

VINÍCIUS HENRIQUE DOS SANTOS

POSTECH

MACHINE LEARNING ENGINEERING

BIG DATA PIPELINES

# AULA 02

---

## SUMÁRIO

O QUE VEM POR AÍ? .....	3
HANDS ON .....	4
SAIBA MAIS .....	5
O QUE VOCÊ VIU NESTA AULA? .....	30
REFERÊNCIAS.....	31

EMSE

## O QUE VEM POR AÍ?

Você está preparado(a) para mergulhar nas principais ferramentas do mercado e entender como elas funcionam e para que servem? Que tal se aventurar nos desafios dessa aula para dominar as principais técnicas de construção do seu pipeline em uma arquitetura local, cloud ou híbrida?

Nessa aula, você entenderá como funcionam mecanismos importantes, como Hadoop, Spark e Airflow, e construirá um projeto do zero, com todas as características do mundo real das grandes organizações.

## HANDS ON

Imagine que você está inserido(a) em um contexto que demanda processamento de volumes massivos de dados, que podem crescer cada vez mais, de forma pouco previsível. O que você faria?

Nessa aula, você entenderá as estratégias e técnicas para dominar ferramentas que te auxiliarão na construção de pipelines em batch para resolver os maiores desafios que aparecerem na sua trajetória profissional.

Para isso, você aprenderá como usar o Spark para processar seus dados e o funcionamento do Hadoop em uma arquitetura de cloud dentro da Amazon.

## SAIBA MAIS

### Instalando o apache AIRFLOW

O AirFlow foi desenvolvido em Python, mas algumas das bibliotecas necessárias para a instalação e funcionamento só podem ser executadas em ambientes Linux. Por conta disso, para usuários Windows, desenvolvi o passo a passo que nos permitirá virtualizar um ambiente Linux.

Primeiro, faça o download do virtualizador de sua preferência. Aqui, eu seguirei com o [Virtual Box](#), pois não tem custos e é compatível com a grande maioria dos sistemas operacionais.



Figura 1 – VirtualBox  
Fonte: elaborado pelo autor (2024)

A versão do Linux que usaremos é o [Ubuntu](#) 22.04.3 LTS (atualmente, a mais recente).

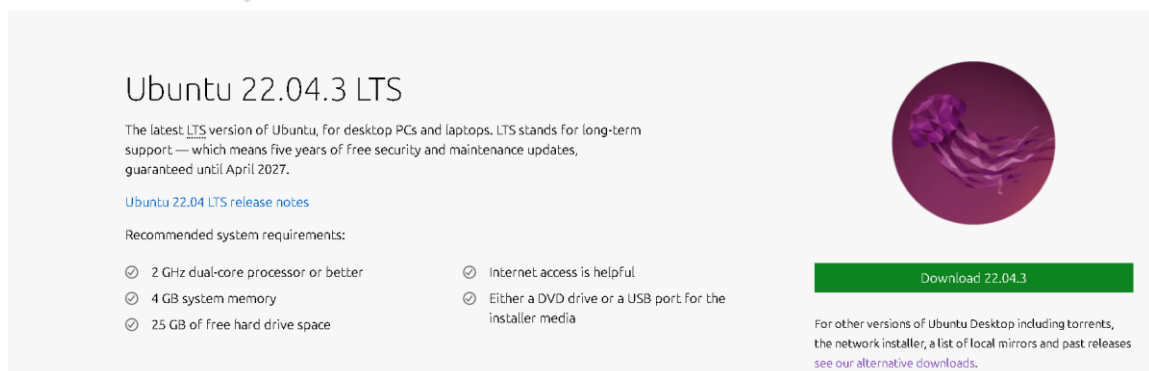


Figura 2 – Ubuntu 22.04.3 LTS  
Fonte: elaborado pelo autor (2024)

Realize a instalação padrão do Virtual Box, abra o aplicativo e clique em “novo”:



Figura 3 – Tela inicial do VirtualBox  
Fonte: elaborado pelo autor (2024)

Preencha as informações necessárias, apontando para a “.iso” que você baixou do Ubuntu:

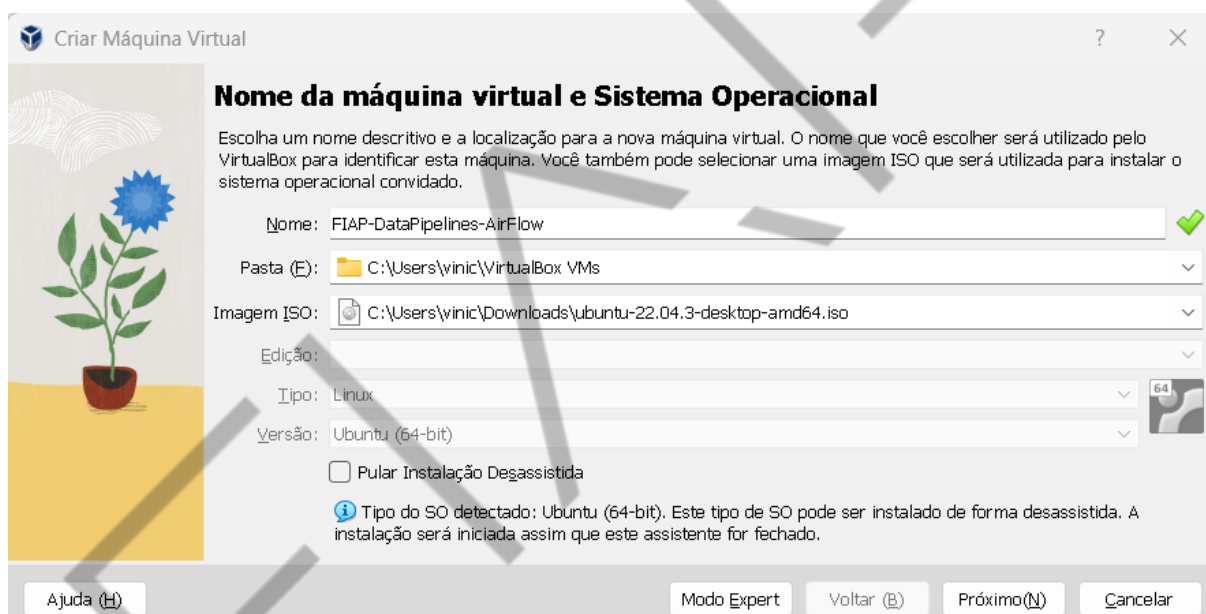


Figura 4 – Nome da máquina virtual e Sistema Operacional  
Fonte: elaborado pelo autor (2024)

Ao avançar, você deverá criar um usuário para essa máquina. Minha sugestão é seguir o mesmo usuário e senha da figura 5.

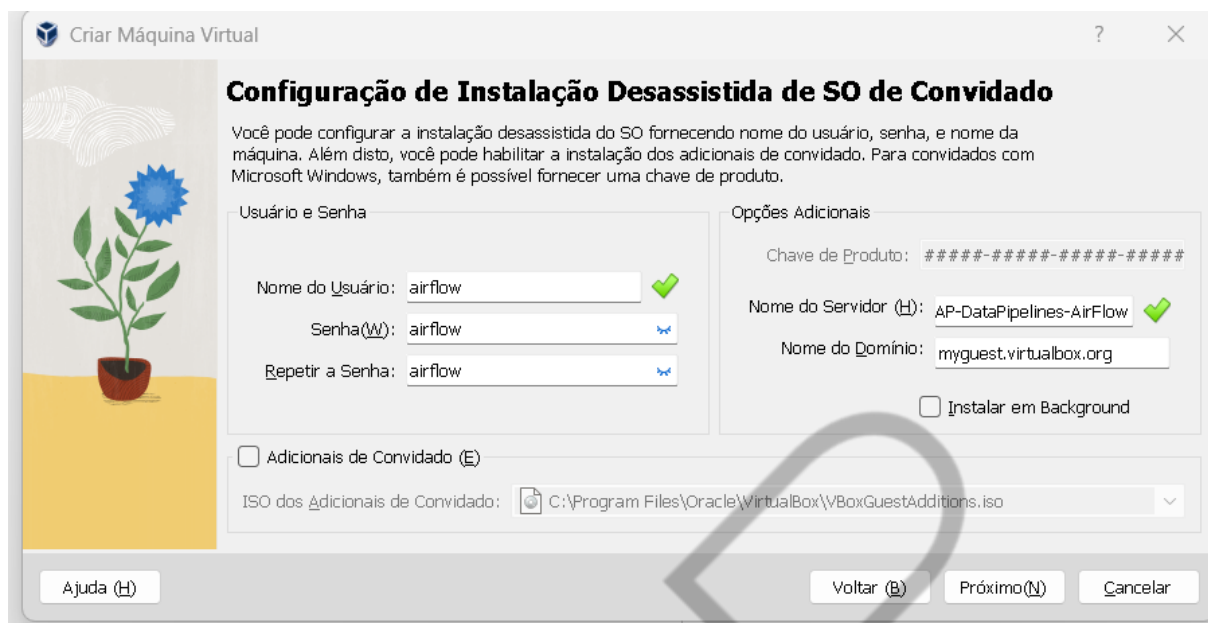


Figura 5 – Criando máquina virtual (2)  
Fonte: elaborado pelo autor (2024)

Agora, defina as configurações de hardware da máquina virtual. Em nosso projeto, o Apache AirFlow não executará nada, apenas coordenará as tarefas dentro dos pipelines; o processamento dos dados será executado dentro de um cluster EMR hospedado na Amazon, então não há necessidade de disponibilizar muito poder de processamento para essa máquina, algo entre 5 e 10 GB de memória é o suficiente.

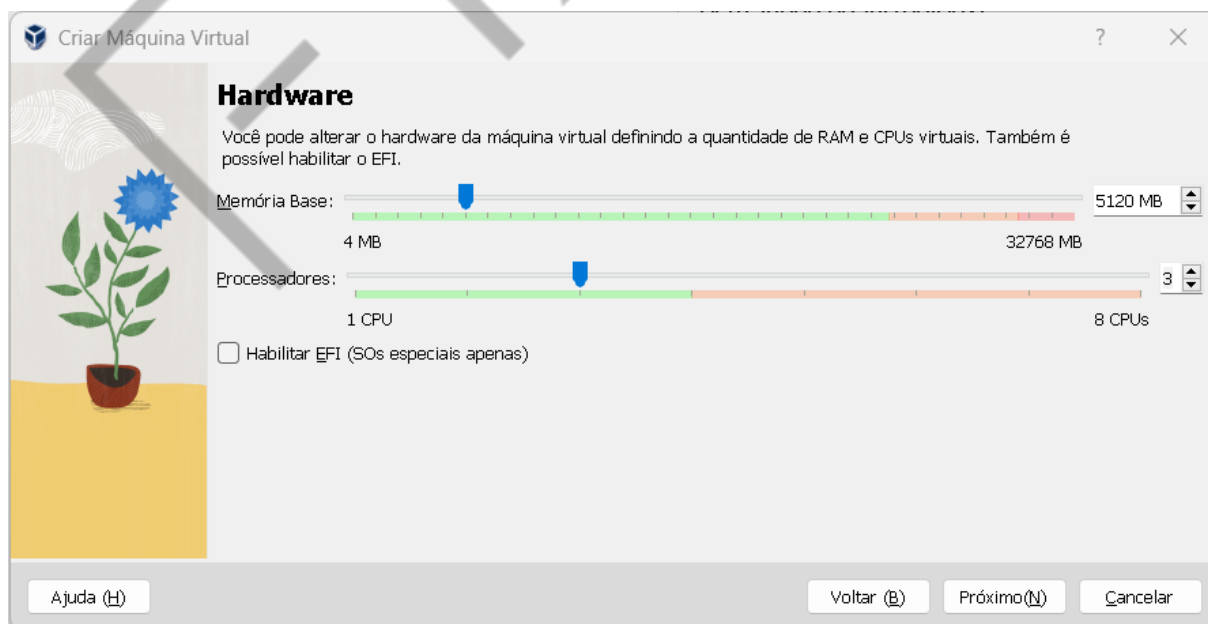


Figura 6 – Hardware  
Fonte: elaborado pelo autor (2024)

## Disco:



Figura 7 – Disco Rígido Virtual  
Fonte: elaborado pelo autor (2024)

Após essa definição, avance e finalize. Depois disso sua máquina virtual será iniciada. Pode demorar alguns minutos para subir a primeira configuração do sistema operacional. Ao finalizar, a VM será iniciada e você deverá fazer as configurações, se atentando em deixar nos padrões sugeridos, exceto essa página.



