VINÍCIUS HENRIQUE DOS SANTOS

MACHINE LEARNING ENGINEERING

BIG DATA PIPELINES

# AULA 05

**SUMÁRIO**

## O QUE VEM POR AÍ?

Você está prestes a conhecer o Apache Nifi, a ferramenta capaz de se conectar e processar TUDO! Ele vem ganhando cada vez mais espaço no mercado por conta da sua grande versatilidade, tolerância a falhas e poder de processamento quando falamos de dados em batch ou tempo real. Aperte os cintos e prepare-se para conhecê-lo de forma prática.

Nós não pararemos por aí: vou te apresentar uma das soluções mais fortes do mercado quando pensamos em bancos de dados de documentos, o ElasticSearch. Essa ferramenta é capaz de indexar, processar e buscar documentos em frações de segundos e nessa aula você aprenderá como inserir, processar e usar os seus documentos dentro dela!

## HANDS ON

Realize as configurações para iniciar os serviços do Apache NIFI, Elasticsearch e Kibana.

Através do Apache NIFI, realize uma movimentação de arquivos bidirecionais:

- Da sua máquina para algum Storage Cloud (do provedor que você tiver mais familiaridade).
- Movimente um arquivo do storage da cloud de sua preferência para a sua máquina local.

Através do Kibana, interaja com o Elasticsearch com o seguinte objetivo:

- Adicione um documento.
- Consulte esse documento.
- Altere esse documento.

## SAIBA MAIS

### Configurando o ambiente para o Apache NiFi

Faça o downlaod do virtualizador de sua preferência; aqui eu seguirei com o Virtual Box, que 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 (a mais recente na data em que esse curso está sendo desenvolvido).

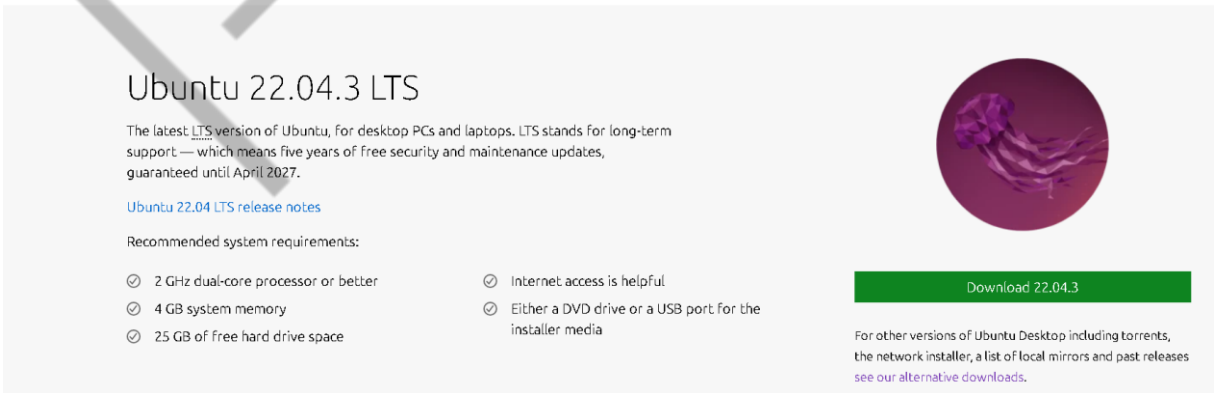


Figura 2 – Ubuntu 22.04.3 LTS
Fonte: elaborado pelo autor (2024)

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:

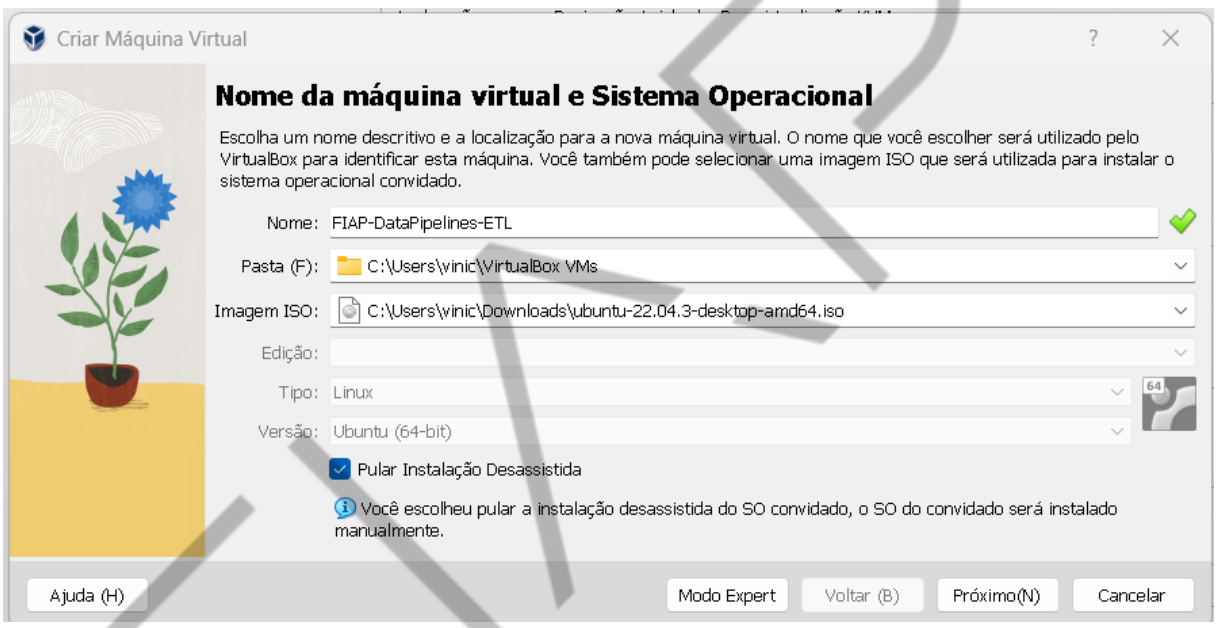


Figura 4 – Nome da máquina virtual e Sistema Operacional
Fonte: elaborado pelo autor (2024)

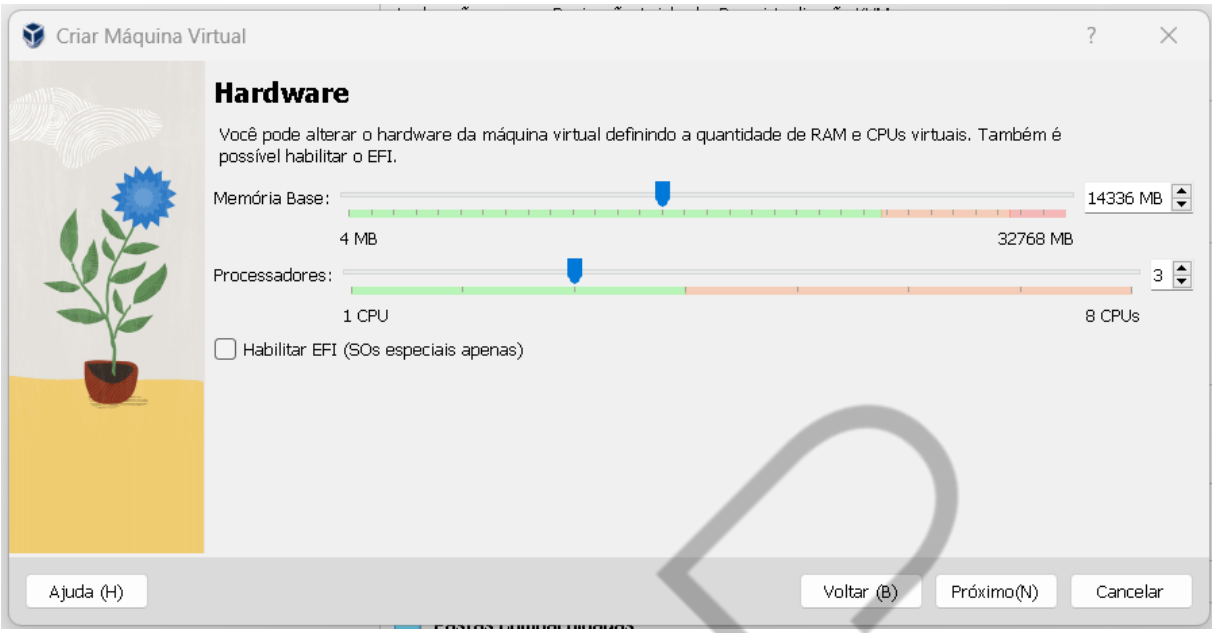


Figura 5 – Hardware
Fonte: elaborado pelo autor (2024)

