Objetivo:

Realizar a criação e configuração de um ambiente onde você seja capaz de usar:

* Python
* Apache Spark (Scala e PySpark)
* Hadoop gerenciando os arquivos de forma distribuída
* Apache Airflow para desenvolvimento de dags e orquestração via WebServer

Configurações

1. Download dos softwares necessários:   
   1. Realize o download da “iso” do ubuntu desktop 24.04 LTS: <https://ubuntu.com/download/desktop>
   2. Realize o download do “Oracle Virtual Box” compatível com seu sistema operacional e instale: <https://www.virtualbox.org/wiki/Downloads>
2. Configuração da máquina virtual:  
   1. Abra o virtual box e clique em “Máquina > Criar”, siga as mesmas configurações das imagens:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente  
  
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente  
  
Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

3. Inicie a máquina virtual e faça as configurações:  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente  
  
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente  
Caso o Ubuntu te sugira algum upgrade de versão, ignore.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Mapa

Descrição gerada automaticamente

1. Ao reiniciar a máquina virtual, clique em “dispositivo > “inserir imagem de CD dos adicionais para Convidado...”

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fora da maquina virtual, no Virtual Box:  
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Volte para a máquina virtual, no terminal do Linux execute os comandos nessa ordem
   1. sudo apt update
   2. sudo apt upgrade
   3. sudo apt install build-essential gcc make perl dkms curl tcl
2. Instale o VS Code ou a IDE de edição de código que mais te agrade:   
   Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente
3. Clique no desenho do “CD” na barra de trabalho, abra um terminal no local que o SO irá abrir para você:   
   Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente

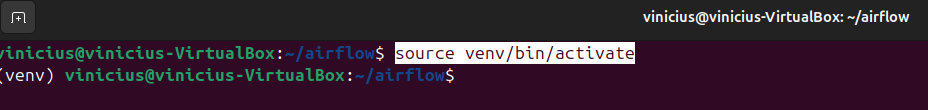
Execute o comando: sudo ./VboxLinuxAdditions.run

1. Crie a pasta da instalação do Apache Airflow na máquina:  
   Texto

   Descrição gerada automaticamente
2. Instale o PPA Deadsnakes para pegar a versão adequada do Python:
   1. sudo add-apt-repository ppa:deadsnakes/ppa
   2. sudo apt install python3.10-venv python3.10-dev python3.10-distutils
   3. sudo apt update
   4. sudo apt install python3.10

Texto

Descrição gerada automaticamente

1. Instalando o Airflow:
   1. cd airflow
   2. python3.10 -m venv venv
   3. source venv/bin/activate  
      
   4. pip install 'apache-airflow==2.8.0' --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.8.0/constraints-3.10.txt"
   5. curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py python get-pip.py
   6. sudo apt install python3-pip
   7. pip install pandas
   8. pip install numpy
   9. pip install pyspark
   10. apache-airflow-providers-apache-spark
   11. sudo apt update
   12. sudo aptupgrade
   13. sudo apt install openjdk-11-jdk
   14. export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64
   15. export PATH=$JAVA\_HOME/bin:$PATH
   16. source ~/.profile
   17. echo $JAVA\_HOME
   18. source venv/bin/activate
2. Lembre-se de sempre exportar a variável “java home” antes de usar o spark
3. Sempre antes de usar o airflow, exporte a variável “airflow home”:
   1. export AIRFLOW\_HOME=/home/vinicius/airflow
   2. Suba o serviço do Airflow: airflow standalone  
      Texto

      Descrição gerada automaticamente com confiança baixa
   3. Abra um novo terminal:
      1. cd airflow
      2. mkdir dags  
         Texto

         Descrição gerada automaticamente
      3. sudo apt install python3-pip
      4. mkdir scripts
   4. Abra um novo terminal para criar a estrutura de pastas para os dados de entrada e saída:
      1. mkdir fs
      2. cd fs
      3. mkdir in
      4. mkdir out  
         Texto

         Descrição gerada automaticamente
4. Reinicie sua máquina virtual e siga os comandos para instalar o Spark usando o Homebrew (gerenciador de pacotes):
   1. sudo apt update
   2. sudo apt install build-essential procps curl file git
   3. /bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"
   4. echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"' >> ~/.profile eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"
   5. source ~/.profile
   6. brew --version
   7. brew install apache-spark
   8. spark-shell  
      Texto

      Descrição gerada automaticamente  
      Texto

      Descrição gerada automaticamente

Seu ambiente está configurado com o Spark pronto para executar jobs desenvolvidos em Scala ou em Pyspark!

