meu cluster na região “**sa-east-1**”, como indicado ao longo desse tutorial.



Passo 19: Criar ambiente virtual Python. Neste tutorial, vou usar uma pasta na minha área de trabalho, porque apagarei tudo após a execução deste tutorial. Mas no caso de um projeto para “vida real”, escolha uma pasta mais apropriada. Para iniciar o terminal, utilize o talho: CTRL + ALT + T. Em seguida, vamos executar alguns comandos (apresentados a seguir) para criar o ambiente virtual, para mais informações sobre ambiente virtual, veja: <https://docs.python.org/pt-br/3/tutorial/venv.html>.

Criar a pasta de trabalho. OBS: dependendo do idioma de instalação e nome de usuário o caminho pasta de área de trabalho pode variar

```
mkdir /home/user/Desktop/desafioaws
```

Abri a pasta de trabalho que você acabou de criar

```
cd /home/user/Desktop/desafioaws
```

Criar ambiente python

```
python3 -m venv tutorial-env
```

Ativar o ambiente

```
source tutorial-env/bin/activate
```

Nessa etapa você vai ver que mudou o ambiente

A screenshot of a terminal window with a dark background. The title bar shows 'user@machine: ~/Desktop/desafioaws'. The terminal content shows the following commands and their outputs: 1. '(base) user@machine:~/Desktop\$ mkdir /home/user/Desktop/desafioaws' 2. '(base) user@machine:~/Desktop\$ cd /home/user/Desktop/desafioaws' 3. '(base) user@machine:~/Desktop/desafioaws\$ python3 -m venv tutorial-env' 4. '(base) user@machine:~/Desktop/desafioaws\$ source tutorial-env/bin/activate' 5. The prompt changes to '(tutorial-env) (base) user@machine:~/Desktop/desafioaws\$'.

Instalar pacote boto3, pacote do Python que fornece interfaces para acesso a AWS. <https://aws.amazon.com/pt/sdk-for-python/>

```
pip install boto3
```

Instalar biblioteca mrjob, uma biblioteca para escrever os seus Jobs (algoritmos) MapReduce. Para mais informações, recomendo a leitura de: <https://medium.com/data-hackers/processamento-distribu%C3%ADdo-de-dados-com-mapreduce-utilizando-python-mrjob-e-emr-c826a617f8b3>

```
pip install mrjob
```

Após a conclusão da instalação, você pode fechar o seu terminal.

Passo 20: Editar arquivos de configuração de mrjob e ssh pem. Para tal, abra outro terminal do Linux (CTRL + ALT + T), e siga os comandos:

Acessar a raiz do usuário:

```
cd ~
```

Veja a lista de arquivos e pastas (incluindo ocultos)

```
ls -a
```

Veja minha lista (a sua será diferente):

```
(base) user@machine:~$ ls -a
.          .gnupg          .r
..         .ipynb_checkpoints R
anaconda3 .ipython        .Rhistory
.apport-ignore.xml .java          snap
apps       .jupyter       .ssh
.bash_history .keras         .ssh
.bash_logout .local         .steam
.bashrc     .mozilla       .steampath
.cache     .mrjob.conf   .steampid
.conda     mrjob.conf    .sudo_as_admin_successful
.condarc   Music         Templates
.config    .pam_environment tensorflow_dir
Desktop    Pictures      .thunderbird
Documents  .pki         Videos
Downloads .profile     .vscode
.dropbox   Public       .wget-hsts
Dropbox    .pulse-cookie
.dropbox-dist .python_history
(base) user@machine:~$
```

Acesse o arquivo de configuração do mrjob (lembre de usar o . antes do nome do arquivo, porque queremos escrever sobre um arquivo oculto):

```
nano .mrjob.conf
```

Copie e cole os dados do arquivo que você editou no passo 18. Para acessar o arquivo de configuração, execute o comando. Após colar o texto, você usa o atalho CTRL + O, para sobrescrever o documento. Em seguida digite ENTER, confirmar a sobrescrita do arquivo. Por fim, use o atalho CTRL + X para fechar.

Crie/edite (dentro da pasta .ssh) um arquivo chamado NOME.pem, onde NOME é o mesmo que você usou no passo 17, e conseqüentemente usou no seu campo “ec2_key_pair_file” do mrjob. Sendo assim, para o meu caso, o comando é:

```
nano .ssh/my-key-sp.pem
```

Abra o arquivo .pem que você fez download no passo 17, com o editor de texto (bloco de notas do Linux). Copie e cole o texto (sua chave) dentro do editor nano. Após colar o texto, use os atalhos i) CTRL + O ; ii) ENTER,; iii) CTRL + X (mesma lógica da edição do mrjob).