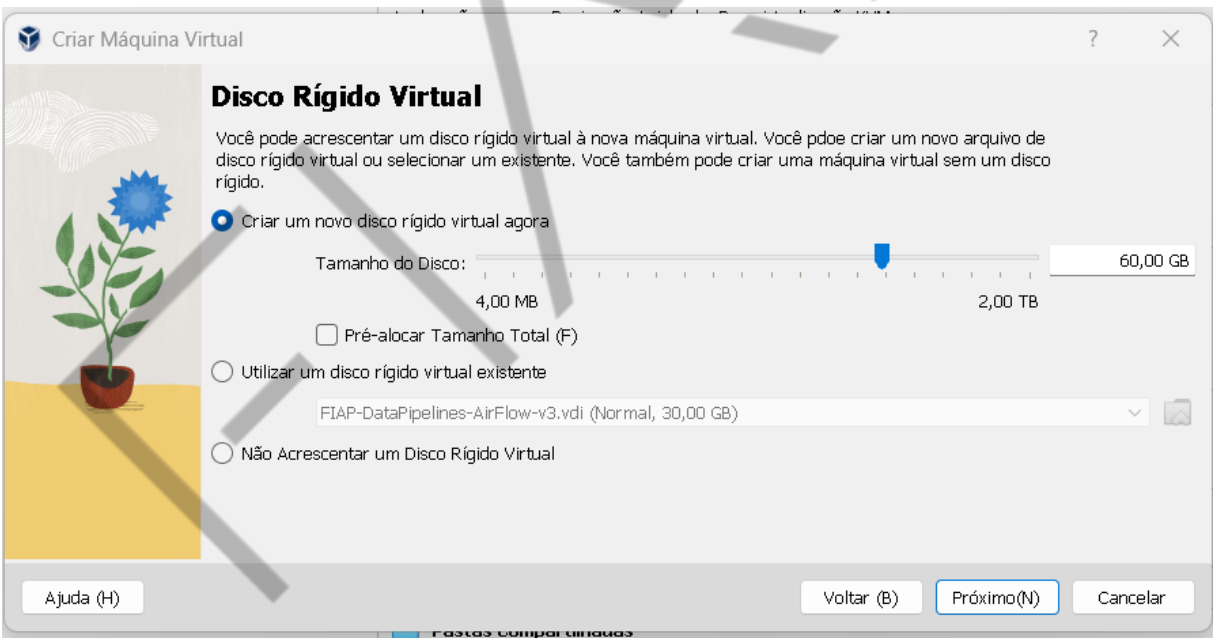


Figura 6 – Disco Rígido Virtual
Fonte: elaborado pelo autor (2024)

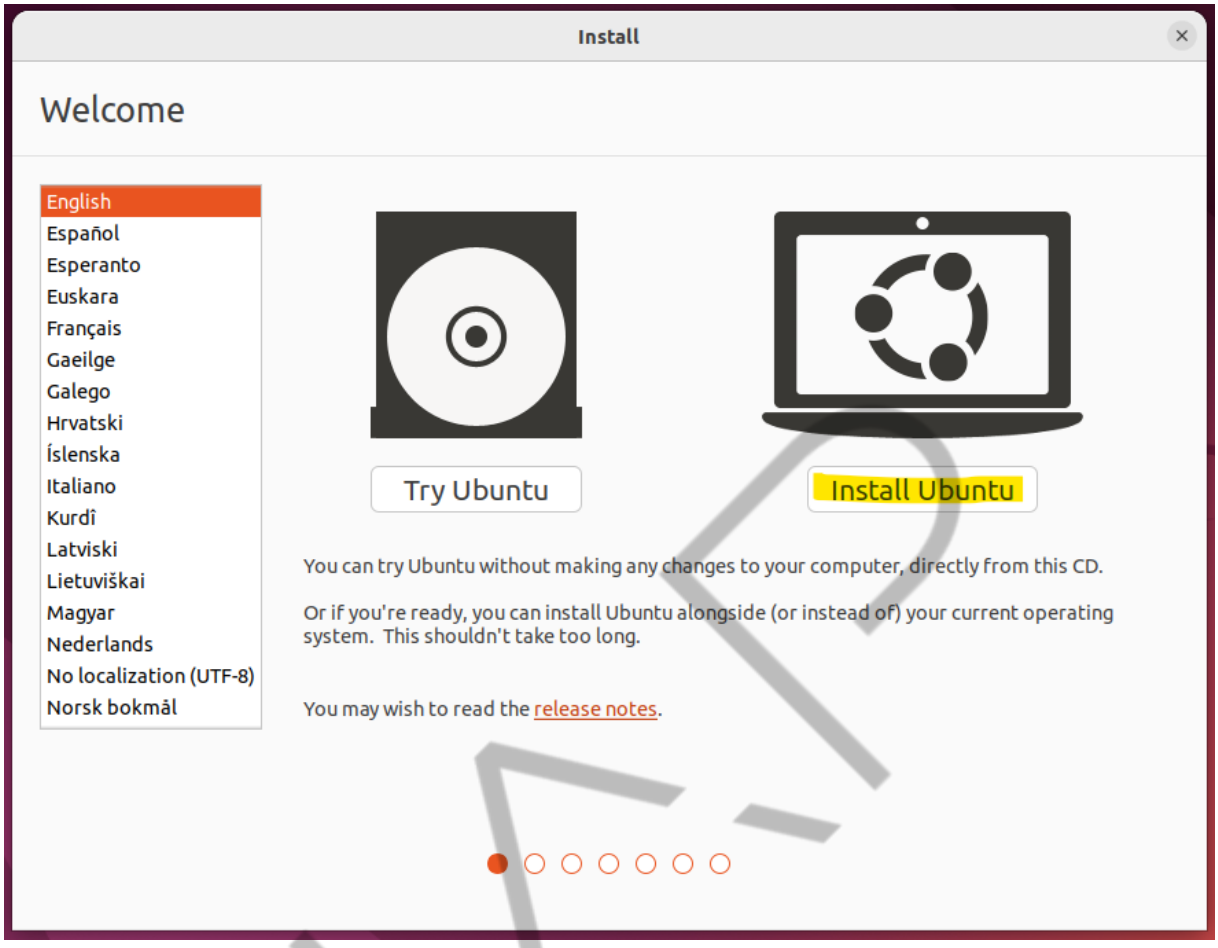


Figura 7 – Install Ubuntu
Fonte: elaborado pelo autor (2024)

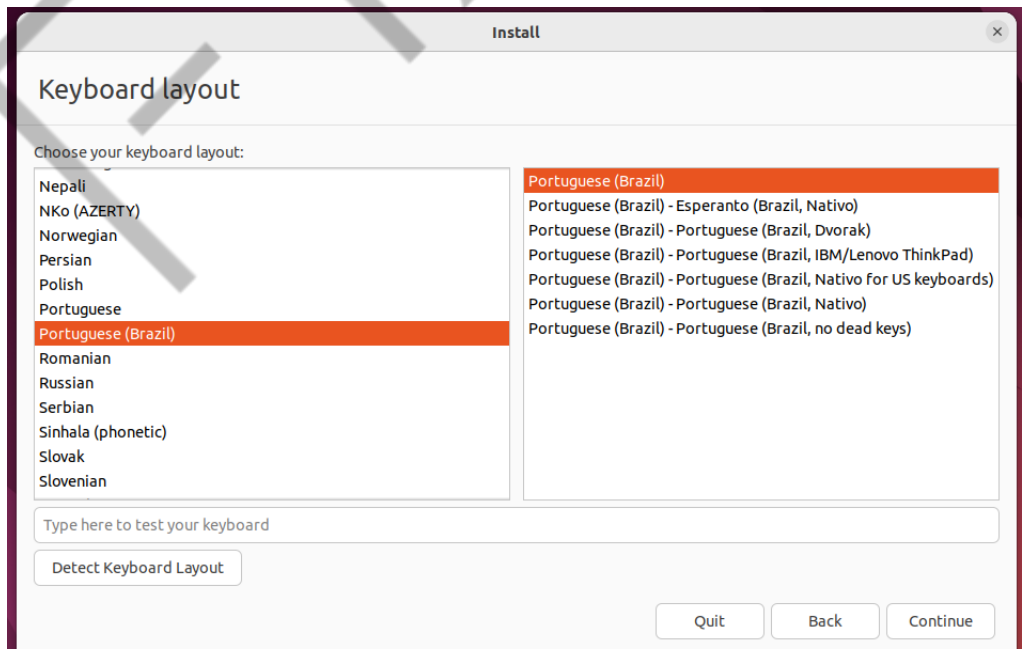


Figura 8 – Escolha da linguagem
Fonte: elaborado pelo autor (2024)

