

# Implementação e Análise de Sistemas Distribuídos: Experiência Prática com HDFS e PySpark

## Equipe

Este trabalho foi desenvolvido na disciplina de Sistemas Distribuídos por:

Autor	Matrícula	Função
Artur da Silva Oliveira	2122082008	Configuração de rede
Eduardo Costa Virmond	1922082002	Implementação HDFS
João Víctor Alves Menezes	2122082017	Configuração PySpark
Juliana Alves Pacheco	2122082026	Documentação
Natália Bastos Pereira	2212082020	Testes e validação
Rafael Dantas Boeira	2122082004	Coordenação e integração

## Resumo

Este documento apresenta uma análise detalhada da implementação e configuração de um sistema distribuído baseado no ecossistema Hadoop, com foco no Sistema de Arquivos Distribuído Hadoop (HDFS) e PySpark. A pesquisa documenta os procedimentos realizados em três laboratórios práticos, abordando desde a instalação e configuração do ambiente base até a execução de tarefas de processamento distribuído. O estudo demonstra a aplicabilidade prática desses sistemas e discute os conhecimentos adquiridos durante o processo de implementação.

## 1. Introdução

Os sistemas distribuídos têm se tornado fundamentais para o processamento de grandes volumes de dados. O ecossistema Hadoop, especialmente com seu sistema de arquivos distribuído (HDFS) e ferramentas como PySpark, representa uma solução robusta para esse cenário. Este documento registra a experiência prática de implementação desses sistemas em um ambiente controlado, demonstrando os procedimentos necessários para sua instalação, configuração e operação.

## 2. Materiais e Métodos

O estudo foi conduzido realizando experimentos em três etapas laboratoriais sequenciais:

### 2.1 Laboratório 1: Configuração do Ambiente Base

O primeiro laboratório concentrou-se na instalação e configuração do sistema operacional Ubuntu Server, que serviu como base para o sistema distribuído. Os procedimentos incluíram:

```
# Instale o sistema operacional Ubuntu Server
# Faça o update e upgrade
```

```
sudo apt update
sudo apt upgrade

# Verifique se a data e hora estão atualizadas
date

# Instale os aplicativos para sincronização de tempo
sudo apt install ntp ntpsec-ntpdate
sudo ntpq -p

# Altere o timezone
sudo dpkg-reconfigure tzdata

# Verifique a data e a hora novamente
date
```

```
Ubuntu 24.04.2 LTS ubuntuserver tty1
ubuntuserver login: ubuntu_user
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Apr  4 01:42:52 AM UTC 2025

 System load:          1.85
 Usage of /:           41.6% of 11.21GB
 Memory usage:         14%
 Swap usage:           0%
 Processes:            116
 Users logged in:     0
 IPv4 address for enp0s3: 10.0.2.15
 IPv6 address for enp0s3: fd00::a00:27ff:fedc:f864

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu_user@ubuntuserver:~$ sudo apt update
```

```
ubuntuserver login: ubuntu_user
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Apr  4 01:42:52 AM UTC 2025

 System load:          1.85
 Usage of /:           41.6% of 11.21GB
 Memory usage:         14%
 Swap usage:           0%
 Processes:            116
 Users logged in:     0
 IPv4 address for enp0s3: 10.0.2.15
 IPv6 address for enp0s3: fd00::a00:27ff:fedc:f864

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

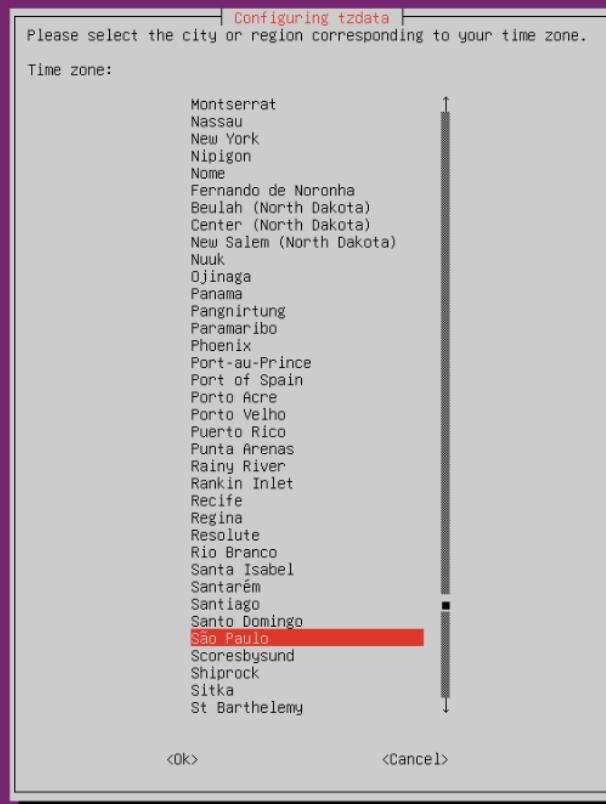
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu_user@ubuntuserver:~$ sudo apt update
[sudo] password for ubuntu_user:
[git:1 http://security.ubuntu.com/ubuntu noble-security InRelease
[git:2 http://archive.ubuntu.com/ubuntu noble InRelease
[git:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
[git:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu_user@ubuntuserver:~$ sudo apt upgrade
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host
ubuntu_user@ubuntuserver:~$ date
Fri Apr  4 01:46:19 AM UTC 2025
ubuntu_user@ubuntuserver:~$ sudo apt install ntp ntpsec-ntpdate
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ntpsec ntpsec-ntpdig python3-ntp
Suggested packages:
  certbot ntpsec-doc ntpsec-ntpviz
The following packages will be REMOVED:
  systemd-timesyncd
The following NEW packages will be installed:
  ntp ntpsec ntpsec-ntpdate ntpsec-ntpdig python3-ntp
0 upgraded, 5 newly installed, 1 to remove and 2 not upgraded.
Need to get 497 kB of archives.
After this operation, 1,287 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

package configuration