Depois disso, você pode encerrar essa seção de terminal.

Passo 21: Se fosse na “vida real”, agora você escreveria seu script python com as atividades a serem executadas. Mas para esse tutorial o professor já deixou o script pronto, e disponibilizou para você fazer download. O nome do arquivo é “dio-live-wordcount-test.py”, link para download: <https://github.com/cassianobrexbit/DIO-LiveCoding-AWS-BigData> (OBS: usei o Script enviado no primeiro Commit do repositório). Faça download deste script e coloque o arquivo dentro da sua pasta onde está o ambiente Python (no meu caso /home/user/Desktop/desafioaws)

Passo 22: Inicie um terminal na pasta onde você salvou seu script Python (CTRL+ALT+T, e navegue até a pasta, ou use o botão direito e vá em “Abrir num terminal”/“Open in terminal”).

Passo 23: para inicializar o envio do seu código para o servidor, o comando resumido é o seguinte (calma, não copie e cole o comando sem antes de terminar de ler todo esse passo):

```
python3 dio-live-wordcount-test.py -r emr
s3://{your_s3_bucket_name}/data/{LIVRO}.txt
--output-dir=s3://{your_s3_bucket_name}/output/logs1
--cloud-tmp-dir=s3://{your_s3_bucket_name}/temp/
```

Substitua {your_s3_bucket_name} pelo nome do bucket que você escolheu no passo 5 (no meu caso foi “desafiobiagolini”). Substitua {LIVRO} pelo nome do arquivo de texto que você fez upload para contagem de palavras.

Sendo assim, meu código ficou:

```
python3 dio-live-wordcount-test.py -r emr
s3://desafio-dio-aws-sp/data/sherlock.txt
--output-dir=s3://desafio-dio-aws-sp/output/logs1 --cloud-tmp-dir=s3://desafio-dio-aws-sp/temp/
```

Passo 24: Aguarde alguns minutos para a execução do job. Após a conclusão, você verá uma mensagem confirmando a execução como a indica na figura a seguir.

```
job output is in s3://desafio-biagolini/output/logs1/  
Removing s3 temp directory s3://desafio-biagolini/temp/dio-live-wordcount-test.user.20210902.204221.375729/...  
Removing temp directory /tmp/dio-live-wordcount-test.user.20210902.204221.375729...  
Removing log files in s3://desafio-biagolini/temp/logs/j-HC3Z96KJACY/...  
Terminating cluster: j-HC3Z96KJACY  
(base) user@machine:~/Desktop/desafioaws$
```

Passo 25: Agora, na aba de serviços S3, você pode acessar a contagem do número de palavras indo até a pasta NOME_BUCKET/output/logs1. Cada um dos arquivos part-0000X detém parte da contagem do número de palavras.

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	_SUCCESS	-	September 2, 2021, 17:48:37 (UTC-03:00)	0 B	Standard
<input type="checkbox"/>	part-00000	-	September 2, 2021, 17:48:32 (UTC-03:00)	19.5 KB	Standard
<input type="checkbox"/>	part-00001	-	September 2, 2021, 17:48:37 (UTC-03:00)	19.6 KB	Standard
<input type="checkbox"/>	part-00002	-	September 2, 2021, 17:48:36 (UTC-03:00)	18.8 KB	Standard
<input type="checkbox"/>	part-00003	-	September 2, 2021, 17:48:37 (UTC-03:00)	19.2 KB	Standard
<input type="checkbox"/>	part-00004	-	September 2, 2021, 17:48:36 (UTC-03:00)	19.8 KB	Standard

Passo 26: Para fazer download do arquivo, selecione o arquivo desejado, clique em ACTIONS, e em seguida DOWNLOAD AS.

The screenshot shows the Amazon S3 console interface for the bucket 'logs1/'. The 'Objects' tab is selected, showing a list of objects. The object 'part-00000' is selected. A red arrow points from the 'Actions' button to the 'Download as' option in the dropdown menu.

Name	Type	Last modified
_SUCCESS	-	September 2, 2021, 17:48:37 (UTC-03:00)
part-00000	-	September 2, 2021, 17:48:32 (UTC-03:00)
part-00001	-	September 2, 2021, 17:48:37 (UTC-03:00)
part-00002	-	September 2, 2021, 17:48:36 (UTC-03:00)
part-00003	-	September 2, 2021, 17:48:37 (UTC-03:00)
part-00004	-	September 2, 2021, 17:48:36 (UTC-03:00)