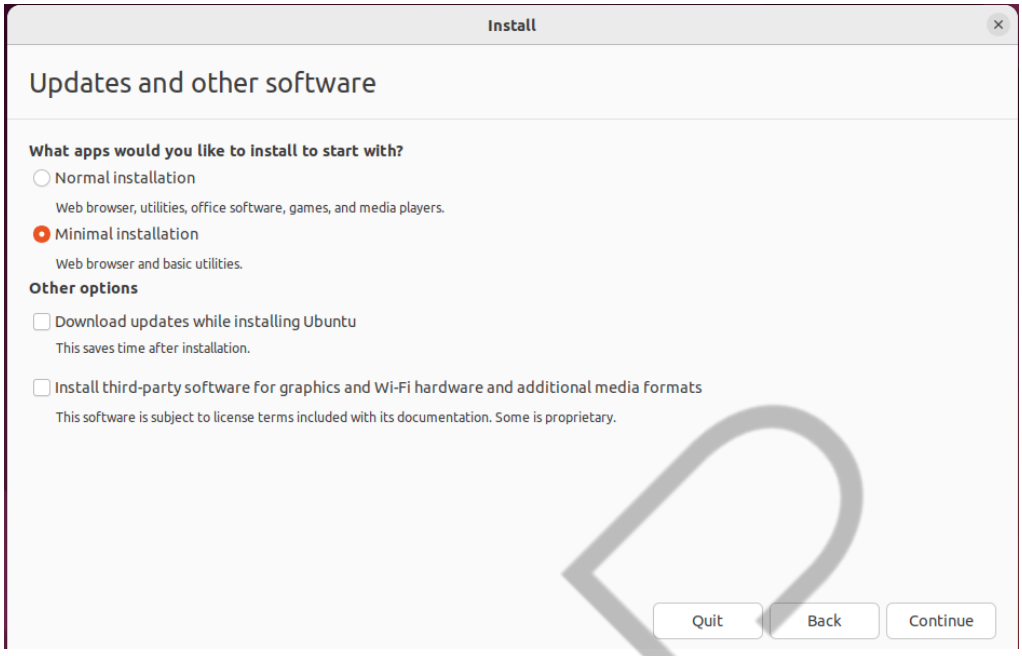


Figura 9 – Instalação mínima
Fonte: elaborado pelo autor (2024)

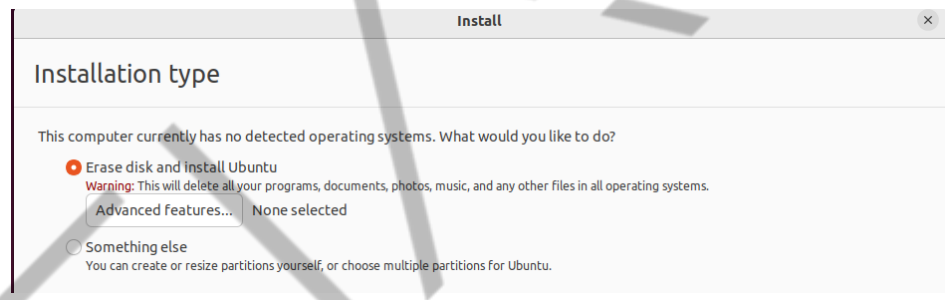


Figura 10 – Tipo de instalação
Fonte: elaborado pelo autor (2024)

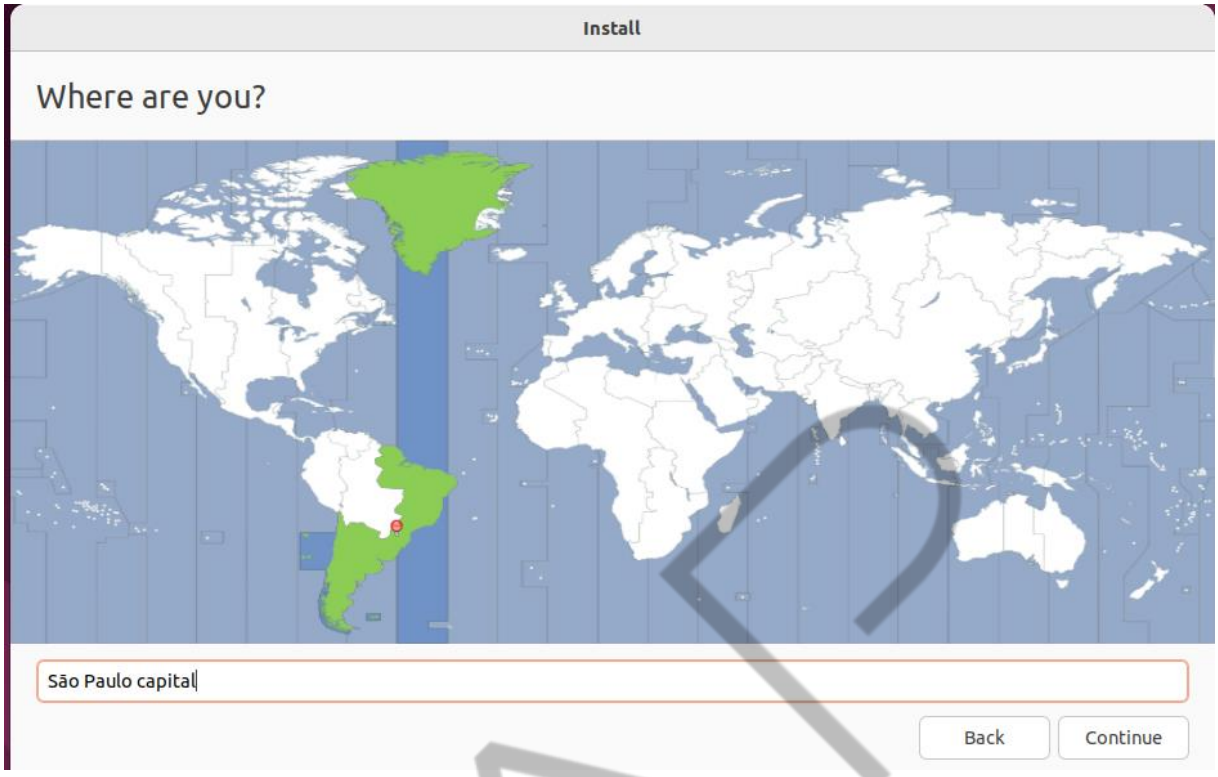


Figura 11 – Localização
Fonte: elaborado pelo autor (2024)

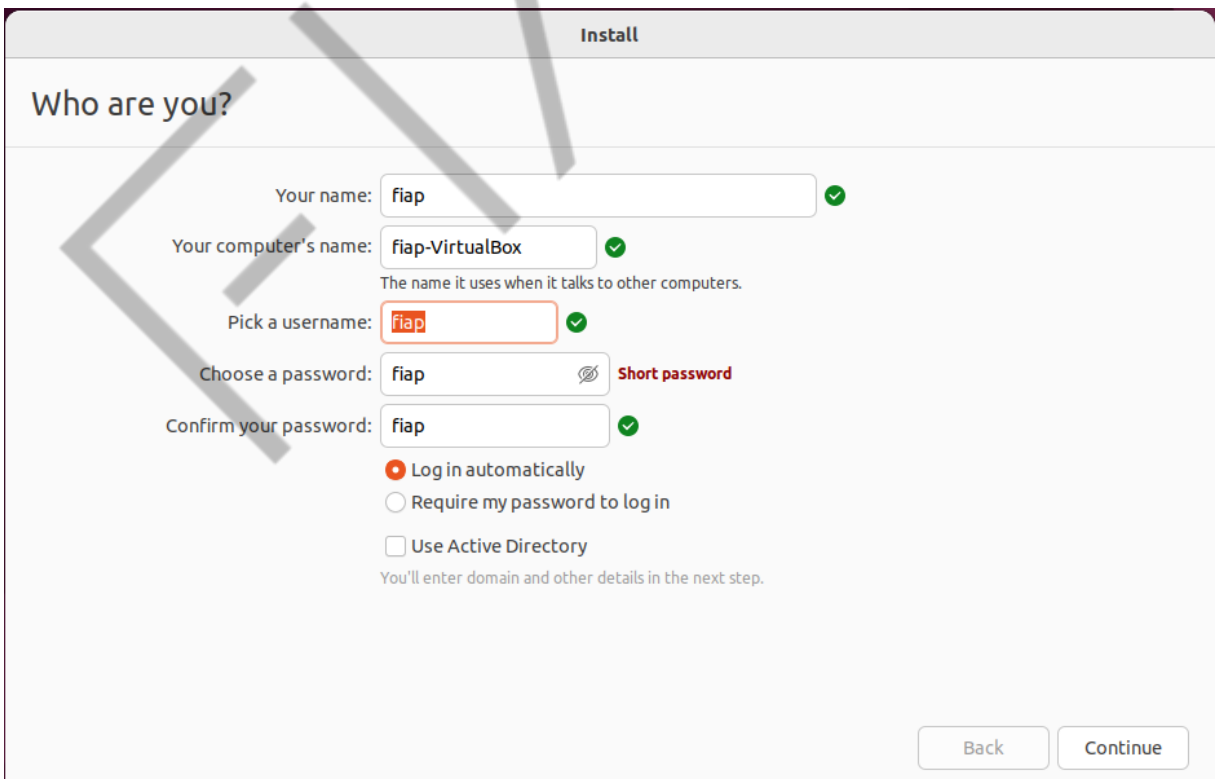


Figura 12 – Escolhendo um usuário
Fonte: elaborado pelo autor (2024)

Agora que o Ubuntu iniciou, vamos atualizar os pacotes e garantir que tudo que precisamos está com a versão atualizada.

sudo apt update

sudo apt upgrade

Depois que os arquivos do sistema operacional forem copiados e instalados, vá em “Dispositivos” no Virtual box e selecione a opção destacada.