```
Current default time zone: 'America/Sao_Paulo'
Local time is now:      Thu Apr  3 22:49:26 -03 2025.
Universal Time is now: Fri Apr  4 01:49:26 UTC 2025.
```

```
ubuntu_user@ubuntuserver:~$ _
```

## 2.2 Laboratório 2: Implementação do HDFS

O segundo laboratório abordou a instalação e configuração do Hadoop Distributed File System, com os seguintes procedimentos:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install pdsh
echo "export PDSH_RCMD_TYPE=ssh" >> .bashrc
ssh-keygen -t rsa -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ssh localhost
sudo apt-get install openjdk-8-jdk

# Baixe do classroom ou da rede local o arquivo hadoop-3.4.0.tar.gz para a
# pasta home
# Alternativamente
# wget https://archive.apache.org/dist/hadoop/common/hadoop-3.4.0/hadoop-
# 3.4.0.tar.gz

tar xvf hadoop-3.4.0.tar.gz
mv hadoop-3.4.0 hadoop
sudo mv hadoop /usr/local/hadoop
sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
# Adicionar linha a seguir:
# export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

sudo nano /etc/environment
# Adicionar as linhas a seguir:
#
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"
# JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"

sudo poweroff
```

## Instalação do Java e Hadoop

```
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  genders libgenders0 libice6 libsm6 libxt6t64 ssh-askpass x11-common
Suggested packages:
  rdist
The following NEW packages will be installed:
  genders libgenders0 libice6 libsm6 libxt6t64 pdsh ssh-askpass x11-common
0 upgraded, 8 newly installed, 0 to remove and 2 not upgraded.
Need to get 451 kB of archives.
After this operation, 1,550 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 libgenders0 amd64 1.22-1build8 [30.8 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 genders amd64 1.22-1build8 [31.0 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 x11-common all 1:7.7+23ubuntu3 [21.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libice6 amd64 2:1.0.10-1build3 [41.4 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 libsm6 amd64 2:1.2.3-1build3 [15.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 libxt6t64 amd64 1:1.2.1-1.2build1 [171 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 ssh-askpass amd64 1:1.2.4.1-16build2 [24.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 pdsh amd64 2.34-3build2 [115 kB]
Fetched 451 kB in 2s (224 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libgenders0:amd64.
(Reading database ... 86740 files and directories currently installed.)
Preparing to unpack .../0-libgenders0_1.22-1build8_amd64.deb ...
Unpacking libgenders0:amd64 (1.22-1build8) ...
Selecting previously unselected package genders.
Preparing to unpack .../1-genders_1.22-1build8_amd64.deb ...
Unpacking genders (1.22-1build8) ...
Selecting previously unselected package x11-common.
Preparing to unpack .../2-x11-common_1%3a7.7+23ubuntu3_all.deb ...
Unpacking x11-common (1:7.7+23ubuntu3) ...
Selecting previously unselected package libice6:amd64.
Preparing to unpack .../3-libice6_2%3a1.0.10-1build3_amd64.deb ...
ubuntu_user@ubuntuserver:~$ echo " export PDASH_RCMD_TYPE=ssh" >> .bashrc
ubuntu_user@ubuntuserver:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu_user/.ssh/id_rsa):
Your identification has been saved in /home/ubuntu_user/.ssh/id_rsa
Your public key has been saved in /home/ubuntu_user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3xNM1bU+V22ILeLv/cMUeyuXn1ds8Adq4JI9A+RI3g4 ubuntu_user@ubuntuserver
The key's randomart image is:
----[RSA 3072]----+
| ...   000 |
| 0 +. . .0+.+|
```

```
E 0= 0.0.B+
  00 =0+ .=B
  S. +0. .=X
  . .0+0
  . 0.0+0
  ..0.
  +
-----[SHA256]-----
ubuntu_user@ubuntuserver:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ubuntu_user@ubuntuserver:~$ sudo systemctl start ssh.service
```