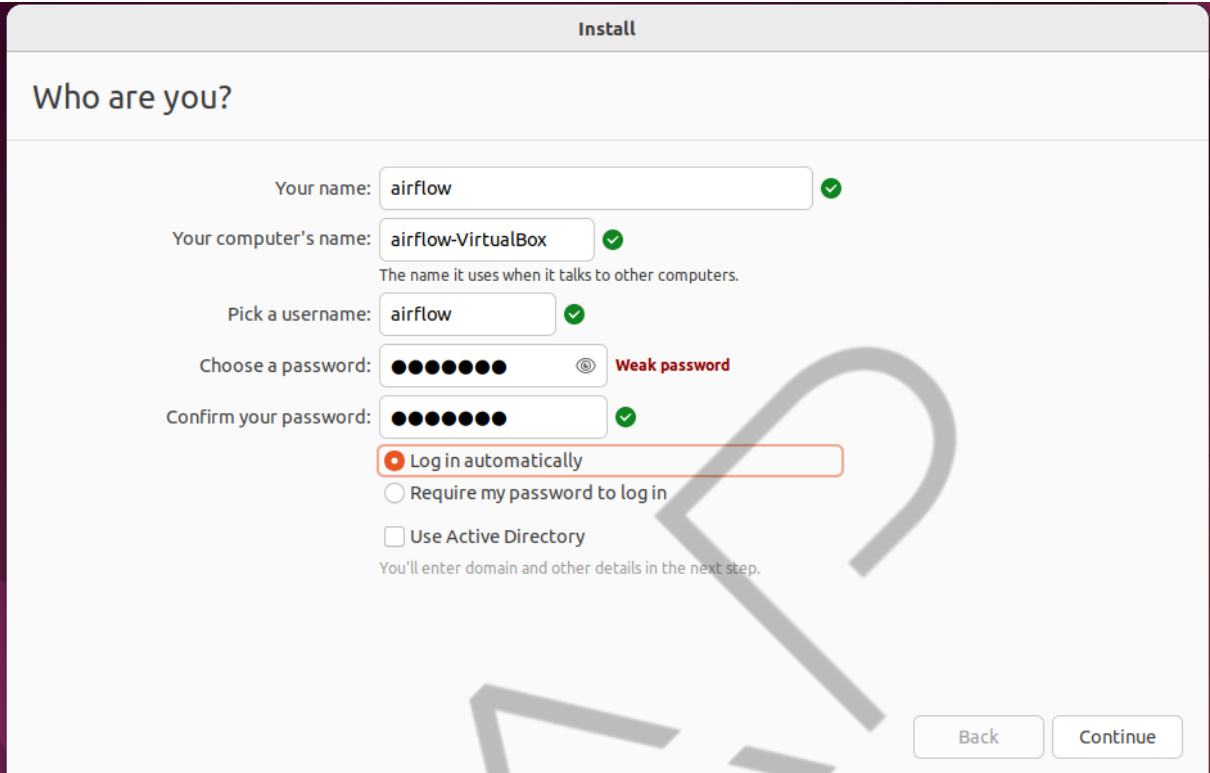


Figura 8 – Configurações  
Fonte: elaborado pelo autor (2024)

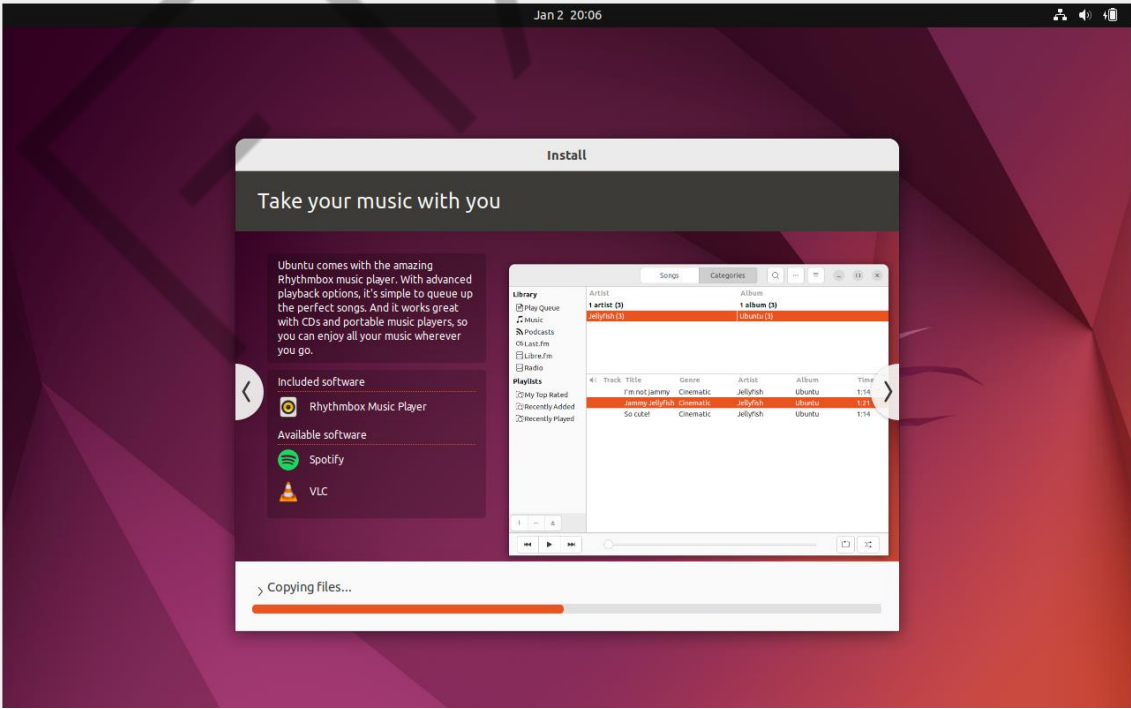


Figura 9 – Configurações (2)  
Fonte: elaborado pelo autor (2024)

Agora que o Ubuntu iniciou, vamos atualizar os pacotes e garantir que tudo o que precisamos está com a versão atualizada. Depois que os arquivos do sistema operacional forem copiados e instalados, vá em “Dispositivos”, no Virtual box, e selecione a opção destacada.

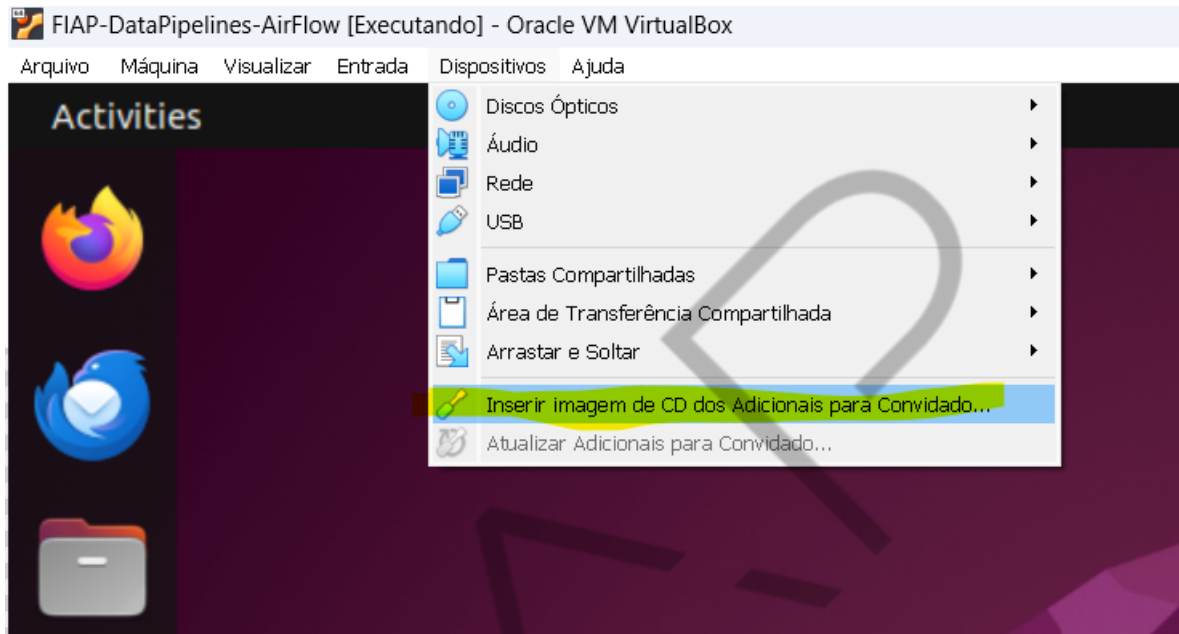


Figura 10 – Inserir Imagem de CD dos Adicionais para Convidado  
Fonte: elaborado pelo autor (2024)

Após isso, desligue a máquina virtual. Com a VM desligada, vá em configurações e faça as alterações para que você tenha os mesmos parâmetros da figura 11 e, então, inicie a máquina novamente.

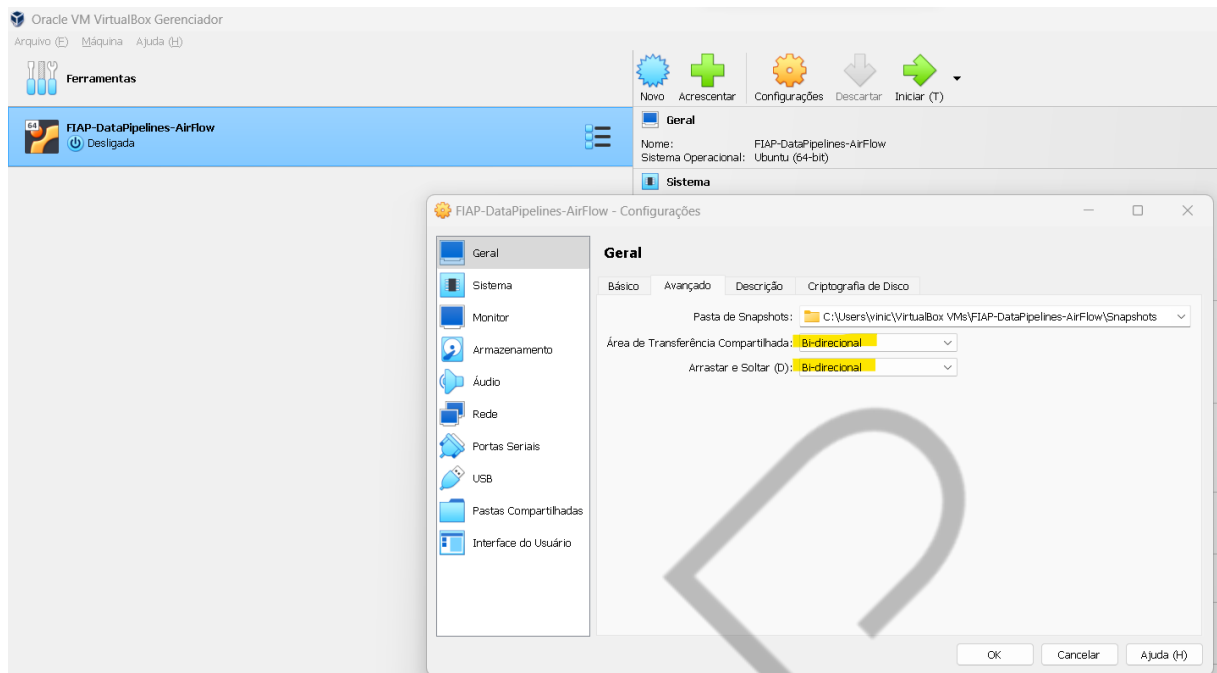


Figura 11 – Parâmetros indicados  
Fonte: elaborado pelo autor (2024)

Abra o terminal (atalho: Control + Alt + T) do Linux para executarmos os comandos necessários:

1. `sudo apt update`
2. `sudo apt upgrade`
3. `sudo apt install build-essential gcc make perl dkms curl tcl`

Abra o catálogo de software do ubuntu e instale o VSCode:

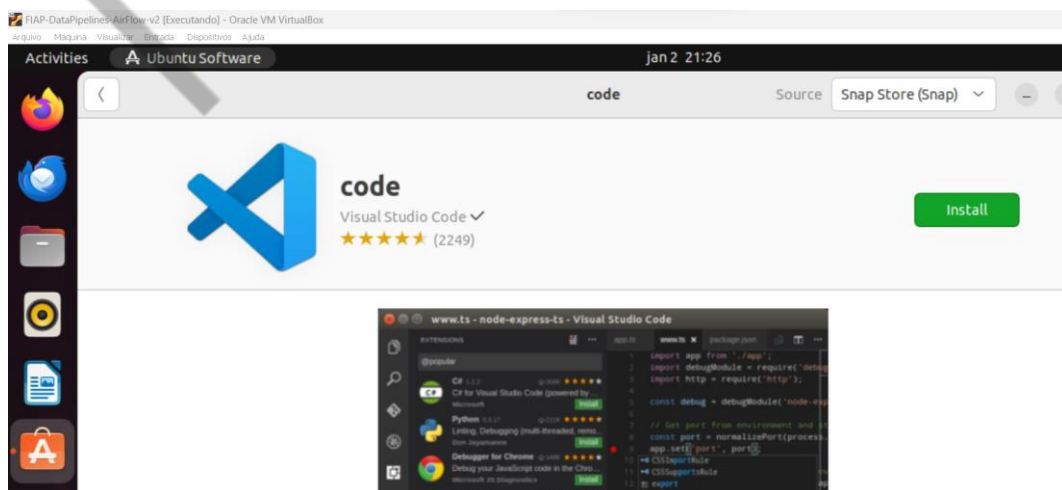


Figura 12 – Instalando o VSCode  
Fonte: elaborado pelo autor (2024)

Com o VS Code aberto, procure pela extensão do Python:

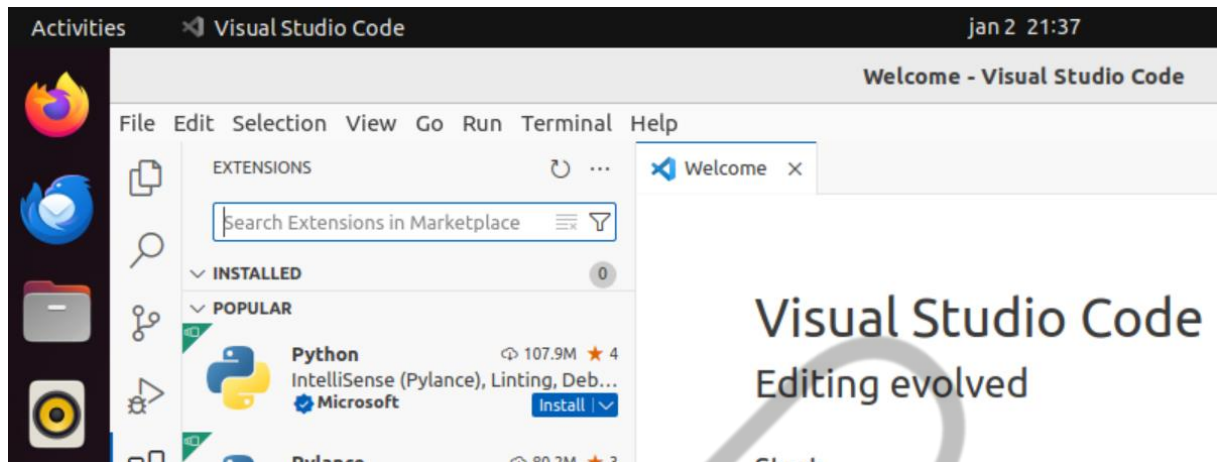


Figura 13 – Extensão do Python  
Fonte: elaborado pelo autor (2024)

Abra o terminal novamente para instalarmos o Python:

```
sudo apt install python3-pip -y
```

```
pip install pandas
```

Clique na imagem do CD que apareceu na barra de ferramentas do Ubuntu:



Figura 14 – Ícone de CD  
Fonte: elaborado pelo autor (2024)

Em seguida, na janela, abra um terminal dentro da pasta mostrada na figura 14.

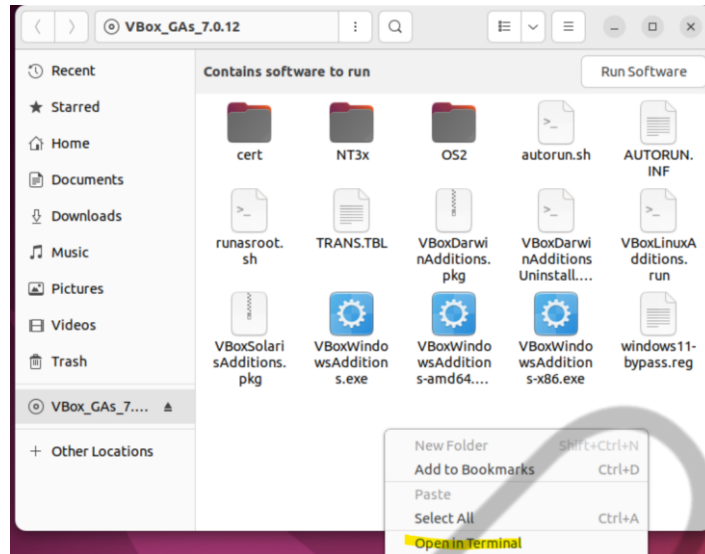


Figura 15 – Abrindo um terminal  
Fonte: elaborado pelo autor (2024)

Execute os seguintes comandos no terminal:

```
sudo apt install linux-headers-$(uname -r) build-essential dkms  
./runasroot.sh
```

Digite sua senha e aguarde a instalação. Quando finalizar, o terminal te mandará apertar a tecla “enter”.

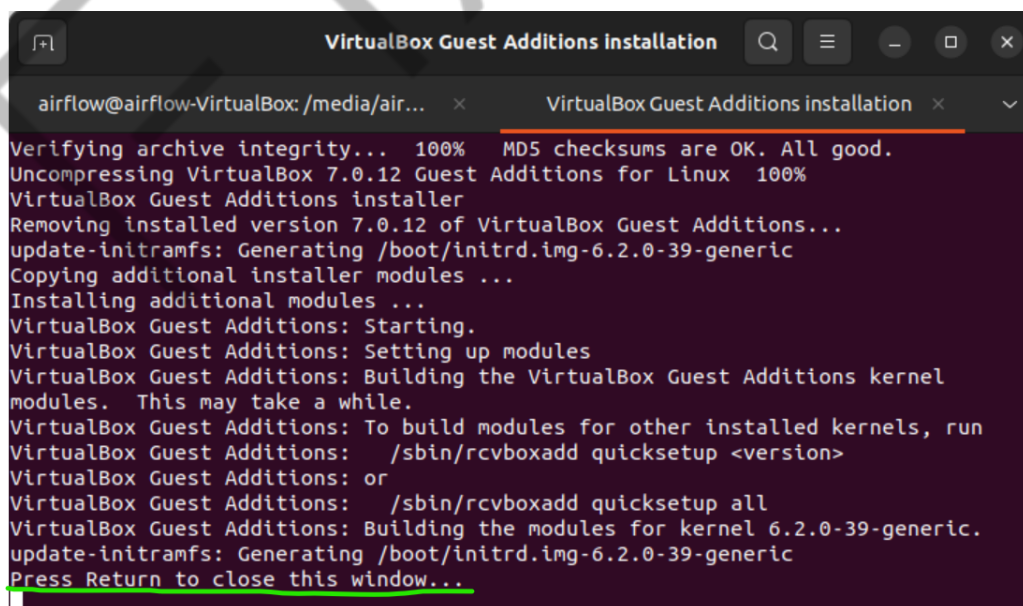
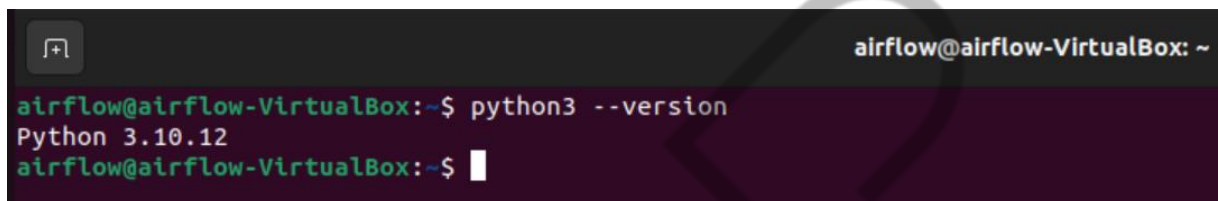


Figura 16 – Instalação finalizada  
Fonte: elaborado pelo autor (2024)

Feche o terminal e reinicie a máquina virtual. Quando ela ligar, você conseguirá copiar e colar textos ou arquivos entre máquina hospedeira e convidado, o que facilitará bastante as coisas daqui para frente.

## Configuração do ambiente dentro da Máquina Virtual

Abra o terminal e verifique qual é a versão do Python instalada: **python --version**.

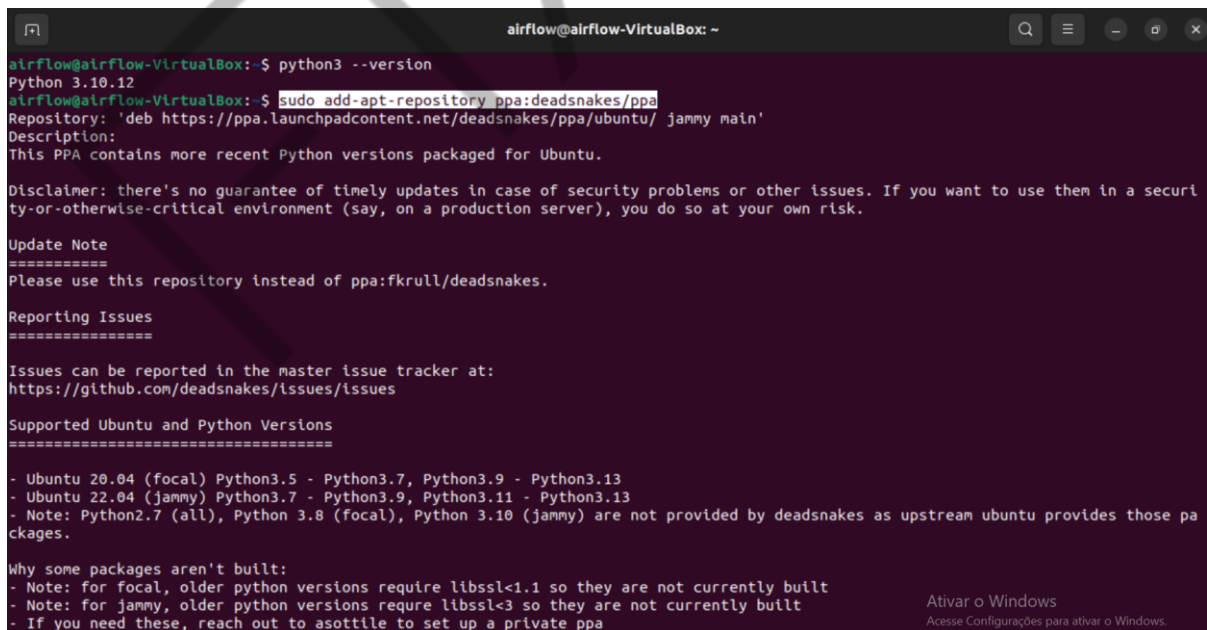
A terminal window titled 'airflow@airflow-VirtualBox: ~' with a dark background. The prompt is 'airflow@airflow-VirtualBox:~\$'. The command 'python3 --version' has been entered, and the output is 'Python 3.10.12'. The prompt is now 'airflow@airflow-VirtualBox:~\$' with a cursor.

```
airflow@airflow-VirtualBox:~$ python3 --version
Python 3.10.12
airflow@airflow-VirtualBox:~$
```

Figura 37 - Configuração do ambiente dentro da máquina virtual  
Fonte: elaborado pelo autor (2024)

Em seguida, execute o comando abaixo para que você possa ter várias versões do Python instaladas na mesma máquina:

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

A terminal window titled 'airflow@airflow-VirtualBox: ~' with a dark background. The prompt is 'airflow@airflow-VirtualBox:~\$'. The command 'python3 --version' has been entered, and the output is 'Python 3.10.12'. The prompt is now 'airflow@airflow-VirtualBox:~\$'. The command 'sudo add-apt-repository ppa:deadsnakes/ppa' has been entered, and the output is: 'Repository: 'deb https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu/ jammy main' Description: This PPA contains more recent Python versions packaged for Ubuntu. Disclaimer: there's no guarantee of timely updates in case of security problems or other issues. If you want to use them in a security-or-otherwise-critical environment (say, on a production server), you do so at your own risk. Update Note: Please use this repository instead of ppa:fkruhl/deadsnakes. Reporting Issues: Issues can be reported in the master issue tracker at: https://github.com/deadsnakes/issues/issues Supported Ubuntu and Python Versions: - Ubuntu 20.04 (focal) Python3.5 - Python3.7, Python3.9 - Python3.13 - Ubuntu 22.04 (jammy) Python3.7 - Python3.9, Python3.11 - Python3.13 - Note: Python2.7 (all), Python 3.8 (focal), Python 3.10 (jammy) are not provided by deadsnakes as upstream ubuntu provides those packages. Why some packages aren't built: - Note: for focal, older python versions require libssl<1.1 so they are not currently built - Note: for jammy, older python versions require libssl<3 so they are not currently built - If you need these, reach out to asottile to set up a private ppa'. The prompt is now 'airflow@airflow-VirtualBox:~\$'.

```
airflow@airflow-VirtualBox:~$ python3 --version
Python 3.10.12
airflow@airflow-VirtualBox:~$ sudo add-apt-repository ppa:deadsnakes/ppa
Repository: 'deb https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu/ jammy main'
Description:
This PPA contains more recent Python versions packaged for Ubuntu.

Disclaimer: there's no guarantee of timely updates in case of security problems or other issues. If you want to use them in a security-or-otherwise-critical environment (say, on a production server), you do so at your own risk.

Update Note
=====
Please use this repository instead of ppa:fkruhl/deadsnakes.

Reporting Issues
=====
Issues can be reported in the master issue tracker at:
https://github.com/deadsnakes/issues/issues

Supported Ubuntu and Python Versions
=====
- Ubuntu 20.04 (focal) Python3.5 - Python3.7, Python3.9 - Python3.13
- Ubuntu 22.04 (jammy) Python3.7 - Python3.9, Python3.11 - Python3.13
- Note: Python2.7 (all), Python 3.8 (focal), Python 3.10 (jammy) are not provided by deadsnakes as upstream ubuntu provides those packages.

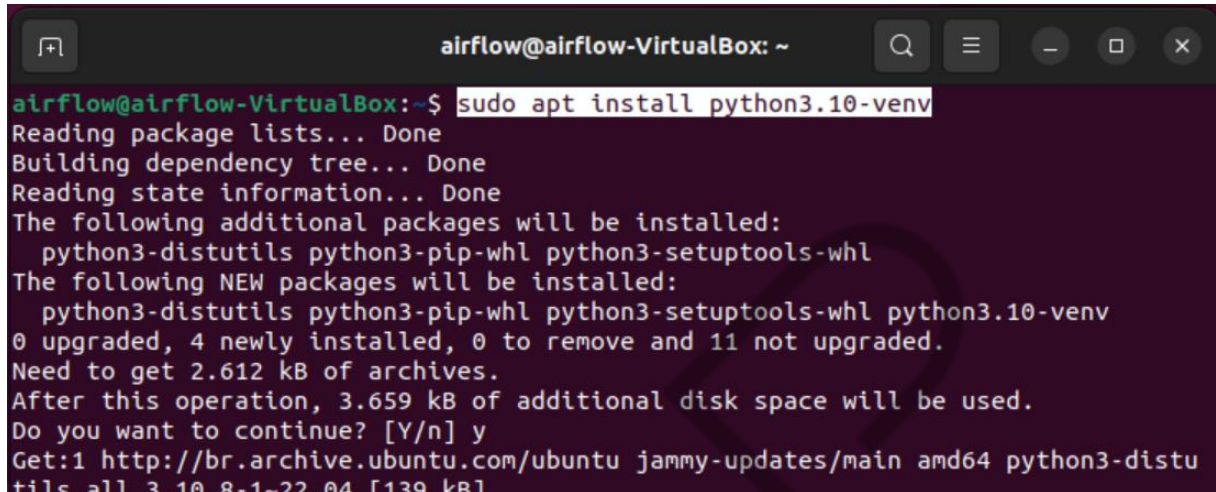
Why some packages aren't built:
- Note: for focal, older python versions require libssl<1.1 so they are not currently built
- Note: for jammy, older python versions require libssl<3 so they are not currently built
- If you need these, reach out to asottile to set up a private ppa

airflow@airflow-VirtualBox:~$
```

Figura 48 - Configuração do ambiente na MV  
Fonte: elaborado pelo autor (2024)



Aperte “enter” novamente e aguarde até que tudo seja instalado. Na sequência, instale o pacote de dependências para criação do ambiente virtual do Python 3.10: **sudo apt install python3.10-venv**.

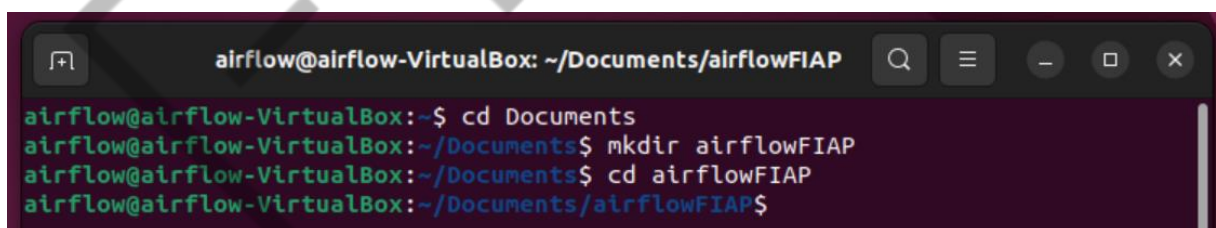
A terminal window titled 'airflow@airflow-VirtualBox: ~' showing the command 'sudo apt install python3.10-venv' being executed. The output shows the package lists being read, the dependency tree being built, and the state information being read. It lists additional packages to be installed (python3-distutils, python3-pip-whl, python3-setuptools-whl) and the new packages to be installed (python3-distutils, python3-pip-whl, python3-setuptools-whl, python3.10-venv). It also shows the disk space requirements and the confirmation to continue. The command is executed successfully.

```
airflow@airflow-VirtualBox:~$ sudo apt install python3.10-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-distutils python3-pip-whl python3-setuptools-whl
The following NEW packages will be installed:
  python3-distutils python3-pip-whl python3-setuptools-whl python3.10-venv
0 upgraded, 4 newly installed, 0 to remove and 11 not upgraded.
Need to get 2.612 kB of archives.
After this operation, 3.659 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-distutils all 3.10.8-1~22.04 [139 kB]
```

Figura 59 – Instalação do pacote de dependências  
Fonte: elaborado pelo autor (2024)

Agora, vamos começar a criar nosso ambiente para instalar o Apache Airflow.

```
cd documents
mkdir airflowFIAP
cd airflowFIAP
```

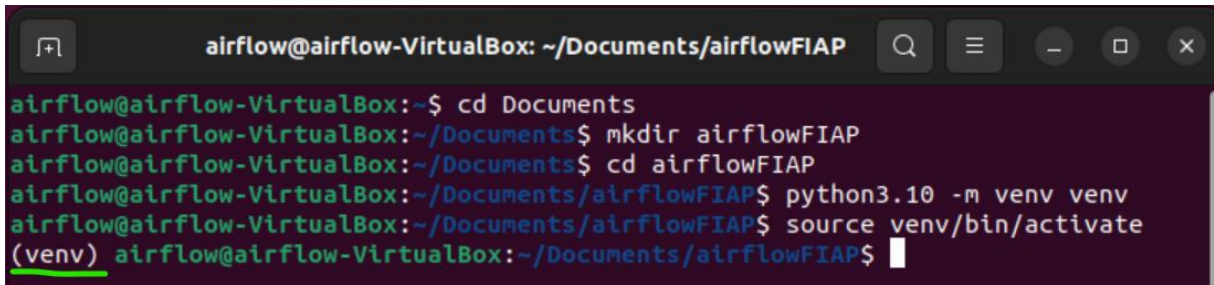
A terminal window titled 'airflow@airflow-VirtualBox: ~/Documents/airflowFIAP' showing the commands 'cd Documents', 'mkdir airflowFIAP', and 'cd airflowFIAP' being executed. The output shows the current directory being changed to ~/Documents, the directory airflowFIAP being created, and the current directory being changed to ~/Documents/airflowFIAP.

```
airflow@airflow-VirtualBox:~$ cd Documents
airflow@airflow-VirtualBox:~/Documents$ mkdir airflowFIAP
airflow@airflow-VirtualBox:~/Documents$ cd airflowFIAP
airflow@airflow-VirtualBox:~/Documents/airflowFIAP$
```

Figura 20 – Criação de ambiente para instalar o Apache Airflow  
Fonte: elaborado pelo autor (2024)

Você acabou de criar e entrar na pasta em que instalaremos a aplicação do Apache AirFlow. Agora, dentro dessa pasta, criaremos o ambiente virtual do Python 3.10:

```
python3.10 -m venv venv
source venv/bin/activate
```

A terminal window titled 'airflow@airflow-VirtualBox: ~/Documents/airflowFIAP'. The terminal shows the following commands and output: 

```
airflow@airflow-VirtualBox:~$ cd Documents
airflow@airflow-VirtualBox:~/Documents$ mkdir airflowFIAP
airflow@airflow-VirtualBox:~/Documents$ cd airflowFIAP
airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ python3.10 -m venv venv
airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ source venv/bin/activate
(venv) airflow@airflow-VirtualBox:~/Documents/airflowFIAP$
```

Figura 21 - Criando o ambiente virtual do Python 3.10  
Fonte: elaborado pelo autor (2024)

Agora que o ambiente virtual está instalado, configurado e ativado, podemos iniciar a instalação do Apache AirFlow. Para isso, abra o terminal novamente, caso tenha encerrado, navegue até a pasta que criamos e ative o ambiente virtual:

```
cd Documents/airflowFIAP
```

```
source venv/bin/activate
```

```
pip install 'apache-airflow==2.8.0' --constraint
```

```
"https://raw.githubusercontent.com/apache/airflow/constraints-2.8.0/constraints-3.10.txt"
```

Aqui, tudo o que é necessário para a instalação da versão 2.8.0 do Airflow será feito através do arquivo de restrições.

Ao finalizar, execute o comando a seguir para setar a variável de ambiente para a “home” do AirFlow em nossa máquina:

```
export AIRFLOW_HOME=~/Documents/airflowFIAP
```

A terminal window titled 'airflow@airflow-VirtualBox: ~/Documents/airflowFIAP'. The terminal shows the following commands and output: 

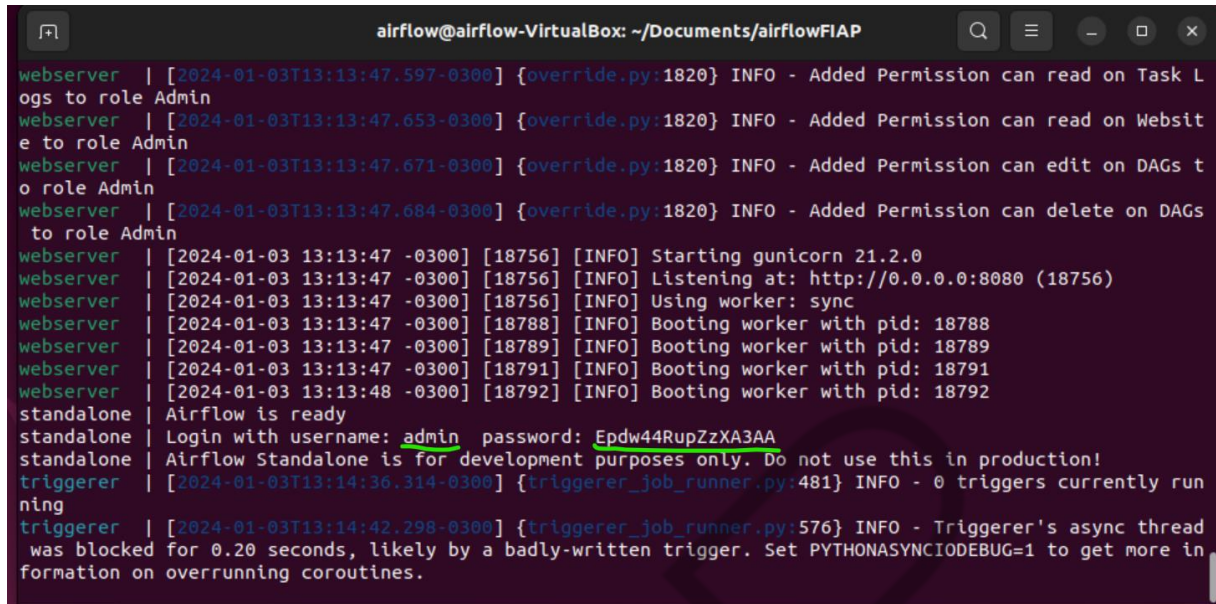
```
(venv) airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ export AIRFLOW_HOME=~/Documents/airflowFIAP
(venv) airflow@airflow-VirtualBox:~/Documents/airflowFIAP$
```