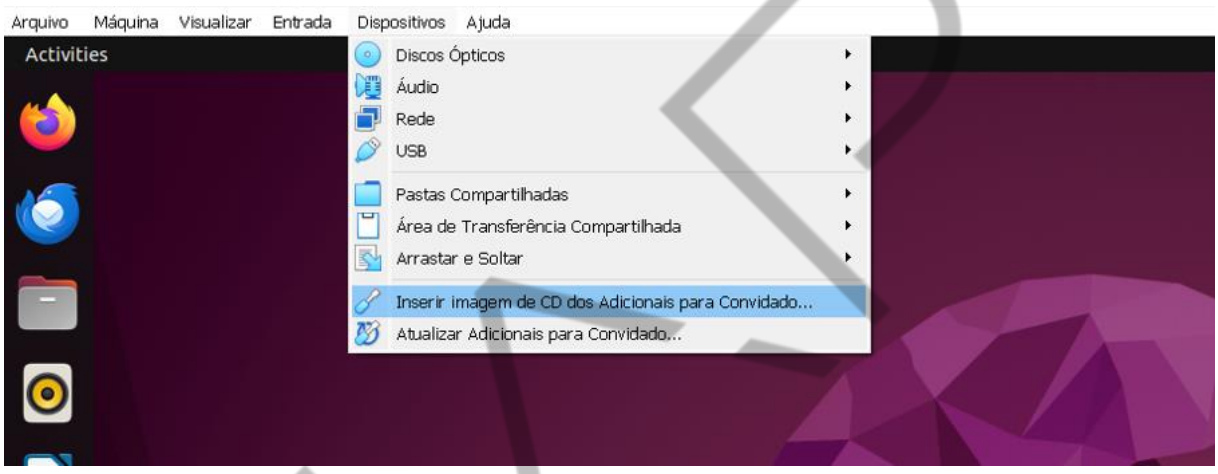


Figura 13 – Inserir Imagem de CD dos Adicionais para Convidado
Fonte: elaborado pelo autor (2024)

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 14 e inicie a máquina novamente.

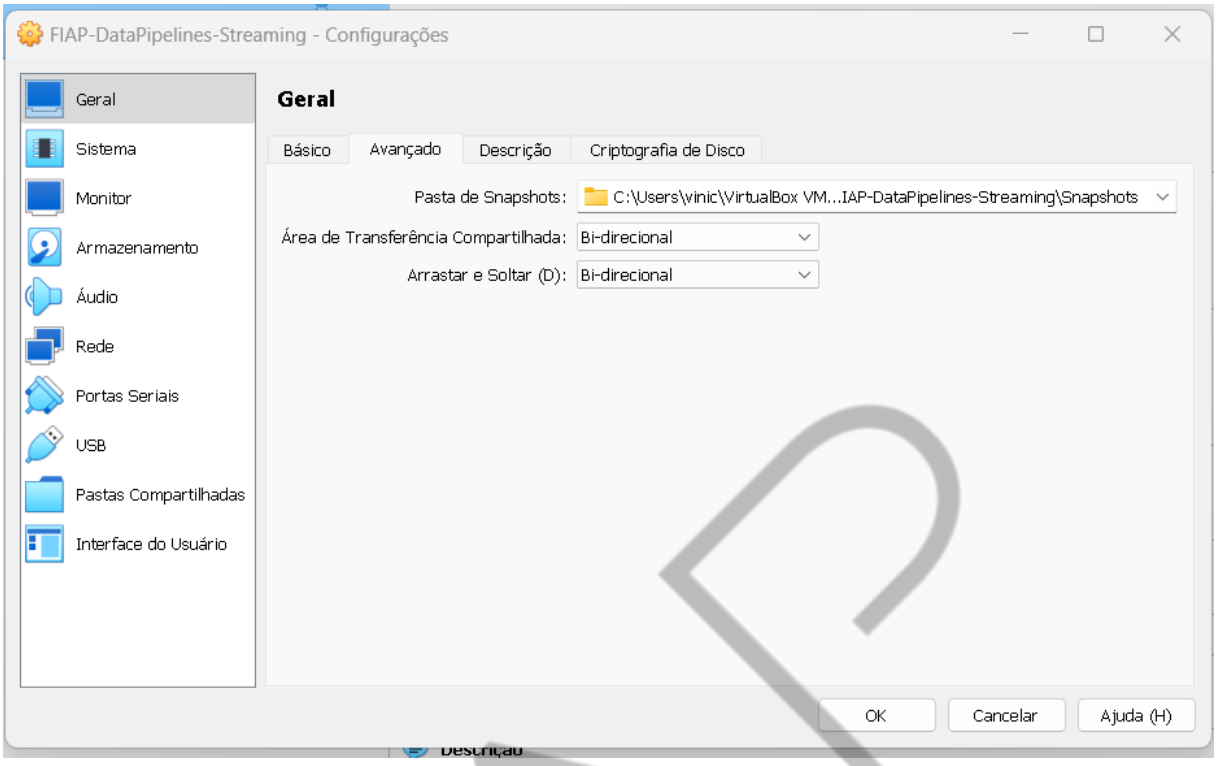


Figura 14 – Parâmetros indicados
Fonte: elaborado pelo autor (2024)

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

sudo apt update

sudo apt upgrade

sudo apt install build-essential gcc make perl dkms curl tcl

Clique na imagem do “CD” que apareceu na barra de ferramentas no Ubuntu.


Figura 15 – Ícone de CD
Fonte: elaborado pelo autor (2024)

Em seguida, na janela, abra um terminal dentro dessa pasta:

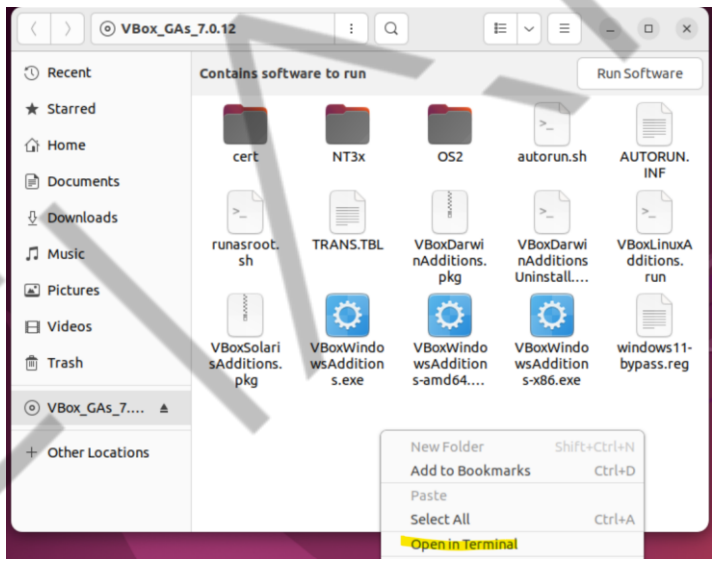


Figura 16 – Abrindo um terminal
Fonte: elaborado pelo autor (2024)

Em seguida, execute esses comandos no terminal:

sudo ./VboxLinuxAdditions.run

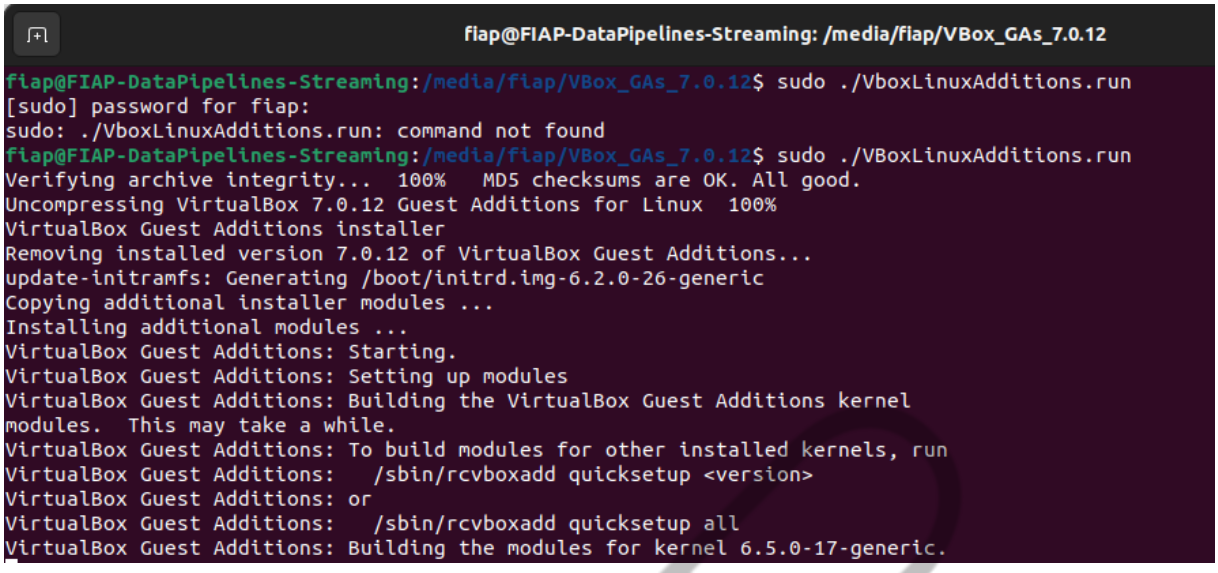


Figura 17 – Execução de comandos no terminal
Fonte: elaborado pelo autor (2024)