## Host hadoop-base

**HostName** localhost

**Port** 2230

**User** ubuntu\_user

**IdentityFile** ~/OneDrive/Documentos/aws-keys/id\_rsa

## Instalação do Java e Hadoop

```
ubuntu_user@ubuntuserver:~$ sudo apt-get install openjdk-8-jdk
[sudo] password for ubuntu_user:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend fontconfig fonts-dejavu-extra gsettings-desktop-schemas
  gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2-data libasound2t64 libasyncns0 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64
  libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcurl5t64 libdatrie1 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau1 libdrm-radeon1 libflac12t64 libgail-common
  libgail18t64 libgdm1 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx-wx libgraphite2-3 libgtk2.0-0t64
  libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev liblcms2-2 liblvm19 libmp3lame0 libmpg123-0t64 libogg0 libopus0 libpango-1.0-0 libpangocairo-1.0-0 libpiciaccs0 libpcsc-lite1 libpixman-1-0
  libpthread-stubs0-dev libpulse0 librsvg2-common libsm-dev libsndfile1 libthai-data libthai0 libvorbis0a libvorbisenc2 libwayland-client0 libwayland-server0 libx11-dev libx11-xcb libxau-dev
  libxaw libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-render0 libxcb-renderer0 libxcb-shape0 libxcb-sync1 libxcb-xfixes0 libxcb-dev libcomposite1 libcursor1 libxdamage1 libxdmp-dev
  libxfixes3 libxt2 libxlib xi libxinerama1 libxbfile1 libxmu0 libxrandr2 libxrender1 libxshmfence1 libxt-dev libxtst6 libxv1 libxxf86dg1 libxxf86vm1 mesa-libgallium mesa-vulkan-drivers openjdk-8-jdk-headless
  openjdk-8-jre openjdk-8-jre-headless session-migration ubuntu-mono x11-utils x1proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre alsa-utils libasound2-plugins cups-common gvfs libice-doc liblcms2-utils opus-tools pccsd pulseaudio librsvg2-bin libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-8-demo openjdk-8-source visualvm
  libns-mdns fonts-nanum fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
  luit
The following NEW packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend fontconfig fonts-dejavu-extra gsettings-desktop-schemas
  gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2-data libasound2t64 libasyncns0 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64
  libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcurl5t64 libdatrie1 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau1 libdrm-radeon1 libflac12t64 libgail-common
  libgail18t64 libgdm1 libgdk-pixbuf2.0-0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx-wx libgraphite2-3 libgtk2.0-0t64
  libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev liblcms2-2 liblvm19 libmp3lame0 libmpg123-0t64 libogg0 libopus0 libpango-1.0-0 libpangocairo-1.0-0 libpiciaccs0 libpcsc-lite1 libpixman-1-0
  libpthread-stubs0-dev libpulse0 librsvg2-common libsm-dev libsndfile1 libthai-data libthai0 libvorbis0a libvorbisenc2 libwayland-client0 libwayland-server0 libx11-dev libx11-xcb libxau-dev
  libxaw libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-render0 libxcb-renderer0 libxcb-shape0 libxcb-sync1 libxcb-xfixes0 libxcb-dev libcomposite1 libcursor1 libxdamage1 libxdmp-dev
  libxfixes3 libxt2 libxlib xi libxinerama1 libxbfile1 libxmu0 libxrandr2 libxrender1 libxshmfence1 libxt-dev libxtst6 libxv1 libxxf86dg1 libxxf86vm1 mesa-libgallium mesa-vulkan-drivers openjdk-8-jdk
  openjdk-8-jre openjdk-8-jre-headless session-migration ubuntu-mono x11-utils x1proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 126 newly installed, 0 to remove and 2 not upgraded.
Need to get 119 MB of archives.
After this operation, 502 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

```
ubuntu_user@ubuntuserver:~$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
--2025-04-03 23:31:50-- https://archive.apache.org/dist/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a884::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 965537117 (921M) [application/x-gzip]
Saving to: 'hadoop-3.4.0.tar.gz'

hadoop-3.4.0.tar.gz          3%[==>]   29.50M  6.77MB/s    eta 3m 50s
ubuntu_user@ubuntuserver:~$ tar xvf hadoop-3.4.0.tar.gz
mv hadoop-3.4.0 hadoop
sudo mv hadoop /usr/local/hadoop
```

```

GNU nano 7.2                                         /usr/local/hadoop/etc/hadoop/hadoop-env.sh *
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

## THIS FILE ACTS AS THE MASTER FILE FOR ALL HADOOP PROJECTS.
## SETTINGS HERE WILL BE READ BY ALL HADOOP COMMANDS. THEREFORE,
## ONE CAN USE THIS FILE TO SET YARN, HDFS, AND MAPREDUCE
## CONFIGURATION OPTIONS INSTEAD OF xxx-env.sh.

## Precedence rules:
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
## {YARN_XYZ|HDFS_XYZ} > HADOOP_XYZ > hard-coded defaults
## Generic settings for HADOOP
## Technically, the only required environment variable is JAVA_HOME,
## All others are optional. However, the defaults are probably not
## preferred. Many sites configure these options outside of Hadoop,
## such as in /etc/profile.d.