Figura 22 – Variável de ambiente para a home do AirFlow  
Fonte: elaborado pelo autor (2024)

Inicie o AirFlow pelo terminal, salve o login e senha criados:

```
airflow standalone
```

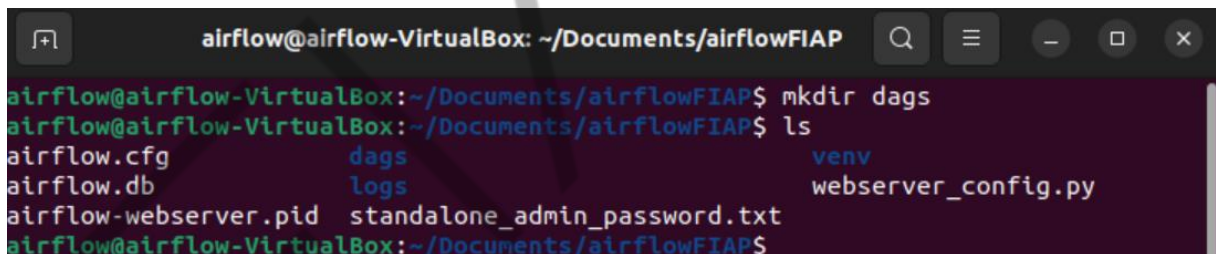




```
airflow@airflow-VirtualBox: ~/Documents/airflowFIAP
webserver | [2024-01-03T13:13:47.597-0300] {override.py:1820} INFO - Added Permission can read on Task L
ogs to role Admin
webserver | [2024-01-03T13:13:47.653-0300] {override.py:1820} INFO - Added Permission can read on Websit
e to role Admin
webserver | [2024-01-03T13:13:47.671-0300] {override.py:1820} INFO - Added Permission can edit on DAGs t
o role Admin
webserver | [2024-01-03T13:13:47.684-0300] {override.py:1820} INFO - Added Permission can delete on DAGs
to role Admin
webserver | [2024-01-03 13:13:47 -0300] [18756] [INFO] Starting gunicorn 21.2.0
webserver | [2024-01-03 13:13:47 -0300] [18756] [INFO] Listening at: http://0.0.0.0:8080 (18756)
webserver | [2024-01-03 13:13:47 -0300] [18756] [INFO] Using worker: sync
webserver | [2024-01-03 13:13:47 -0300] [18788] [INFO] Booting worker with pid: 18788
webserver | [2024-01-03 13:13:47 -0300] [18789] [INFO] Booting worker with pid: 18789
webserver | [2024-01-03 13:13:47 -0300] [18791] [INFO] Booting worker with pid: 18791
webserver | [2024-01-03 13:13:48 -0300] [18792] [INFO] Booting worker with pid: 18792
standalone | Airflow is ready
standalone | Login with username: admin password: Epdw44RupZzXA3AA
standalone | Airflow Standalone is for development purposes only. Do not use this in production!
triggerer | [2024-01-03T13:14:36.314-0300] {triggerer_job_runner.py:481} INFO - 0 triggers currently run
ning
triggerer | [2024-01-03T13:14:42.298-0300] {triggerer_job_runner.py:576} INFO - Triggerer's async thread
was blocked for 0.20 seconds, likely by a badly-written trigger. Set PYTHONASYNCIODEBUG=1 to get more in
formation on overrunning coroutines.
```

Figura 23 – Iniciando o AirFlow  
Fonte: elaborado pelo autor (2024)

Crie uma pasta para gerenciarmos nossas dags, seguindo os comandos da figura 24:



```
airflow@airflow-VirtualBox: ~/Documents/airflowFIAP
airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ mkdir dags
airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ ls
airflow.cfg          dags                venv
airflow.db           logs               webserver_config.py
airflow-webserver.pid standalone_admin_password.txt
```

Figura 24 – Comandos  
Fonte: elaborado pelo autor (2024)

Abra um novo terminal e execute o seguinte comando para a instalação do “pip”:

```
sudo apt install python3-pip
```

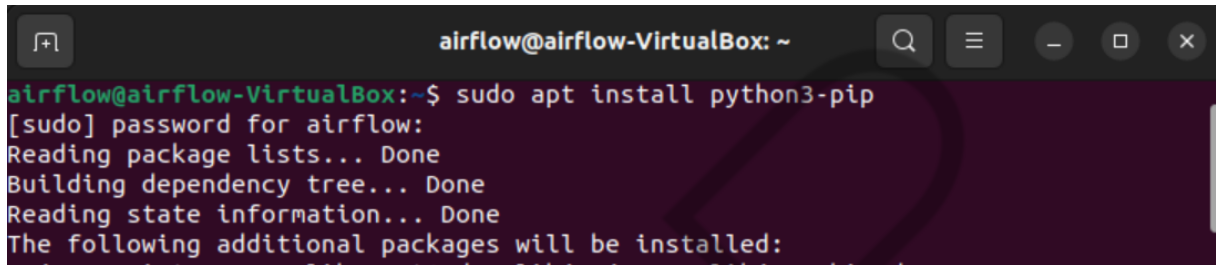
```
pip install apache-airflow
```

```
pip install apache-airflow-providers-amazon
```

```
pip install scikit-learn
```

```
pip install connexion[swagger-ui]
```

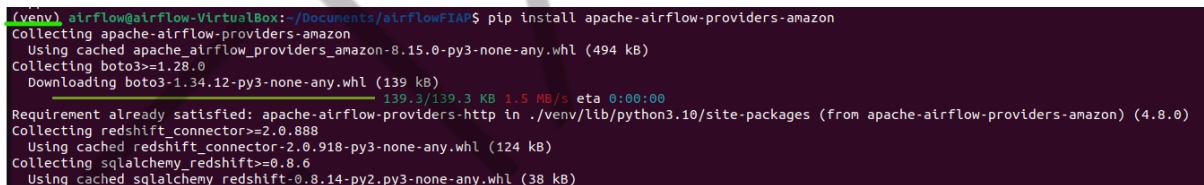
```
pip install pyspark  
pip install boto3  
pip install pandas  
pip install pendulum  
pip install virtualenv
```



```
airflow@airflow-VirtualBox: ~  
airflow@airflow-VirtualBox:~$ sudo apt install python3-pip  
[sudo] password for airflow:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:
```

Figura 25 – Instalação do pip  
Fonte: elaborado pelo autor (2024)

Ative seu ambiente virtual novamente e execute os mesmos comandos. Isso garantirá que o ambiente virtual terá disponível os mesmos pacotes que teremos ao desenvolver nosso código, pois o AirFlow executará os códigos usando o ambiente virtual em que foi instalado.



```
(venv) airflow@airflow-VirtualBox:~/Documents/airflowFIAP$ pip install apache-airflow-providers-amazon  
Collecting apache-airflow-providers-amazon  
  Using cached apache_airflow_providers_amazon-8.15.0-py3-none-any.whl (494 kB)  
Collecting boto3<=1.28.0  
  Downloading boto3-1.34.12-py3-none-any.whl (139 kB)  
    139.3/139.3 KB 1.5 MB/s eta 0:00:00  
Requirement already satisfied: apache-airflow-providers-http in ./venv/lib/python3.10/site-packages (from apache-airflow-providers-amazon) (4.8.0)  
Collecting redshift_connector<=2.0.888  
  Using cached redshift_connector-2.0.918-py3-none-any.whl (124 kB)  
Collecting sqlalchemy_redshift<=0.8.6  
  Using cached sqlalchemy_redshift-0.8.14-py2.py3-none-any.whl (38 kB)
```

Figura 26 – O AirFlow executará os códigos usando o ambiente virtual em que foi instalado  
Fonte: elaborado pelo autor (2024)

Todas as bibliotecas devem ser instaladas no ambiente virtual e local.

```
pip install apache-airflow  
pip install apache-airflow-providers-amazon  
pip install scikit-learn  
pip install pyspark  
pip install connexion[swagger-ui]  
pip install pandas  
pip install pendulum
```

```
pip install virtualenv
```

```
pip install boto3
```

Pronto, seu ambiente está configurado e pronto para a aula prática.

## Configurando AWS

Faça o login na sua conta e entre no console da AWS. Em seguida, procure pelo serviço IAM:

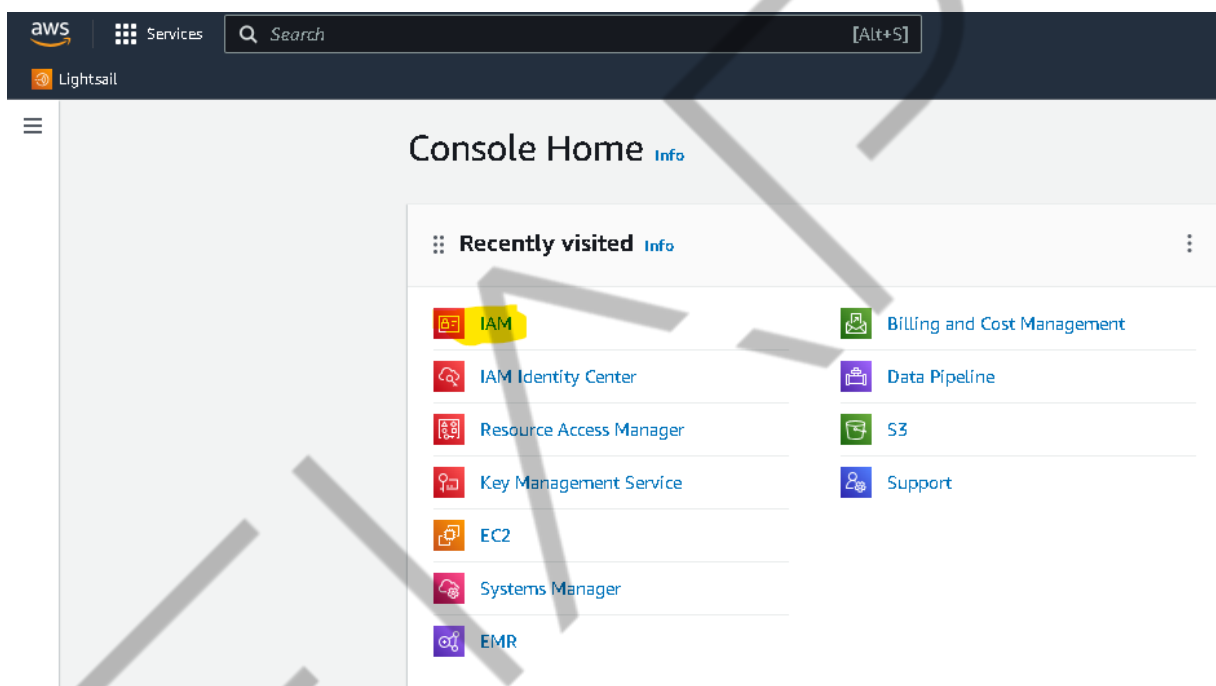


Figura 27 – Console AWS  
Fonte: elaborado pelo autor (2024)

Na tela do serviço IAM, clique em “users”, no canto esquerdo:

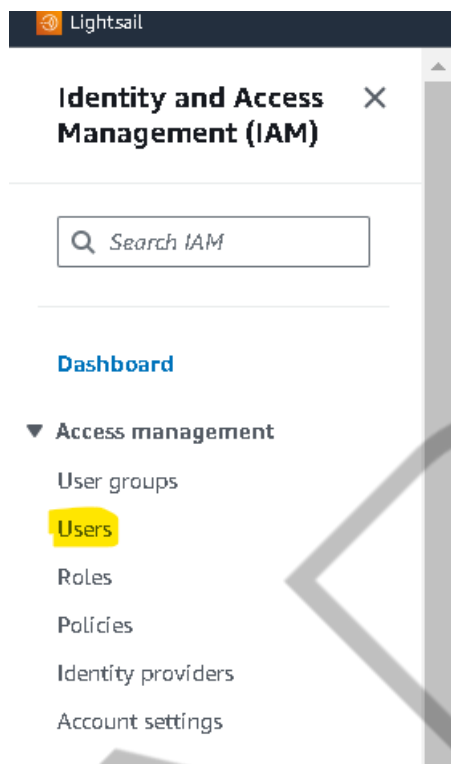


Figura 28 - Users  
Fonte: elaborado pelo autor (2024)

Em seguida, clique em “Create user” e, após ser redirecionado, siga as configurações da figura 29.