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

Abra um novo terminal e execute o seguinte comando: sudo apt install default-jdk

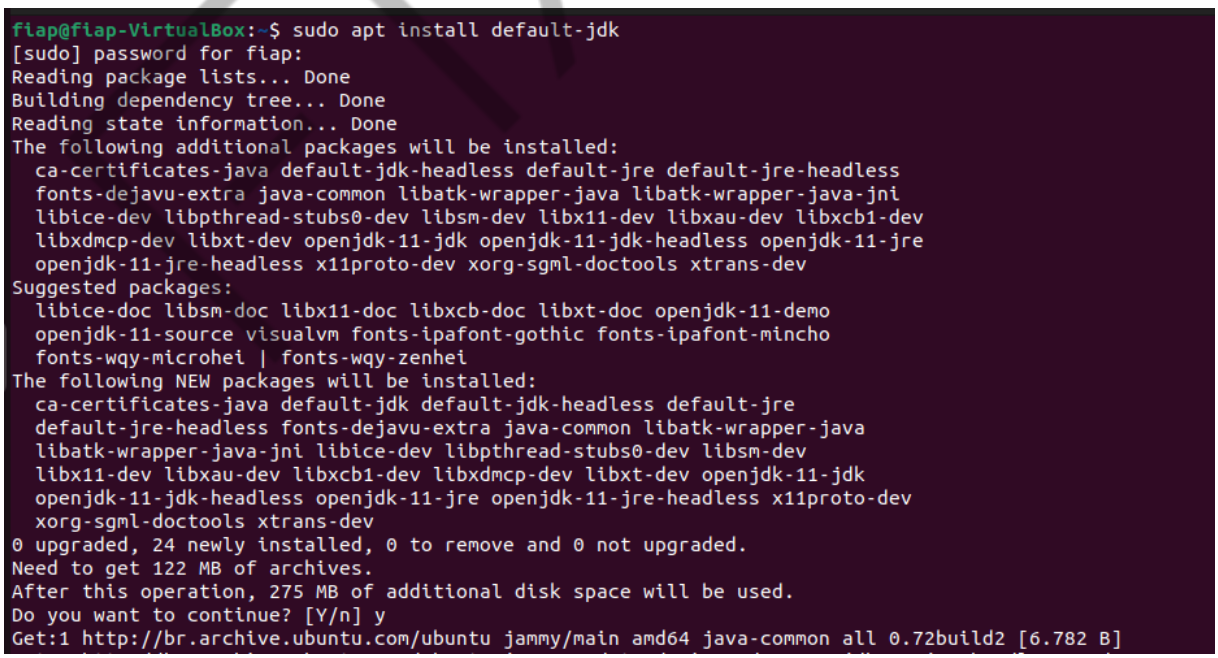


Figura 18 – Comando sudo apt install default-jdk
Fonte: elaborado pelo autor (2024)

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

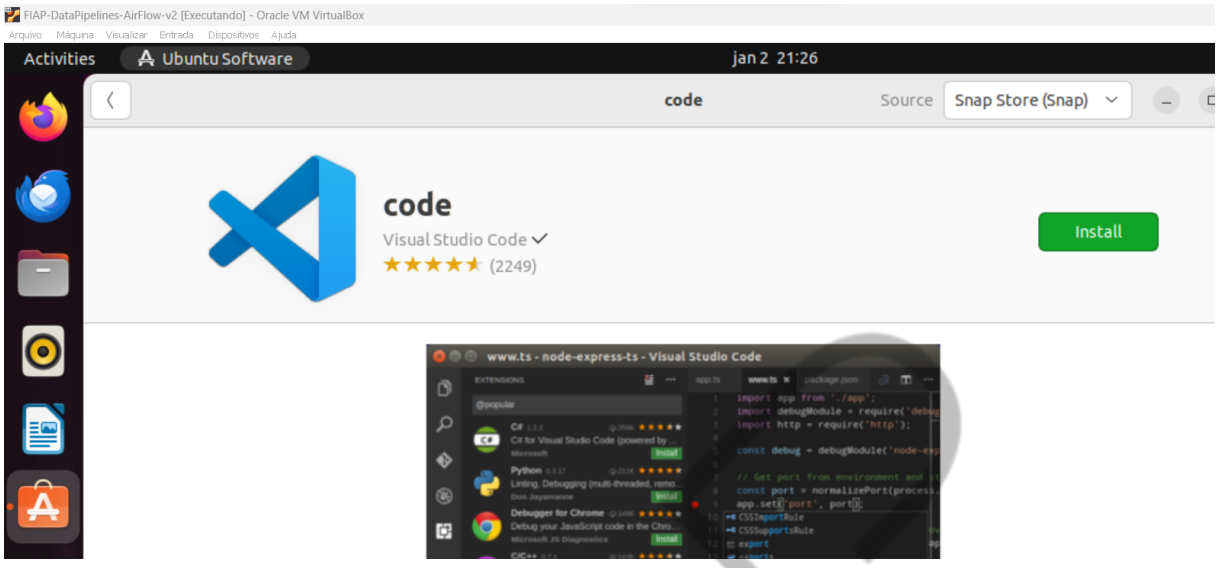


Figura 19 – Instalando o VSCode
Fonte: elaborado pelo autor (2024)

Instale o Docker:

sudo apt-get update

```
fiap@fiap-VirtualBox:~$ sudo apt-get update
[sudo] password for fiap:
Hit:1 http://br.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://br.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://br.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (83,5 kB/s)
Reading package lists... Done
fiap@fiap-VirtualBox:~$
```

Figura 20 – Instalando o Docker
Fonte: elaborado pelo autor (2024)

sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg \

lsb-release

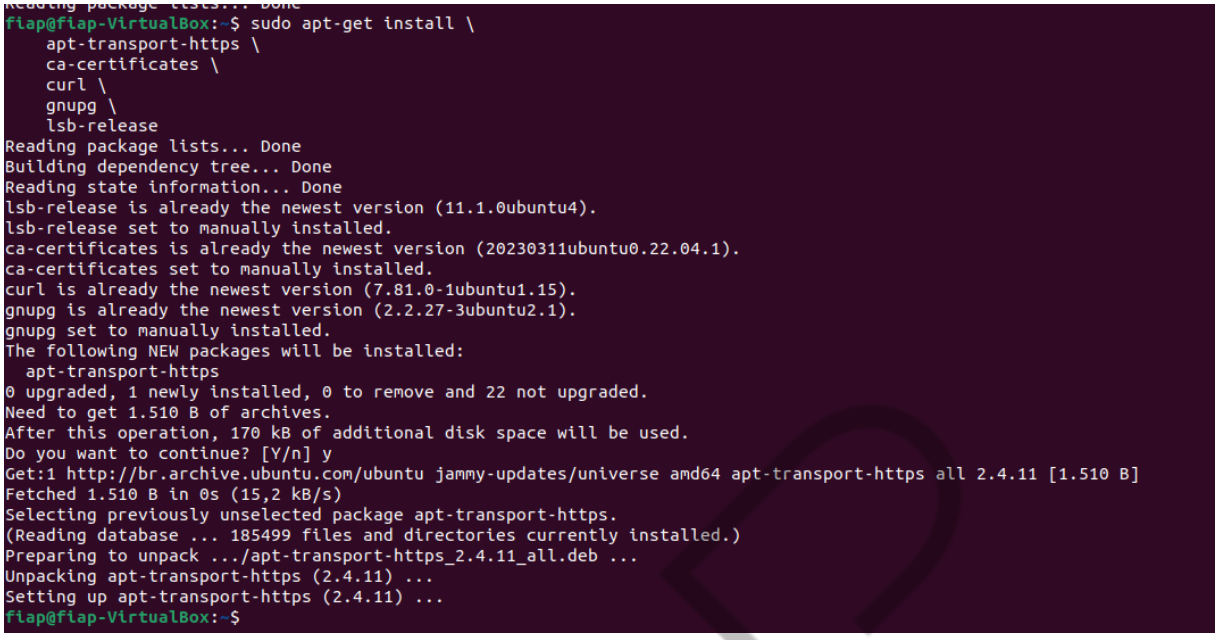


Figura 21 – Docker (1)
Fonte: elaborado pelo autor (2024)

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

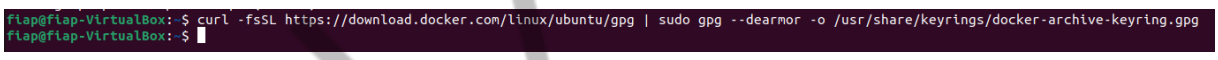


Figura 22 – Docker (2)
Fonte: elaborado pelo autor (2024)