## The java implementation to use. By default, this environment
## variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

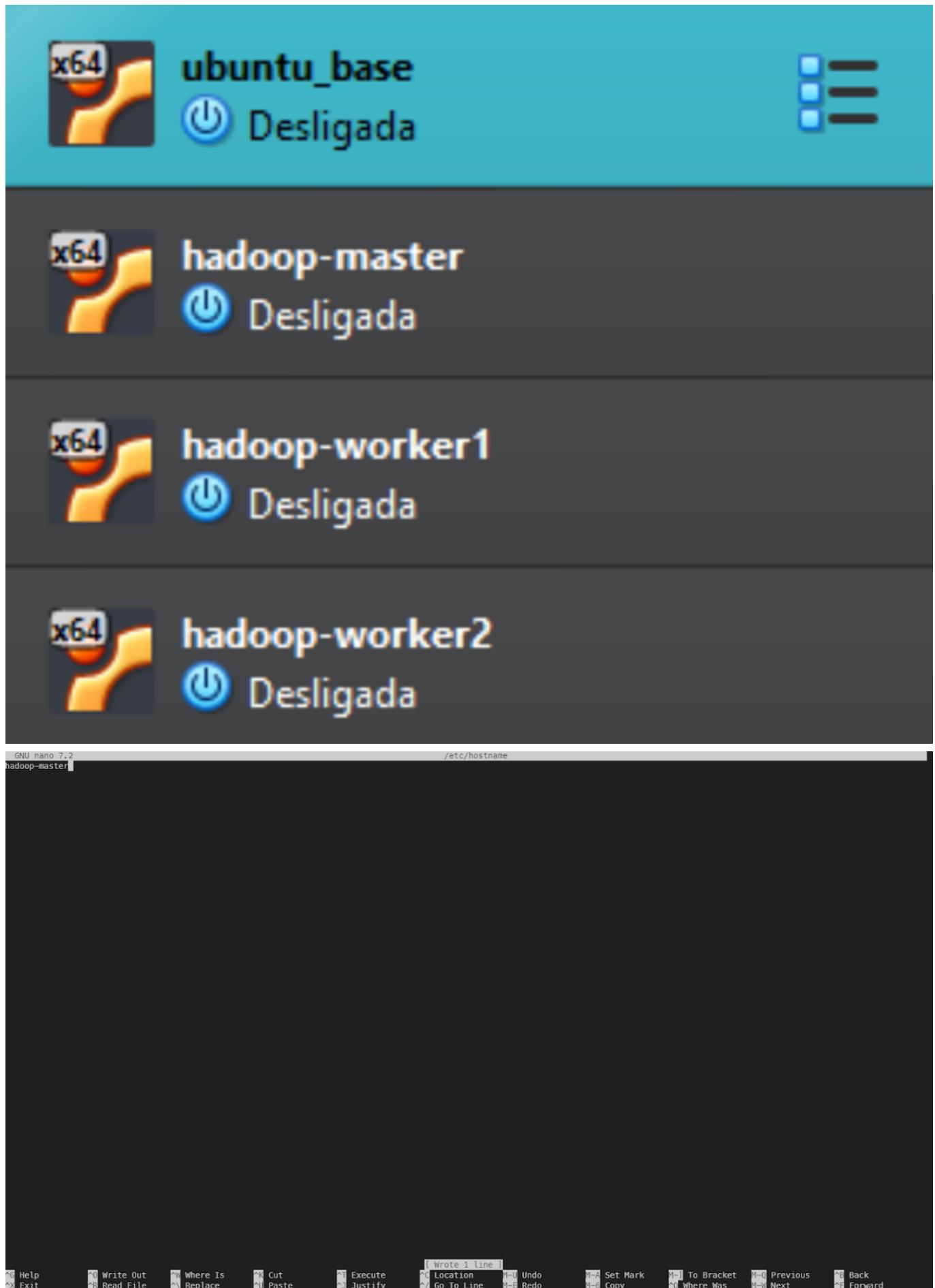
## The language environment in which Hadoop runs. Use the English
## environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8

GNU nano 7.2                                         /etc/environment *
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"

ubuntu_user@ubuntuserver:~$ sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
ubuntu_user@ubuntuserver:~$ sudo nano /etc/environment
ubuntu_user@ubuntuserver:~$ sudo poweroff
ubuntu_user@ubuntuserver:~$ 

```

Após criar uma placa de rede HostOnly na máquina virtual, crie três clones da máquina base: 1 master e 2 workers.



The screenshot shows two separate terminal windows side-by-side. Both windows are running the nano text editor on the file /etc/hostname.

The top window is titled "hadoop-worker1" and shows the command "sudo nano /etc/hostname". The bottom window is titled "hadoop-worker2" and also shows the command "sudo nano /etc/hostname".

The nano interface includes a menu bar with options like Help, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, To Bracket, Copy, Where Was, Previous, Next, Back, and Forward. The status bar at the bottom of each window displays the path "/etc/hostname \*".