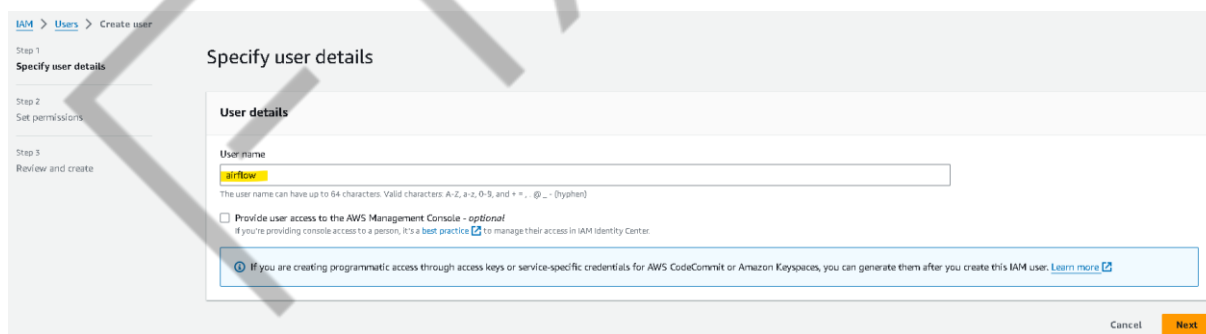


Figura 29 – Detalhes de usuário  
Fonte: elaborado pelo autor (2024)

Ao prosseguir, selecione “Attach policies directly” para já configurarmos algumas políticas para esse novo usuário da nossa conta. Localize as políticas da figura 30.

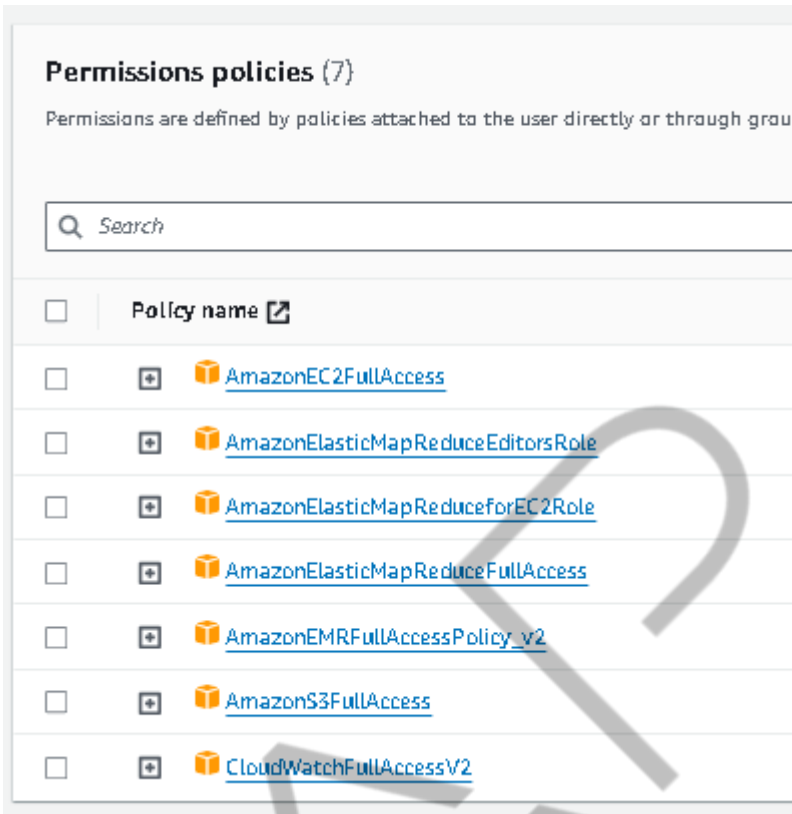


Figura 30 – Permission policies  
Fonte: elaborado pelo autor (2024)

Depois de selecionar essas sete políticas relacionadas aos serviços que usaremos, avance até a criação do usuário, o que pode levar alguns minutos. Assim que criar, clique no hiperlink do nome do seu usuário (no meu caso, “airflow”).

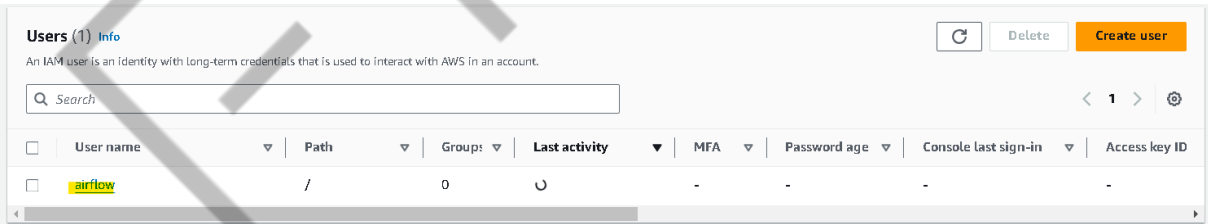


Figura 31 - Hiperlink  
Fonte: elaborado pelo autor (2024)

Na sequência, clique no hiperlink “Create access key”:

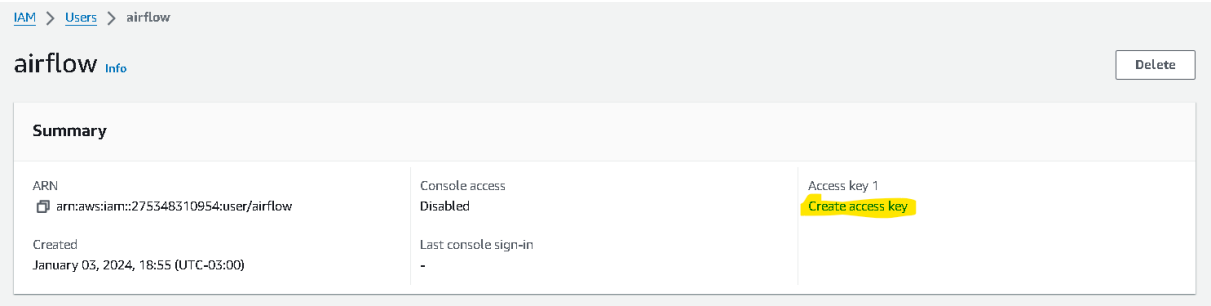


Figura 32 – Create access key  
Fonte: elaborado pelo autor (2024)

Selecione a mesma configuração da figura 33 e prossiga.

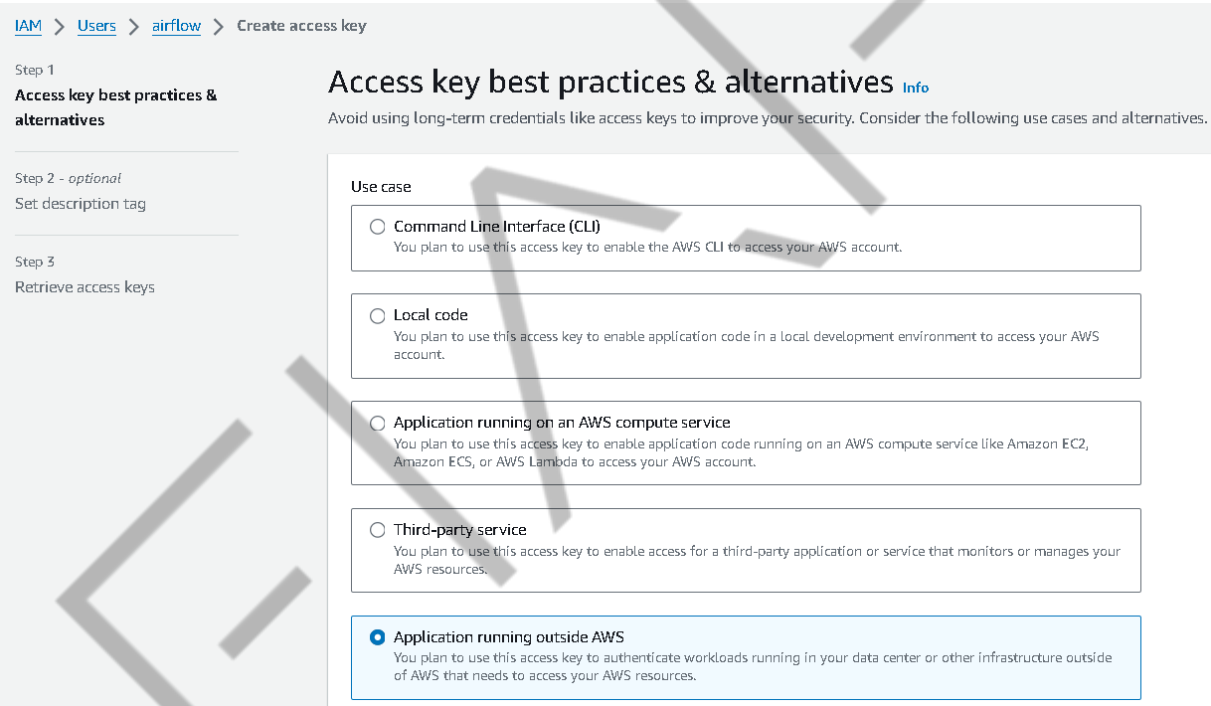
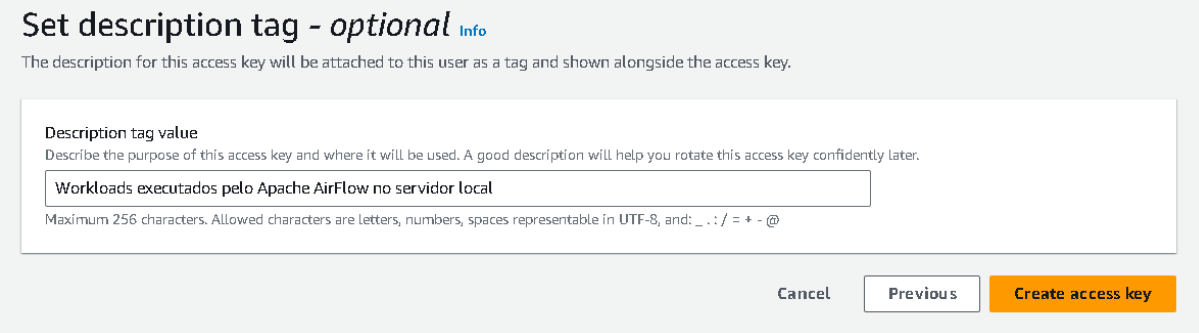


Figura 33 – Configuração access key  
Fonte: elaborado pelo autor (2024)

Coloque sua descrição e crie sua chave:



**Set description tag - optional** [Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

**Description tag value**  
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

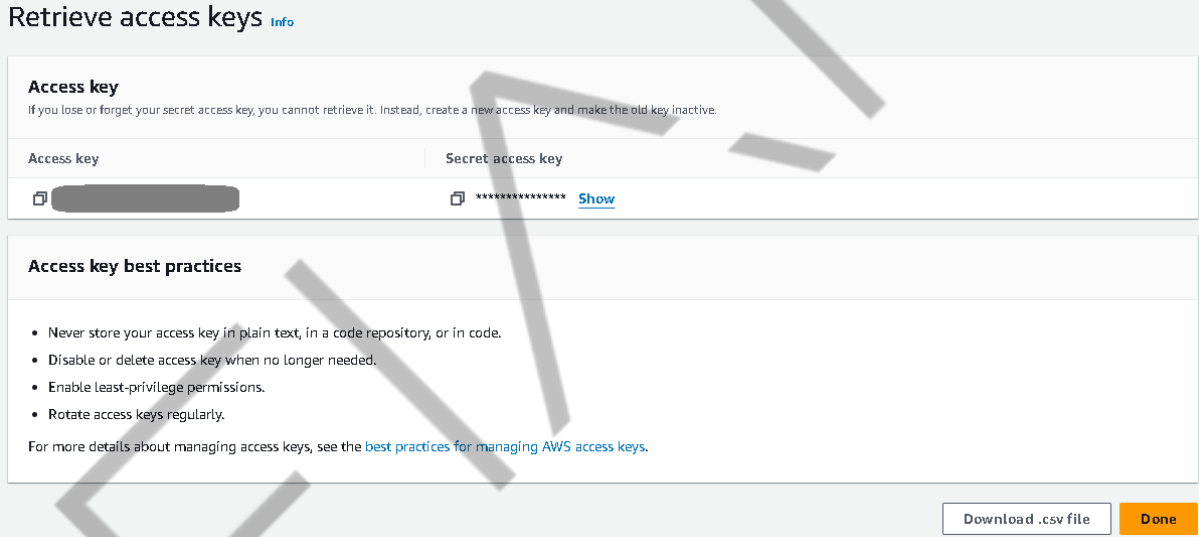
Workloads executados pelo Apache AirFlow no servidor local

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: \_ . : / = + - @

Cancel Previous **Create access key**



Figura 34 – Description tag value  
Fonte: elaborado pelo autor (2024)

Ao gerar sua chave, salve suas credenciais (Access Key e Secret Access key) em um local seguro e nunca forneça a ninguém, pois ela não será exibida novamente.