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

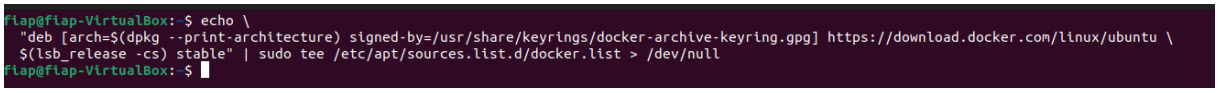


Figura 23 – Docker (3)
Fonte: elaborado pelo autor (2024)

sudo apt-get update

```
fiap@fiap-VirtualBox:~$ sudo apt-get update
Hit:1 http://br.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://br.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://br.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48,8 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [27,6 kB]
Fetched 76,4 kB in 1s (56,6 kB/s)
Reading package lists... Done
```

Figura 24 – Docker (4)
Fonte: elaborado pelo autor (2024)

sudo apt-get install docker-ce docker-ce-cli containerd.io

```
fiap@fiap-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 22 not upgraded.
Need to get 123 MB of archives.
After this operation, 445 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.28-1 [29,6 MB]
Get:3 http://br.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Get:4 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Get:5 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.10 [3.166 kB]
Get:6 http://br.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61,5 kB]
Get:7 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28,2 kB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.13.0-1~ubuntu.22.04~jammy [29,5 MB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:25.0.4-1~ubuntu.22.04~jammy [13,7 MB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:25.0.4-1~ubuntu.22.04~jammy [24,3 MB]
Get:11 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:25.0.4-1~ubuntu.22.04~jammy [9.320 kB]
Get:12 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.24.7-1~ubuntu.22.04~jammy [12,1 MB]
Fetched 123 MB in 24s (5.080 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 185503 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../01-containerd.io_1.6.28-1_amd64.deb ...
Unpacking containerd.io (1.6.28-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../02-docker-buildx-plugin_0.13.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-buildx-plugin (0.13.0-1~ubuntu.22.04~jammy) ...
```

Figura 25 – Docker (5)
Fonte: elaborado pelo autor (2024)

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

```
fiap@fiap-VirtualBox:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 12.1M  100 12.1M    0     0  1638k      0  0:00:07  0:00:07 --:--:-- 1460k
fiap@fiap-VirtualBox:~$
```

Figura 26 – Docker (6)
Fonte: elaborado pelo autor (2024)

sudo chmod +x /usr/local/bin/docker-compose

```
fiap@fiap-VirtualBox:~$ sudo chmod +x /usr/local/bin/docker-compose
fiap@fiap-VirtualBox:~$
```

Figura 27 – Docker (7)
Fonte: elaborado pelo autor (2024)

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

```
fiap@fiap-VirtualBox:~$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Figura 28 – Docker (8)
Fonte: elaborado pelo autor (2024)

docker-compose --version

```
fiap@fiap-VirtualBox:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

Figura 29 – Docker (9)
Fonte: elaborado pelo autor (2024)

Crie uma pasta chamada “nifi” e abra ela usando o VS Code. Além disso, crie um arquivo chamado “docker-compose.yml” com o seguinte conteúdo:

```yaml
version: '3'
version: '3'

services:
  nifi:
    cap_add:
      - NET_ADMIN # low port bindings
    image: apache/nifi
    container_name: nifi
    environment:
      - SINGLE_USER_CREDENTIALS_USERNAME=fiap
      - SINGLE_USER_CREDENTIALS_PASSWORD=fiap123456
      - NIFI_WEB_HTTP_PORT=8080
      - NIFI_WEB_HTTP_HOST=0.0.0.0
    ports:
      - "8080:8080/tcp" # HTTP interface
      - "8443:8443/tcp" # HTTPS interface
      - "514:514/tcp" # Syslog
      - "514:514/udp" # Syslog
      - "2055:2055/udp" # NetFlow
    volumes:
      - nifi-conf:/opt/nifi/nifi-current/conf
      - nifi_flowfile:/opt/nifi/nifi-current/flowfile_repository
      - nifi_content:/opt/nifi/nifi-current/content_repository
      - /home/fiap/nifi/arquivos:/opt/nifi/nifi-current/arquivos
```

```
    restart: unless-stopped
```

Figura 30 – docker-compose.yml (1)
Fonte: elaborado pelo autor (2024)

```
volumes:
  nifi-conf:
  nifi_flowfile:
  nifi_content:
```

Figura 31 – volumes
Fonte: elaborado pelo autor (2024)

docker-compose.yml

```
version: '3'

services:
  nifi:
    cap_add:
      - NET_ADMIN # low port bindings
    image: apache/nifi
    container_name: nifi
    environment:
      - SINGLE_USER_CREDENTIALS_USERNAME=fiap
      - SINGLE_USER_CREDENTIALS_PASSWORD=fiap123456
      - NIFI_WEB_HTTP_PORT=8080
      - NIFI_WEB_HTTP_HOST=0.0.0.0
    ports:
      - "8080:8080/tcp" # HTTP interface
      - "8443:8443/tcp" # HTTPS interface
      - "514:514/tcp" # Syslog
      - "514:514/udp" # Syslog
      - "2055:2055/udp" # NetFlow
    volumes:
      - nifi-conf:/opt/nifi/nifi-current/conf
      - nifi_flowfile:/opt/nifi/nifi-current/flowfile_repository
      - nifi_content:/opt/nifi/nifi-current/content_repository
      - /home/fiap/nifi/arquivos:/opt/nifi/nifi-current/arquivos
    restart: unless-stopped
volumes:
  nifi-conf:
  nifi_flowfile:
  nifi_content:
```

Figura 32 – docker-compose.yml (2)
Fonte: elaborado pelo autor (2024)

Dentro da pasta “nifi”, crie uma segunda pasta chamada “arquivos”. E dentro dessa, cole os 3 arquivos csv que recebeu e crie uma outra pasta chamada “destino”:

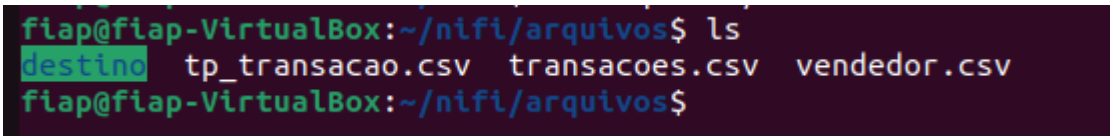


Figura 33 – Pasta “destino”
Fonte: elaborado pelo autor (2024)

Dentro dessa pasta crie uma pasta chamada “arquivos”

Execute o comando “sudo docker-compose up -d”

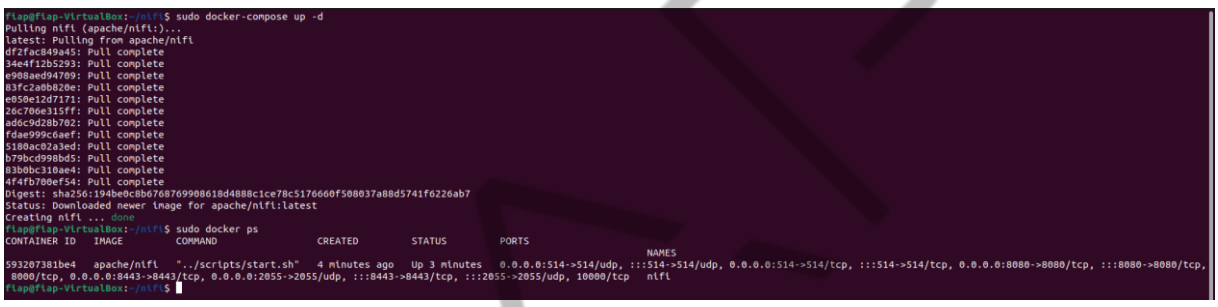


Figura 34 – sudo docker-compose up -d
Fonte: elaborado pelo autor (2024)