```
# Editar o arquivo hostname em cada máquina: hadoop-master, hadoop-worker1,  
hadoop-worker2  
sudo nano /etc/hostname  
  
# Faça o reboot dos servidores!  
# Verifique o nome da placa, que no caso pode ser enp0s8  
ip addr  
  
# Editar os arquivos de rede das máquinas  
# Colocar os IPs 172.31.110.10, 172.31.110.11 e 172.31.110.12 nas máquinas
```

```
master e workers 1 e 2, respectivamente
cd /etc/netplan
sudo nano XXXXX.yaml

# Exemplo
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [172.31.110.10/24]

sudo netplan apply

# Editar o arquivo /etc/hosts incluindo os nomes e IPs
# hadoop-master, hadoop-worker1 e hadoop-worker2
# Não colocar 127.0.1.1 com o hostname
sudo nano /etc/hosts
# Adicionar:
172.31.110.10 hadoop-master
172.31.110.11 hadoop-worker1
172.31.110.12 hadoop-worker2
```

## Configuração de rede e arquivo hosts

```
GNU nano 7.2                                         50-cloud-init.yaml *
```

```
network:
version: 2
ethernets:
enp0s3:
  dhcp4: true
enp0s8:
  dhcp4: false
addresses: [172.31.110.10/24]
```

```
GNU nano 7.2                                         50-cloud-init.yaml *
```

```
network:
version: 2
ethernets:
enp0s3:
  dhcp4: true
enp0s8:
  dhcp4: false
addresses: [172.31.110.11/24]
```

```
GNU nano 7.2                                         50-cloud-init.yaml */  
network:  
version: 2  
ethernets:  
  enp0s3:  
    dhcp4: true  
  enp0s8:  
    dhcp4: false  
    addresses: [172.31.110.12/24]
```

```
ubuntu_user@hadoop-master:~$ sudo netplan apply  
[sudo] password for ubuntu_user:  
ubuntu_user@hadoop-master:~$
```

```
GNU nano 7.2  
127.0.0.1 localhost  
172.31.110.10 hadoop-master  
172.31.110.11 hadoop-worker1  
172.31.110.12 hadoop-worker2
```

```
# The following lines are desirable for IPv6 capable hosts  
::1      ip6-localhost ip6-loopback  
fe00::0  ip6-localnet  
ff00::0  ip6-mcastprefix  
ff02::1  ip6-allnodes  
ff02::2  ip6-allrouters
```

```
<configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>

</configuration>
```

```
<configuration>

<property>
    <name>mapreduce.jobtracker.address</name>
    <value>hadoop-master:9001</value>
</property>

<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>

<property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
```

```
<configuration>

<property>
<name>yarn.resourcemanager.hostname</name>
<value>hadoop-master</value>
</property>

<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

<property>
<name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>0.0.0.0:8088</value>
</property>

</configuration>
```

hadoop-worker1

hadoop-worker2

```
# Copiar os arquivos hadoop_files que estão no Classroom ou na rede local
# para /usr/local/hadoop/etc/hadoop no master
sudo apt-get install unzip
unzip hadoop_files
cp ~/hadoop_files/*.* /usr/local/hadoop/etc/hadoop

# Verifique o nome do hadoop-master e workers nos arquivos
# core-site.xml hdfs-site.xml mapred-site.xml workers yarn-site.xml
# Deve-se ajustar os arquivos para seu cluster (principalmente os nomes
hosts)
```

```
# O mesmo arquivo deve ir para os workers, lembrando que tem que certificar
os nomes hosts:
scp /usr/local/hadoop/etc/hadoop/* hadoop-
worker1:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-
worker2:/usr/local/hadoop/etc/hadoop/
```

## Transferência de arquivos de configuração

```
ubuntu_user@hadoop-master:~$ cp ~/hadoop_files/* /usr/local/hadoop/etc/hadoop/
```

```
ubuntu_user@hadoop-master:~$ scp /usr/local/hadoop/etc/hadoop/* hadoop-worker1:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-worker2:/usr/local/hadoop/etc/hadoop/
The authenticity of host 'hadoop-worker1 (172.31.110.11)' can't be established.
ED25519 key fingerprint is SHA256:PS3bSr0w+1zNvxQ709RjBpirxACEt1BTgBD1E0dRGtc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'hadoop-worker1' (ED25519) to the list of known hosts.
ubuntu_user@hadoop-worker1's password:
capacity-scheduler.xml
configuration.xsl
container-executor.cfg
core-site.xml
hadoop-env.cmd
hadoop-env.sh
hadoop-metrics2.properties
hadoop-policy.xml
hadoop-user-functions.sh.example
hdfs-rbf-site.xml
hdfs-site.xml
httpfs-env.sh
httpfs-log4j.properties
httpfs-site.xml
kms-acls.xml
kms-env.sh
kms-log4j.properties
kms-site.xml
log4j.properties
mapred-env.cmd
mapred-env.sh
mapred-queues.xml.template
mapred-site.xml
scp: local "/usr/local/hadoop/etc/hadoop/shellprofile.d" is not a regular file
scp: failed to upload file /usr/local/hadoop/etc/hadoop/shellprofile.d to /usr/local/hadoop/etc/hadoop/
ssl-client.xml.example
ssl-server.xml.example
user_ec_policies.xml.template
workers
yarn-env.cmd
yarn-env.sh
yarnservice-log4j.properties
yarn-site.xml
The authenticity of host 'hadoop-worker2 (172.31.110.12)' can't be established.
ED25519 key fingerprint is SHA256:PS3bSr0w+1zNvxQ709RjBpirxACEt1BTgBD1E0dRGtc.
This host key is known by the following other names/addresses:
 ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'hadoop-worker2' (ED25519) to the list of known hosts.
ubuntu_user@hadoop-worker2's password:
capacity-scheduler.xml
configuration.xsl
container-executor.cfg
core-site.xml
hadoop-env.cmd
hadoop-env.sh
hadoop-metrics2.properties
hadoop-policy.xml
hadoop-user-functions.sh.example
hdfs-rbf-site.xml
hdfs-site.xml
httpfs-env.sh
httpfs-log4j.properties
httpfs-site.xml
kms-acls.xml
kms-env.sh
kms-log4j.properties
kms-site.xml
log4j.properties
mapred-env.cmd
mapred-env.sh
mapred-queues.xml.template
mapred-site.xml
scp: local "/usr/local/hadoop/etc/hadoop/shellprofile.d" is not a regular file
scp: failed to upload file /usr/local/hadoop/etc/hadoop/shellprofile.d to /usr/local/hadoop/etc/hadoop/
ssl-client.xml.example
ssl-server.xml.example
user_ec_policies.xml.template
workers
yarn-env.cmd
yarn-env.sh
yarnservice-log4j.properties
yarn-site.xml
ubuntu_user@hadoop-master:~$
```