**Retrieve access keys** [Info](#)

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 [Redacted]	 ***** <a href="#">Show</a>

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file Done

Figura 35 – Access key e secret access key  
Fonte: elaborado pelo autor (2024)

Novamente no serviço do IAM, crie uma regra com este nome: “EMR\_EC2\_DefaultRole”.

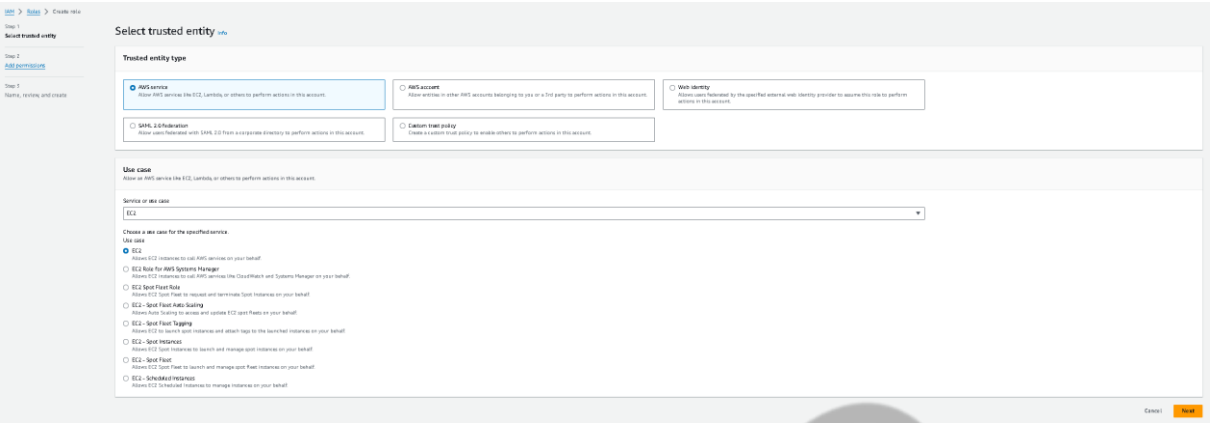


Figura 36 – Regra EMR\_EC2\_DefaultRole  
Fonte: elaborado pelo autor (2024)

Encontre e anexe todas as políticas da figura 37.

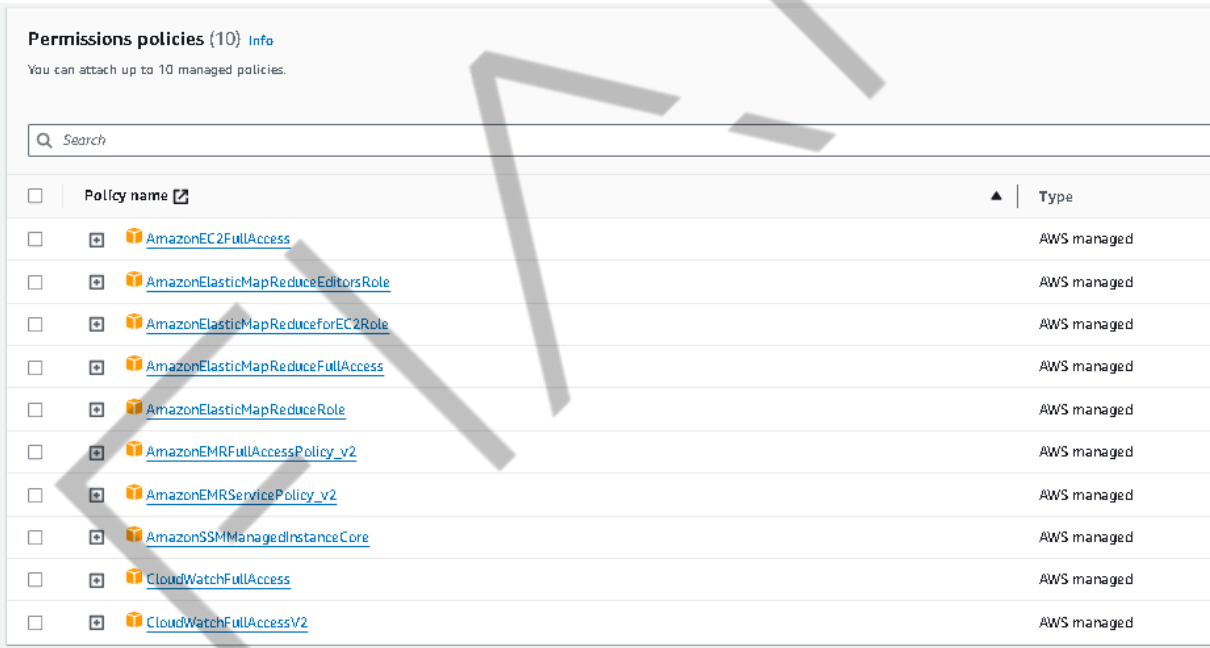


Figura 37 – Políticas  
Fonte: elaborado pelo autor (2024)

Crie uma nova regra chamada “regras-emr” e, seguindo o mesmo procedimento dentro do IAM, anexe as políticas da figura 38.



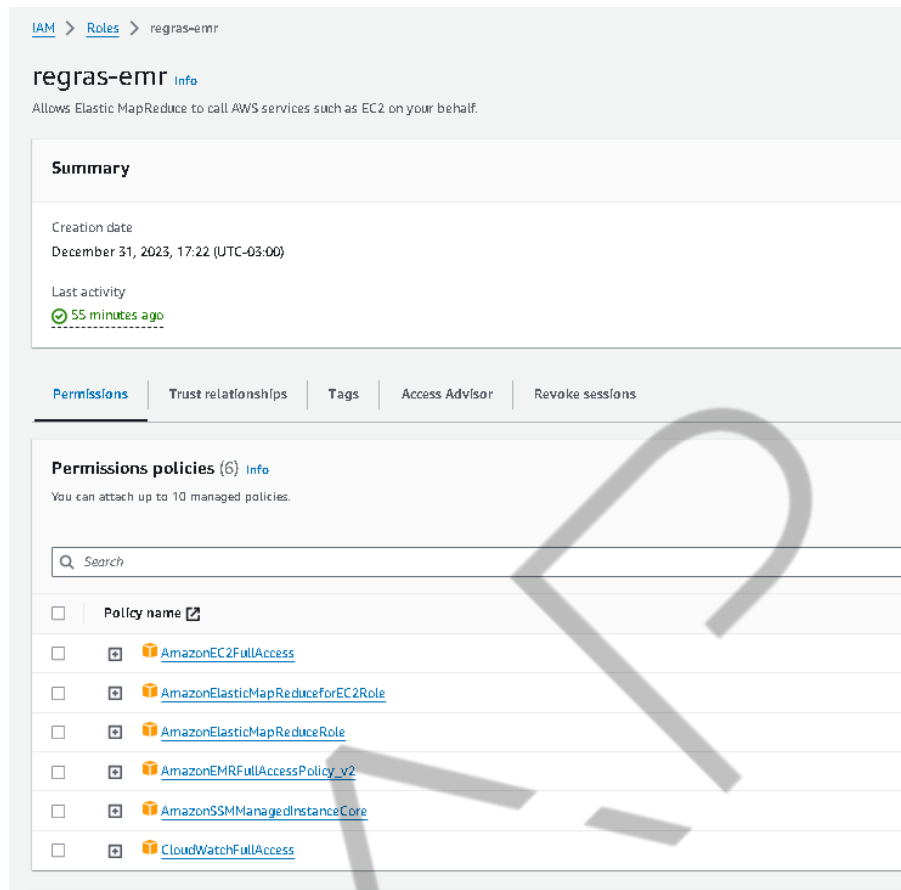


Figura 38 – regras-emr  
Fonte: elaborado pelo autor (2024)

Vá até o serviço do S3, no console, e crie um bucket que será destinado aos logs do EMR.

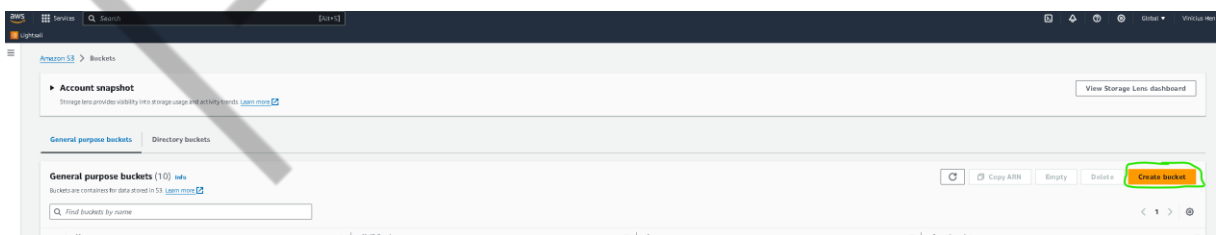


Figura 39 – criar bucket  
Fonte: elaborado pelo autor (2024)

Use as configurações básicas sugeridas pelo S3 e crie uma pasta chamada “logs” dentro desse bucket.

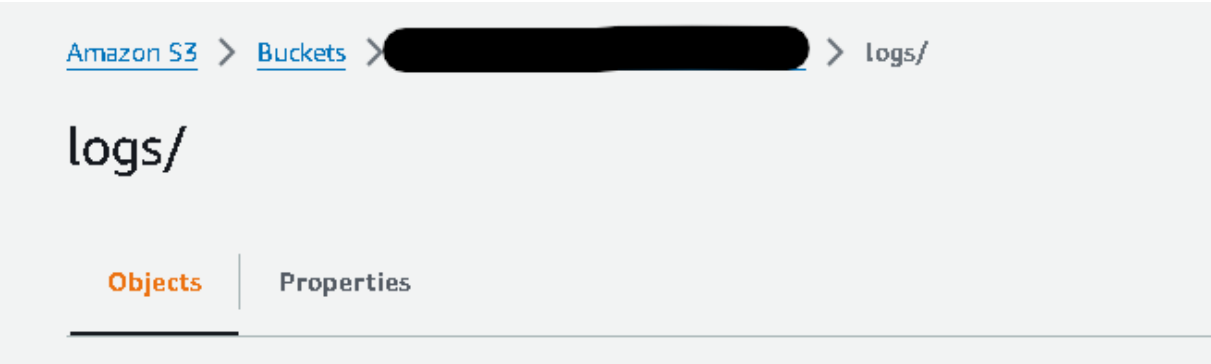


Figura 40 – pasta logs  
Fonte: elaborado pelo autor (2024)

Crie outro bucket para receber os dados e os scripts dos jobs que executaremos.