Ao concluir, abra o navegador e digite http://localhost:8080/nifi/

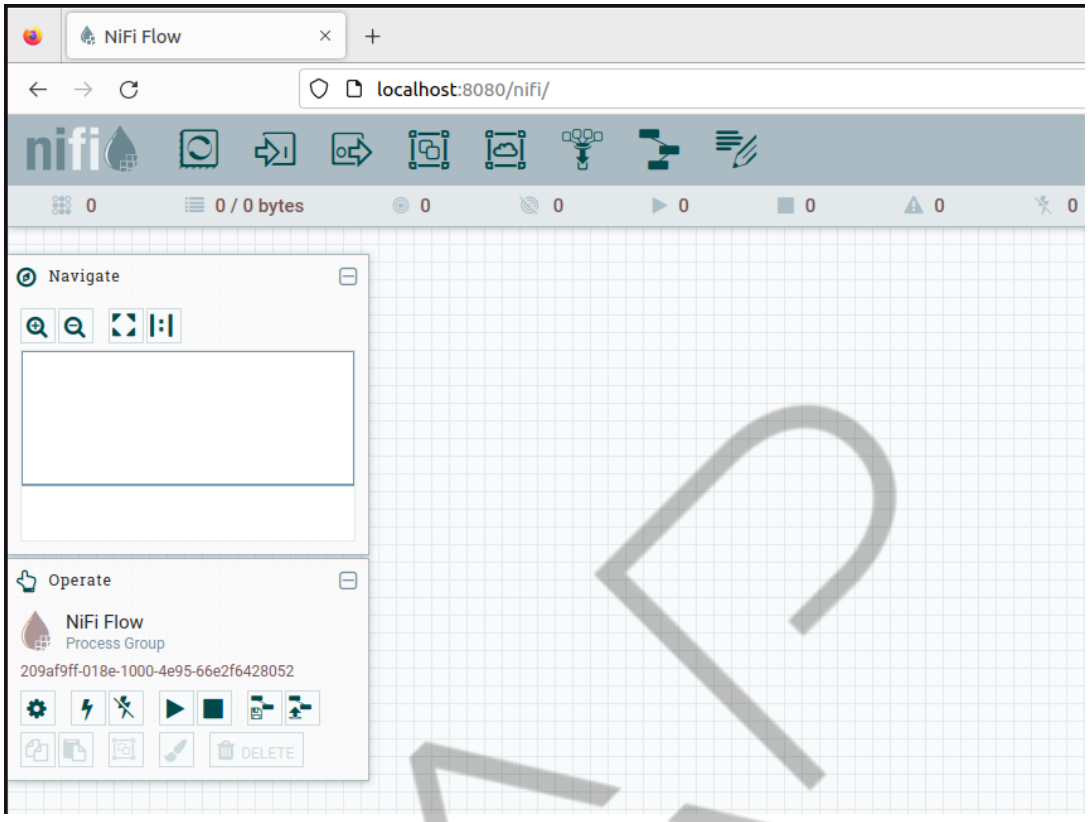


Figura 35 – http://localhost:8080/nifi/
Fonte: elaborado pelo autor (2024)

**Configure a sua conta as ferramentas ELK**

Se estiver com algum container rodando, desligue-o para seguir com as configurações:

sudo docker stop $(sudo docker ps -a -q)

```
fiap@fiap-VirtualBox:~$ sudo docker stop $(sudo docker ps -a -q)
de4827d6e5bb
fiap@fiap-VirtualBox:~$
```

Figura 36 – sudo docker stop $(sudo docker ps -a -q)
Fonte: elaborado pelo autor (2024)

Crie uma pasta chamada “elk”:

```
fiap@fiap-VirtualBox:~$ mkdir elk
fiap@fiap-VirtualBox:~$
```

Figura 37 – Pasta “elk”
Fonte: elaborado pelo autor (2024)

Entre na pasta “elk” e crie as seguintes pastas:

```
fiap@fiap-VirtualBox:~/elk$ mkdir elasticsearch
fiap@fiap-VirtualBox:~/elk$ mkdir kibana
```

Figura 38 – Elasticsearch e Kibana
Fonte: elaborado pelo autor (2024)

Abra essa pasta pelo VS Code e crie um arquivo chamado “docker-compose.yml”.

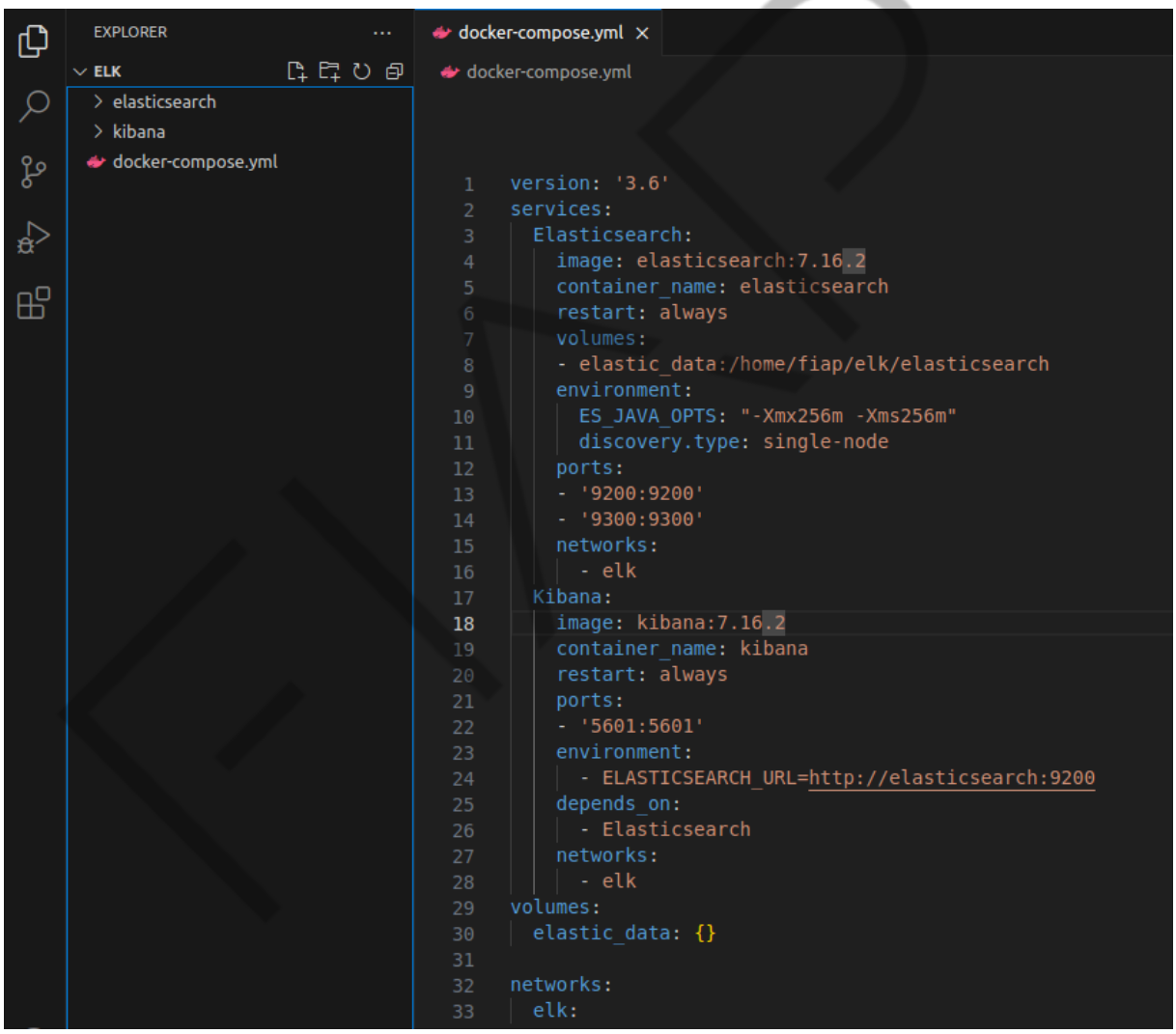


Figura 39 – docker-compose.yml (1)
Fonte: elaborado pelo autor (2024)

```
version: '3.6'
services:
  Elasticsearch:
    image: elasticsearch:7.16.2
    container_name: elasticsearch
```

```
    restart: always
    volumes:
    - elastic_data:/home/fiap/elk/elasticsearch
    environment:
      ES_JAVA_OPTS: "-Xmx256m -Xms256m"
      discovery.type: single-node
    ports:
    - '9200:9200'
    - '9300:9300'
    networks:
      - elk
  Kibana:
    image: kibana:7.16.2
    container_name: kibana
    restart: always
    ports:
    - '5601:5601'
    environment:
      - ELASTICSEARCH_URL=http://elasticsearch:9200
    depends_on:
      - Elasticsearch
    networks:
      - elk
volumes:
  elastic_data: {}

networks:
  elk:
```

Figura 40 – docker-compose.yml (2)
Fonte: elaborado pelo autor (2024)

Navegue até a pasta “elk” pelo terminal:

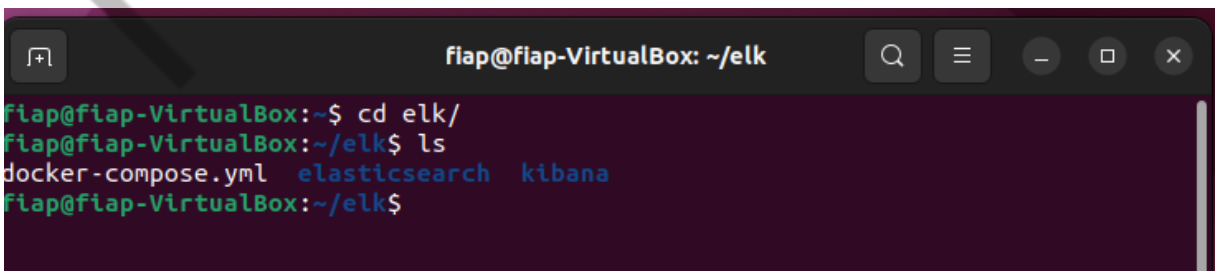


Figura 41 – Pasta elk no terminal
Fonte: elaborado pelo autor (2024)