```

.bashrc
108  # enable programmable completion features (you don't need to enable
109  # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110  # sources /etc/bash.bashrc).
111  if ! shopt -oq posix; then
112      if [ -f /usr/share/bash-completion/bash_completion ]; then
113          . /usr/share/bash-completion/bash_completion
114      elif [ -f /etc/bash_completion ]; then
115          . /etc/bash_completion
116      fi
117  fi
118  export PDSH_RCMD_TYPE=ssh
119
120  export HADOOP_HOME="/usr/local/hadoop"
121  export HADOOP_COMMON_HOME=$HADOOP_HOME
122  export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
123  export HADOOP_HDFS_HOME=$HADOOP_HOME
124  export HADOOP_MAPRED_HOME=$HADOOP_HOME
125  export HADOOP_YARN_HOME=$HADOOP_HOME

```

- **ubuntu\_user@hadoop-master:~\$ source ~/.bashrc**
- **ubuntu\_user@hadoop-master:~\$ echo \$HADOOP\_HOME**  
**/usr/local/hadoop**
- **ubuntu\_user@hadoop-master:~\$**

```

2025-04-04 00:12:39,054 INFO namenode.FSDirectory: GLOBAL serial map: bits=29 maxEntries=536870911
2025-04-04 00:12:39,054 INFO namenode.FSDirectory: USER serial map: bits=24 maxEntries=16777215
2025-04-04 00:12:39,054 INFO namenode.FSDirectory: GROUP serial map: bits=24 maxEntries=16777215
2025-04-04 00:12:39,054 INFO namenode.FSDirectory: XATTR serial map: bits=24 maxEntries=16777215
2025-04-04 00:12:39,081 INFO util.GSet: Computing capacity for map INodeMap
2025-04-04 00:12:39,081 INFO util.GSet: VM type      = 64-bit
2025-04-04 00:12:39,081 INFO util.GSet: 1.0% max memory 437.5 MB = 4.4 MB
2025-04-04 00:12:39,081 INFO util.GSet: capacity     = 2^19 = 524288 entries
2025-04-04 00:12:39,082 INFO namenode.FSDirectory: ACLs enabled? true
2025-04-04 00:12:39,082 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? true
2025-04-04 00:12:39,082 INFO namenode.FSDirectory: XAttrs enabled? true
2025-04-04 00:12:39,083 INFO namenode.NameNode: Caching file names occurring more than 10 times
2025-04-04 00:12:39,098 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAccessTimeOnlyChange: false, snapshotDiffAllowSnapRootDescendant: true, maxSnapshotFSLimit: 65536, maxSnapshotLimit: 65536
2025-04-04 00:12:39,090 INFO snapshot.SnapshotManager: dfs.namenode.snapshot.deletion.ordered = false
2025-04-04 00:12:39,092 INFO snapshot.SnapshotManager: SkipList is disabled
2025-04-04 00:12:39,097 INFO util.GSet: Computing capacity for map cachedBlocks
2025-04-04 00:12:39,097 INFO util.GSet: VM type      = 64-bit
2025-04-04 00:12:39,097 INFO util.GSet: 0.25% max memory 437.5 MB = 1.1 MB
2025-04-04 00:12:39,097 INFO util.GSet: capacity     = 2^17 = 131072 entries
2025-04-04 00:12:39,133 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2025-04-04 00:12:39,133 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2025-04-04 00:12:39,133 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2025-04-04 00:12:39,145 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2025-04-04 00:12:39,146 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2025-04-04 00:12:39,155 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2025-04-04 00:12:39,155 INFO util.GSet: VM type      = 64-bit
2025-04-04 00:12:39,156 INFO util.GSet: 0.02999999329447746% max memory 437.5 MB = 134.4 KB
2025-04-04 00:12:39,156 INFO util.GSet: capacity     = 2^14 = 16384 entries
2025-04-04 00:12:39,228 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2057482078-172.31.110.10-1743736359209
2025-04-04 00:12:39,273 INFO common.Storage: Storage directory /usr/local/hadoop/data/nameNode has been successfully formatted.
2025-04-04 00:12:39,493 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/data/nameNode/current/fsimage.ckpt_00000000000000000000 using no compression
2025-04-04 00:12:39,662 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/data/nameNode/current/fsimage.ckpt_00000000000000000000 of size 403 bytes saved in 0 seconds
2025-04-04 00:12:39,695 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2025-04-04 00:12:39,702 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2025-04-04 00:12:39,706 INFO namenode.FSNamesystem: Stopping services started for active state
2025-04-04 00:12:39,706 INFO namenode.FSNamesystem: Stopping services started for standby state
2025-04-04 00:12:39,711 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2025-04-04 00:12:39,711 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN MSG: Shutting down NameNode at hadoop-master/172.31.110.10

```