Instalando o Hadoop:

1. Abra o terminal:
   1. wget <https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz>
   2. tar -xzf hadoop-3.3.6.tar.gz
   3. sudo mv hadoop-3.3.6 /usr/local/hadoop
   4. export HADOOP\_HOME=/usr/local/hadoop
   5. export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop
   6. export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop
   7. export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin
   8. source ~/.bashrc
2. Abra o arquivo de configurações do hadoop:
   1. nano $HADOOP\_CONF\_DIR/core-site.xml
   2. adicione o seguinte xml dentro do arquivo:  
      <configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>  
Texto

Descrição gerada automaticamente

* 1. Salve as modificações
  2. Abra o arquivo nano $HADOOP\_CONF\_DIR/hdfs-site.xml  
     <configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///usr/local/hadoop/hdfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///usr/local/hadoop/hdfs/datanode</value>

</property>

</configuration>  
Texto

Descrição gerada automaticamente

* 1. Abra o arquivo nano $HADOOP\_CONF\_DIR/yarn-site.xml  
       
     <configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>localhost:8025</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>localhost:8030</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>localhost:8050</value>

</property>

</configuration>  
Texto

Descrição gerada automaticamente

* 1. nano /usr/local/hadoop/etc/hadoop/mapred-site.xml  
       
     <configuration>

<property>

<name>mapreduce.framework.name</name>

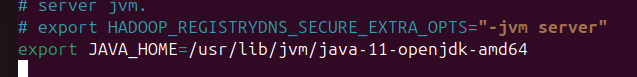
<value>yarn</value>

</property>

</configuration>  
Texto

Descrição gerada automaticamente

1. No terminal, execute os seguintes comandos:
   1. update-alternatives --config java
   2. nano ~/.bashrc
   3. Ao abrir o arquivo anterior no editor nano, adicione as seguintes linhas:
      1. export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64
      2. export PATH=$PATH:$JAVA\_HOME/bin
      3. export HADOOP\_HOME=/usr/local/hadoop
      4. export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop
      5. export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin  
         Texto

         Descrição gerada automaticamente
      6. Salve e feche o arquivo
   4. source ~/.bashrc
   5. echo $JAVA\_HOME
   6. hdfs namenode -format
   7. sudo apt install openssh-server
   8. sudo service ssh start
   9. ssh-keygen -t rsa -P "" -f ~/.ssh/id\_rsa
   10. cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys
   11. chmod 777 ~/.ssh/authorized\_keys
   12. chmod 700 ~/.ssh
   13. chmod 600 ~/.ssh/id\_rsa
   14. chmod 644 ~/.ssh/id\_rsa.pub
   15. chmod 600 ~/.ssh/authorized\_keys
2. Abra o arquivo de configurações do hadoop para parametrizar o java:
   1. nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
   2. Adicione a seguinte linha e salve o arquivo no editor de texto:
      1. export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64  
         
3. Inicie o serviço do hadoop e do yarn:
   1. start-dfs.sh
   2. start-yarn.sh

Texto

Descrição gerada automaticamente

**Explicando os scripts:**

1. 00\_massa\_dados.py 🡪 Script responsável por gerar os dados “fake” que serão usados nesse laboratório, tendo como saída 3 arquivos CSVs. Ele deve ser executado dentro do ambiente “venv”:
   1. usuarios.csv
   2. produtos.csv
   3. vendas.csv
2. 01\_processa\_vendas\_spark.py 🡪 Script responsável por criar uma sessão Spark, irá ler os dados em CSV no diretório “home/vinicius/fs/in/”, processará usando o Apache Spark, depositará os dados em formato parquet no diretório “/home/vinicius/fs/out”

**Desafio proposto:**

Agora que você tem o Apache Airflow (orquestrador dos Jobs e das dags) em sua máquina local, não se faz mais necessário qualquer interação com a AWS para processar e armazenar os dados usando o Apache Spark. Dado esse cenário, desenvolva uma dag para ser executada no Apache Airflow com os seguintes passos:

1. Gerar a massa de dados com base no script “00\_massa\_dados.py”, simulando um sistema que disponibiliza os dados de vendas no caminho “/home/vinicius/fs/in” no diretório local.
2. Através do Airflow, crie uma task usando o BashOperator para copiar os arquivos do disco local para dentro do cluster hadoop que você configurou. Esses comandos vão ajudar:
   1. /usr/local/hadoop/bin/hdfs dfs -copyFromLocal -f /home/vinicius/fs/in/vendas.csv /home/vinicius/fs/in/
   2. /usr/local/hadoop/bin/hdfs dfs -copyFromLocal -f /home/vinicius/fs/in/produtos.csv /home/vinicius/fs/in/
   3. /usr/local/hadoop/bin/hdfs dfs -copyFromLocal -f /home/vinicius/fs/in/usuarios.csv /home/vinicius/fs/in/
3. Usar o Apache Spark para processar esses dados e convertê-los em parquet usando como base o script “01\_processa\_vendas\_spark.py”. Lembre-se que você deve chamar o Spark através de um comando conhecido como Sparksubmit dentro do Airflow.
4. Os arquivos parquet serão processados pelo Spark e gravados dentro do hdfs. Crie uma task no airflow para fazer a cópia do HDFS para o disco local.

Você pode fazer sua dag do zero, e caso precise de alguma ajuda recorra ao script “01\_processa\_vendas.py” na pasta dags.

Lembre-se:

1. Sempre exporte as variáveis do airflow\_home quando iniciar o ambiente venv
2. Sempre inicie os serviços do hadoop e yarn
3. Suas dags devem estar dentro da pasta “dags” da instalação do Apache Airflow para que o serviço consiga executar os scripts