Execute o seguinte comando:

sudo docker-compose up -d

```
fiap@fiap-VirtualBox:~/elk$ sudo docker-compose up -d
Creating network "elk_elk" with the default driver
Creating volume "elk_elastic_data" with default driver
Pulling Elasticsearch (elasticsearch:7.16.2)...
7.16.2: Pulling from library/elasticsearch
7b1a6ab2e44d: Pulling fs layer
c0f9908af642: Pulling fs layer
7b1a6ab2e44d: Downloading [>                                                  ]  285.9kB/28.57MBwnloading [==================>
                              ]  2.923MB/28.57MBiting
7b1a6ab2e44d: Pull complete
c0f9908af642: Pull complete
04928834f6a2: Pull complete
6f647d55d420: Pull complete
da7afec4a2d7: Pull complete
146b4a7018b9: Pull complete
dc193d8d7c71: Pull complete
e3deb19b2c38: Pull complete
8b79893ce5a3: Pull complete
Digest: sha256:fe7ae76ec33e1222ece5216e3a9c6346999a73d3fb65256abb01638758db4b5d
Status: Downloaded newer image for elasticsearch:7.16.2
Pulling Kibana (kibana:7.16.2)...
7.16.2: Pulling from library/kibana
a1d0c7532777: Pull complete
700212586d98: Pull complete
013f69c97c90: Pull complete
7c92d785e3b2: Pull complete
efddc883940a: Pull complete
05ac9ba18fc0: Pull complete
1d6c552d0781: Pull complete
752fb7e12a50: Pull complete
be34b62839b5: Pull complete
9d7a37976b34: Pull complete
39cd3d8ea5b3: Pull complete
0c75809c03a7: Pull complete
99bff35135e1: Pull complete
Digest: sha256:cbff0e7f8200798130dc9ebca666c89d440f203272d66b007763ef554b21d0f0
Status: Downloaded newer image for kibana:7.16.2
Creating elasticsearch ... done
Creating kibana        ... done
fiap@fiap-VirtualBox:~/elk$
```

Figura 42 – sudo docker-compose up -d
Fonte: elaborado pelo autor (2024)

sudo apt install ufw

```
fiap@fiap-VirtualBox:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
```

Figura 43 – sudo apt install ufw
Fonte: elaborado pelo autor (2024)

sudo ufw enable

```
fiap@fiap-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
fiap@fiap-VirtualBox:~$
```

Figura 44 – sudo ufw enable
Fonte: elaborado pelo autor (2024)

sudo ufw allow 9200/tcp

sudo ufw allow 5601/tcp

sudo ufw allow 9600/tcp

sudo ufw allow 9300/tcp

```
fiap@fiap-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
fiap@fiap-VirtualBox:~$ sudo ufw allow 9200/tcp
Rule added
Rule added (v6)
fiap@fiap-VirtualBox:~$ sudo ufw allow 5601/tcp
Rule added
Rule added (v6)
fiap@fiap-VirtualBox:~$ sudo ufw allow 9600/tcp
Rule added
Rule added (v6)
fiap@fiap-VirtualBox:~$ sudo ufw allow 9300/tcp
Rule added
Rule added (v6)
fiap@fiap-VirtualBox:~$
```

Figura 45 – sudo ufw allow
Fonte: elaborado pelo autor (2024)

Agora, digite no navegador: http://localhost:5601

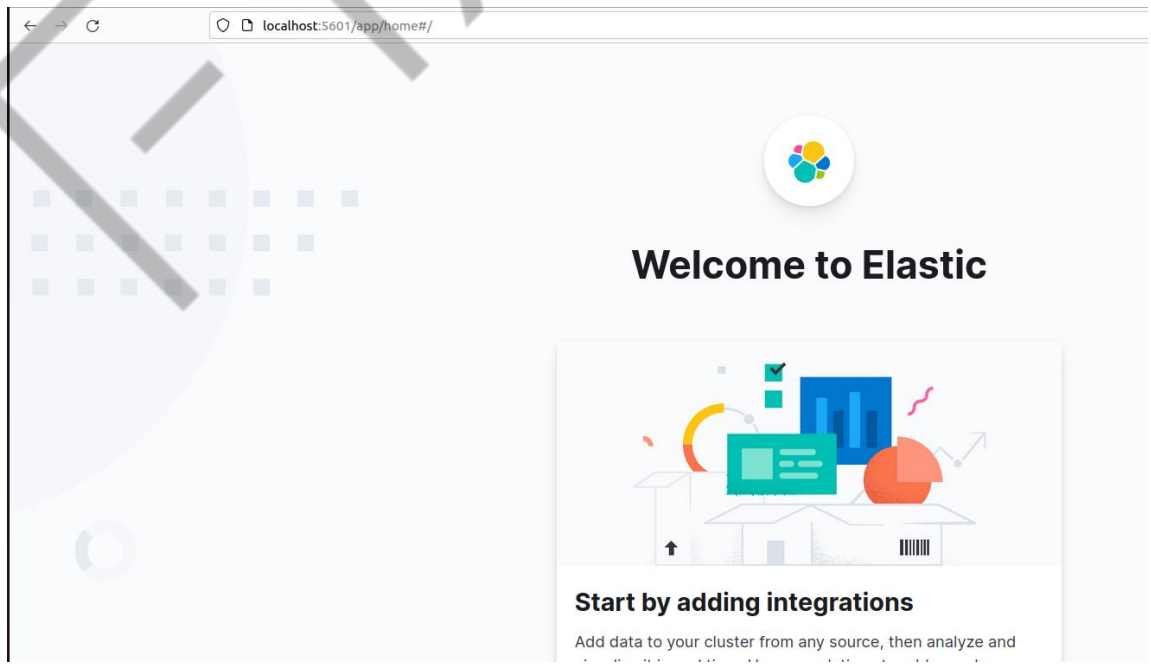


Figura 46 – http://localhost:5601 (1)
Fonte: elaborado pelo autor (2024)

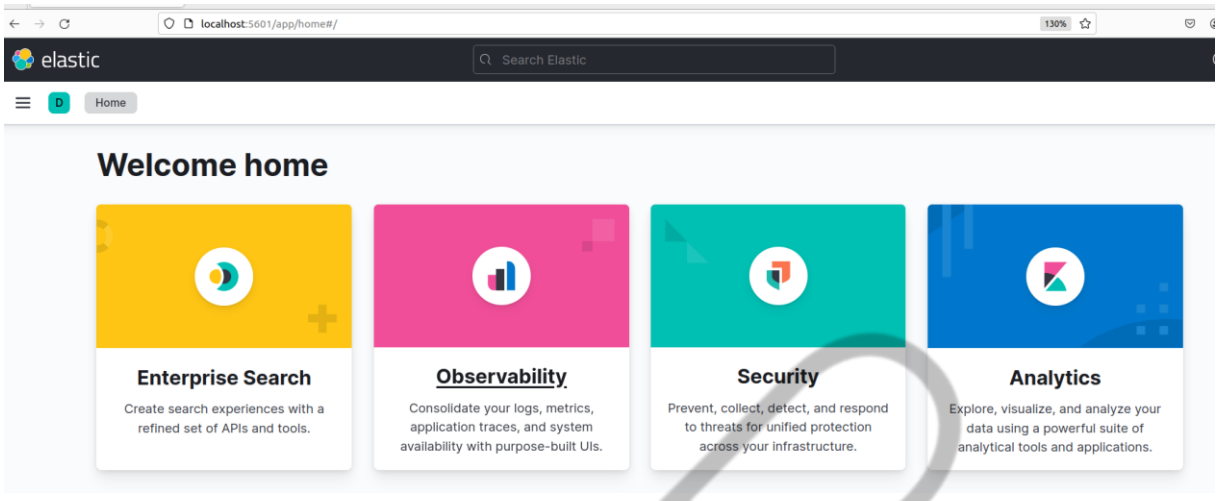


Figura 47 – http://localhost:5601 (2)
Fonte: elaborado pelo autor (2024)

## O QUE VOCÊ VIU NESTA AULA?

Nessa aula você entendeu o funcionamento de uma das ferramentas mais versáteis para ETL, tratamento de dados e arquivos: o Apache NIFI! Entendemos como usar a ferramenta para se conectar a diversas fontes de dados, realizamos uma demonstração com dados gerados artificialmente e colocamos isso em prática em nosso pipeline.

Além disso, você foi apresentado(a) ao ElasticSearch, peça central da stack ELK, e entendeu como manusear, inserir, consultar as informações, alterar e deletar um documento.

# REFERÊNCIAS

ELASTIC. **Data in**: documents and indices. 2024. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/documents-indices.html>. Acesso em: 28 mai. 2024.

ELASTIC. **What is ElasticSearch?** 2024. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>. Acesso em: 28 mai. 2024.

NIFI APACHE. **Apache Nifi Overview**. 2024. Disponível em: <https://nifi.apache.org/documentation/v2/>. Acesso em: 28 mai. 2024.

## PALAVRAS-CHAVE

**Palavras-chave:** ETL. ELK. Elastic. NiFi. Apache. Elasticsearch. Kibana.  Document.