```
# Criando um ambiente virtual para o Hadoop
sudo nano ~/.bashrc
# Acrescente as seguintes variáveis em .bashrc do master
export HADOOP_HOME="/usr/local/hadoop"
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME

# Recarregar as variáveis de ambiente
source ~/.bashrc

# Executar o comando para a formação do HDFS:
hdfs namenode -format

# Iniciar o HDFS
start-dfs.sh

# O comando jps serve para verificar os Datanodes e Namenodes ativos
jps
```

Saída do comando jps com nós ativos

```
● ubuntu_user@hadoop-master:~$ start-dfs.sh
Starting namenodes on [hadoop-master]
Starting datanodes
Starting secondary namenodes [hadoop-master]
● ubuntu_user@hadoop-master:~$ jps
4470 Jps
4331 SecondaryNameNode
4060 NameNode
○ ubuntu_user@hadoop-master:~$ █
ubuntu_user@hadoop-worker2:~$ jps
2257 Jps
● ubuntu_user@hadoop-worker1:~$ jps
2485 Jps
```

```
# Crie outro redirecionamento de porta para o Master na porta 9870 para
9870
```

```
# Acesse: http://localhost:9870/
# Clique em Utilities >> Browse the file system
```

### Interface web do HDFS em localhost:9870

Nome	Protocolo	Endereço IP do Hospedeiro	Porta do Hospedeiro	IP do Convidado	Porta do Convidado
Rule 1	TCP		2230		22
Rule 2	TCP		9870		9870

OK      Cancelar

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name

Showing 0 to 0 of 0 entries

● **ubuntu\_user@hadoop-master:~\$ stop-dfs.sh**  
**Stopping namenodes on [hadoop-master]**  
**Stopping datanodes**  
**Stopping secondary namenodes [hadoop-master]**

### 2.3 Laboratório 3: Operações HDFS e PySpark

O terceiro laboratório focou na utilização prática do HDFS e implementação do PySpark para processamento de dados:

```
# Procedimentos para a limpeza e formação do HDFS, caso necessário
# Deletar todos os arquivos das pastas dataNode e nameNode (é necessário
que essas pastas existam)
```

```
sudo rm -R /usr/local/hadoop/data/*  
  
# Deletar os arquivos temporários  
sudo rm -R /tmp/*  
  
# Executar a formação do nameNode  
hdfs namenode -format  
  
# Iniciar o HDFS  
start-dfs.sh  
  
# Comandos exemplo para a operação do HDFS  
hdfs dfs -mkdir /teste  
  
# Clique em Utilities >> Browse the file system  
vi ~/arquivo_de_teste.txt  
hdfs dfs -put ~/arquivo_de_teste.txt /teste  
hdfs dfs -ls /teste  
mkdir ~/tmp  
hdfs dfs -get /teste ~/tmp/  
ls ~/tmp  
hdfs dfs -rm /teste/arquivo_de_teste.txt  
hdfs dfs -ls /teste  
hdfs dfs -rm -r /teste  
hdfs dfs -ls /
```

## Operações de manipulação de arquivos no HDFS

- **ubuntu\_user@hadoop-master:~\$ sudo rm -R /usr/local/hadoop/data/\***  
## deletar os arquivos temporários  
  
sudo rm -R /tmp/\*  
  
## executar a formação do nameNode  
  
hdfs namenode -format
- **ubuntu\_user@hadoop-master:~\$ start-dfs.sh**  
Starting namenodes on [hadoop-master]  
Starting datanodes  
Starting secondary namenodes [hadoop-master]
- **ubuntu\_user@hadoop-master:~\$ hdfs dfs -mkdir /teste**

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<input type="checkbox"/>	drwxr-xr-x	ubuntu_user	supergroup	0 B	Apr 04 00:34	0	0 B	teste

Showing 1 to 1 of 1 entries

arquivo\_de\_teste.txt ubuntu\_user X

arquivo\_de\_teste.txt

1 OLA MUNDO, COMO ESTAO?