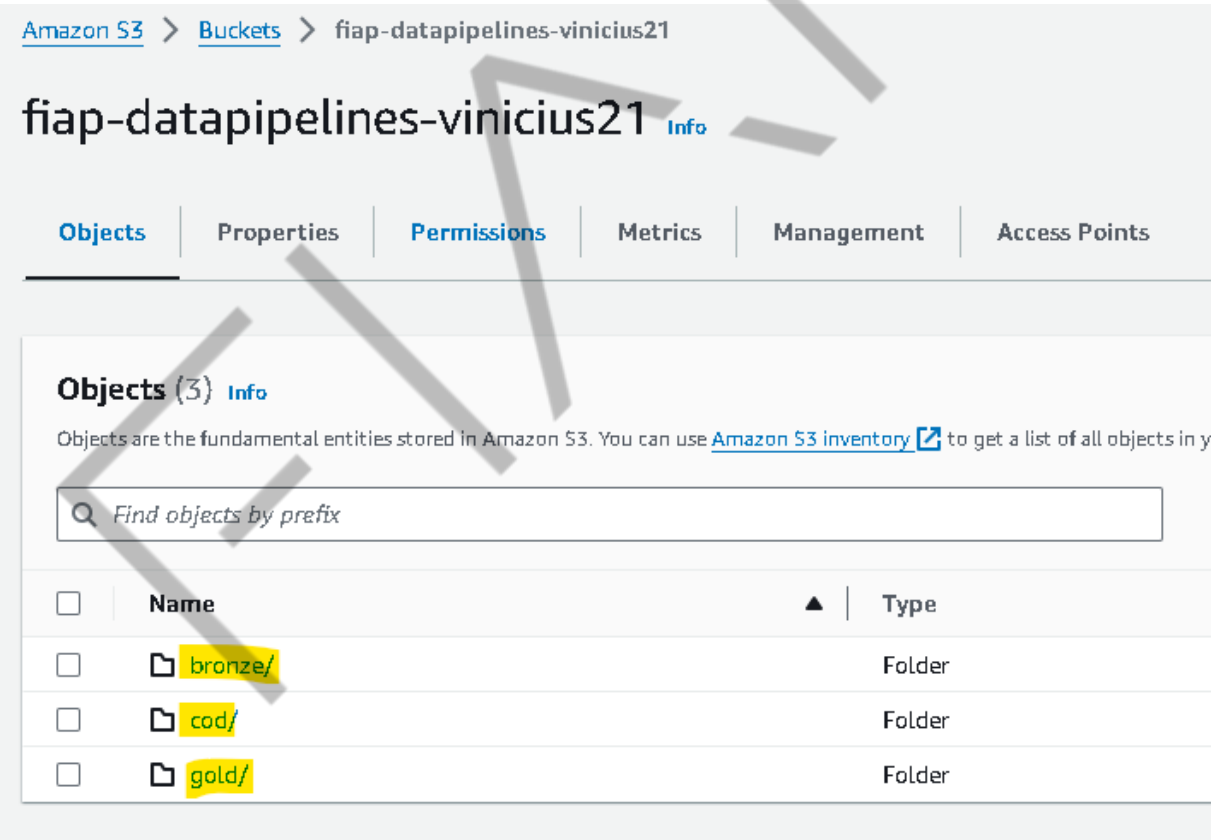


Figura 41 - Buckets  
Fonte: elaborado pelo autor (2024)

Esse bucket deve conter as 3 pastas da figura 41:

- 1. Bronze: receberá os arquivos brutos da carga de trabalho.

2. Gold: receberá o resultado após os processamentos realizados nas cargas de trabalho.
3. Cod: onde o processo buscará o código para realizar o “Spark submit” e processar os dados.

## Criação do KeyPair para uso no EC2

Navegue até o serviço do EC2 no console e encontre o hiperlink “Key pairs”:

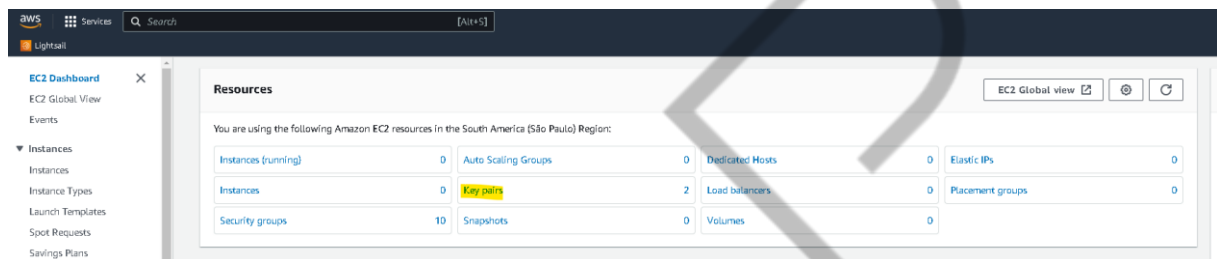


Figura 42 - Key pairs  
Fonte: elaborado pelo autor (2024)

Vá em “Criar nova Key Pair” e mantenha as configurações indicadas na figura 43.

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 ASCII 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.  
[Add new tag](#)  
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

Figura 43 – configurações do key pair  
Fonte: elaborado pelo autor (2024)

Vá até o serviço de VPC em seu console e clique em “Create VPC”.

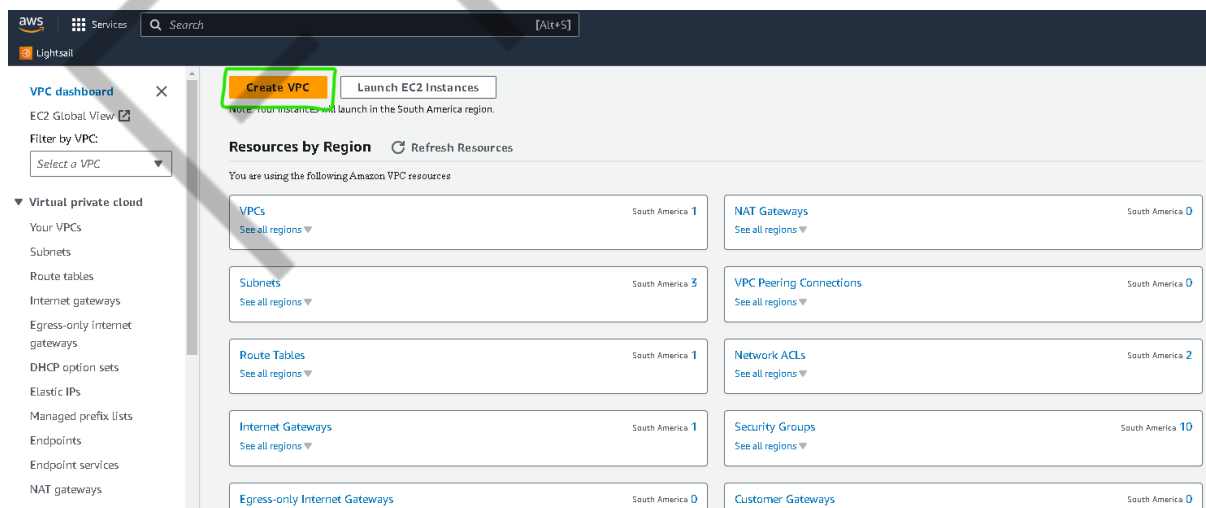


Figura 44 – Create VPC  
Fonte: elaborado pelo autor (2024)

Siga as configurações das figuras 45 e 46 e crie sua VPC e subnet, caso ainda não tenha.

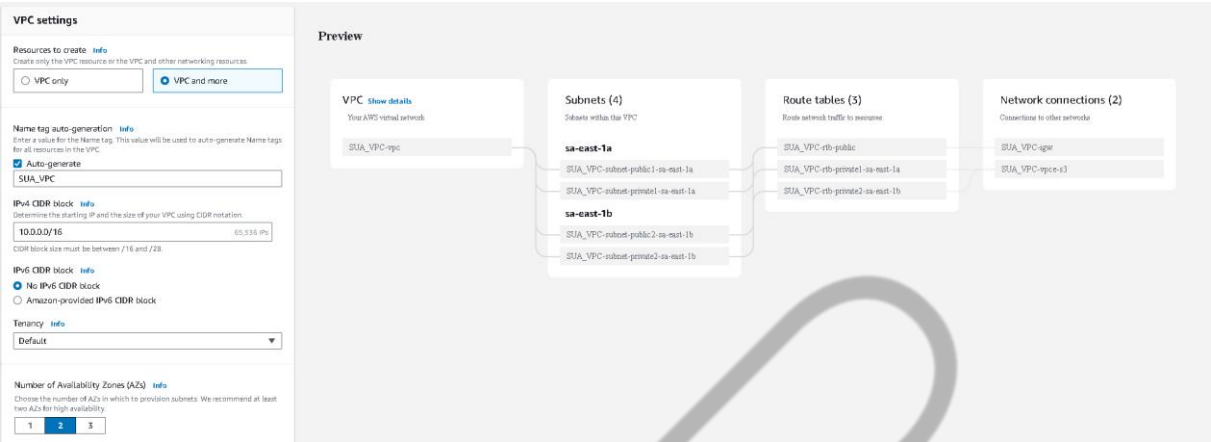


Figura 45 – configurações VPC (1)  
Fonte: elaborado pelo autor (2024)

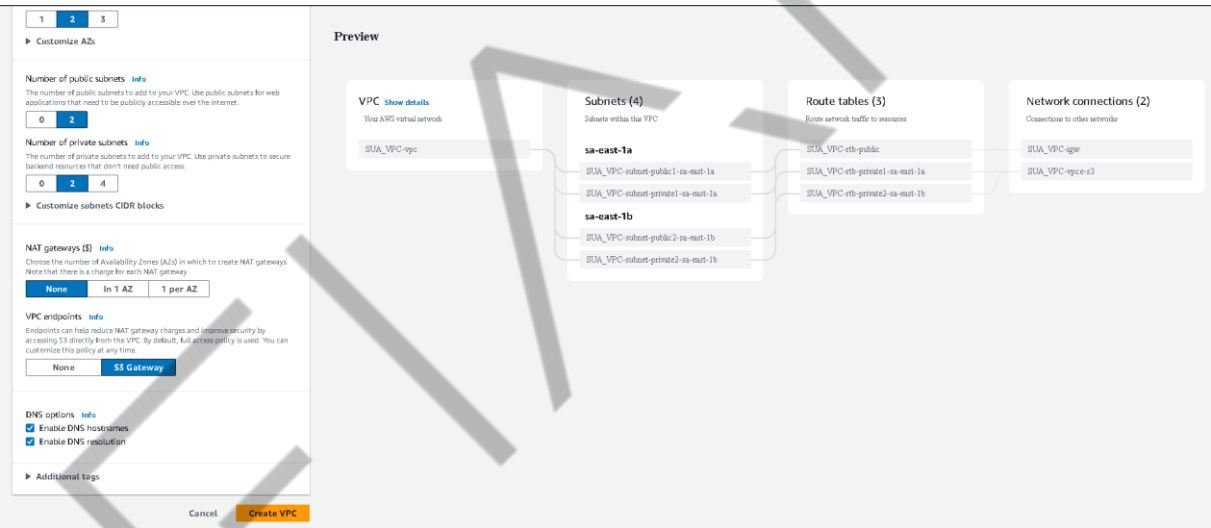


Figura 46 - configurações VPC (2)  
Fonte: elaborado pelo autor (2024)

## O QUE VOCÊ VIU NESTA AULA?

Nessa aula você entendeu o que é uma rotina em batch e quais são os principais componentes (Hadoop, Spark, HDFS, Airflow). Além disso, conseguiu desenvolver scripts que resolvem os problemas reais de trabalho em qualquer empresa e colocou em prática todo esse conhecimento em um projeto de construção de um pipeline planejado, robusto e escalável para entregar os resultados que o negócio necessita, combinando o uso do Airflow, EMR, Spark e Python.

## REFERÊNCIAS

APACHE FOUNDATION. **Quick Start**. 2023. Disponível em: <<https://spark.apache.org/docs/latest/quick-start.html>>. Acesso em: 21 mar. 2024.

APACHE FOUNDATION. **What is Airflow™?**. 2023. Disponível em: <<https://airflow.apache.org/docs/apache-airflow/stable/index.html>>. Acesso em: 21 mar. 2024.

DAMJI, J. **Learning Spark: Lightning-Fast Data Analytics**. Sebastopol: O'Reilly Media, 2020.

HURWITZ, J. **Big Data for Dummies**. EUA: For Dummies, 2013.

## **PALAVRAS-CHAVE**

**Palavras-chave:** Airflow. Python. Spark. Hadoop. HDFS. EMRFS. EMR. Data Pipeline.

EMRFS





POSTECH