```

● ubuntu_user@hadoop-master:~$ hdfs dfs -mkdir /teste
● ubuntu_user@hadoop-master:~$ hdfs dfs -put ~/arquivo_de_teste.txt /teste
● ubuntu_user@hadoop-master:~$ hdfs dfs -ls /teste
Found 1 items
-rw-r--r-- 2 ubuntu_user supergroup 22 2025-04-04 00:52 /teste/arquivo_de_teste.txt
● ubuntu_user@hadoop-master:~$ mkdir ~/tmp
● ubuntu_user@hadoop-master:~$ hdfs dfs -get /teste ~/tmp/
● ubuntu_user@hadoop-master:~$ ls ~/tmp
teste
● ubuntu_user@hadoop-master:~$ hdfs dfs -rm /teste/arquivo_de_teste.txt
hdfs dfs -ls /teste
hdfs dfs -rm -r /teste
hdfs dfs -ls /
Deleted /teste/arquivo_de_teste.txt
Deleted /teste
Found 2 items
drwxr-xr-x - ubuntu_user supergroup 0 2025-03-23 13:20 /tmp
drwxr-xr-x - ubuntu_user supergroup 0 2025-03-23 13:20 /user

```

```

# Testando o YARN
start-yarn.sh # ou /usr/local/hadoop/sbin/start-all.sh
yarn application -list
cat $HADOOP_HOME/logs/yarn-*-resourcemanager-*.log | tail -n 50

# Crie outro redirecionamento de porta para o Master no IP 172.31.110.10 na
porta 8088
# Acesse: http://localhost:8088/
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
examples-3.4.0.jar pi 30 100

```

Interface web do YARN em localhost:8088



## All Applications

Cluster Metrics													
About Nodes Node Labels Applications													
NEW NEW_SAVING SUBMITTED ACCEPTED RUNNING FINISHED FAILED KILLED													
Scheduler													
Tools													
Showing 0 to 0 of 0 entries													
No data available in table													

## Cálculo de n

ubuntu_user@hadoop-master:~\$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar pi 30 100
Number of Maps = 30
Samples per Map = 100
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Wrote input for Map #10
Wrote input for Map #11
Wrote input for Map #12
Wrote input for Map #13
Wrote input for Map #14
Wrote input for Map #15
Wrote input for Map #16
Wrote input for Map #17
Wrote input for Map #18
Wrote input for Map #19
Wrote input for Map #20
Wrote input for Map #21
Wrote input for Map #22
Wrote input for Map #23
Wrote input for Map #24
Wrote input for Map #25
Wrote input for Map #26
Wrote input for Map #27
Wrote input for Map #28
Wrote input for Map #29
Starting Job
2025-04-04 00:57:01,709 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at hadoop-master/172.31.110.10:8032
2025-04-04 00:57:02,964 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/ubuntu_user/.staging/job_17437385
2025-04-04 00:57:03,368 INFO input.FileInputFormat: Total input files to process : 30
2025-04-04 00:57:03,556 INFO mapreduce.JobSubmitter: number of splits:30
2025-04-04 00:57:04,436 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1743738957991_0001
2025-04-04 00:57:04,437 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-04 00:57:05,353 INFO conf.Configuration: resource-types.xml not found
2025-04-04 00:57:05,353 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-04 00:57:06,069 INFO impl.YarnClientImpl: Submitted application application_1743738957991_0001
2025-04-04 00:57:06,124 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1743738957991_0001/
2025-04-04 00:57:06,124 INFO mapreduce.Job: Running job: job_1743738957991_0001
2025-04-04 00:57:18,342 INFO mapreduce.Job: Job job_1743738957991_0001 running in uber mode : false
2025-04-04 00:57:18,343 INFO mapreduce.Job: map 0% reduce 0%

Show 20+ entries													
ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores
application_1743738957991_0001	ubuntu_user	QuasiMonteCarlo	MAPREDUCE		root/default	0	Fri Apr 4 00:57:05 2025	Fri Apr 4 00:57:05 2025	N/A	RUNNING	UNDEFINED	15	15

```
Map-Reduce Framework
    Map input records=30
    Map output records=60
    Map output bytes=540
    Map output materialized bytes=840
    Input split bytes=4700
    Combine input records=0
    Combine output records=0
    Reduce input groups=2
    Reduce shuffle bytes=840
    Reduce input records=60
    Reduce output records=0
    Spilled Records=120
    Shuffled Maps =30
    Failed Shuffles=0
    Merged Map outputs=30
    GC time elapsed (ms)=8452
    CPU time spent (ms)=48870
    Physical memory (bytes) snapshot=8004456448
    Virtual memory (bytes) snapshot=78907682816
    Total committed heap usage (bytes)=7669284864
    Peak Map Physical memory (bytes)=313053184
    Peak Map Virtual memory (bytes)=2552188928
    Peak Reduce Physical memory (bytes)=224133120
    Peak Reduce Virtual memory (bytes)=2553503744
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=3540
File Output Format Counters
    Bytes Written=97
Job Finished in 156.251 seconds
Estimated value of Pi is 3.14133333333333333333
```

```
mkdir ~/gutenberg
wget https://www.gutenberg.org/cache/epub/20417/pg20417.txt >
~/gutenberg/pg20417.txt
cd ~/gutenberg/
hdfs dfs -mkdir -p /teste
hdfs dfs -put ~/gutenberg/pg20417.txt /teste
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
examples-3.4.0.jar wordcount /teste/pg20417.txt
/user/ubuntu_user/gutenberg-output/
hdfs dfs -cat /user/ubuntu_user/gutenberg-output/part-r-00000
```

Resultado da operação wordcount

```

ubuntu_user@hadoop-master:~$ mkdir ~gutenberg
ubuntu_user@hadoop-master:~$ wget https://www.gutenberg.org/cache/epub/20417/pg20417.txt > ~/gutenberg/pg20417.txt
--2025-04-04 01:01:21-- https://www.gutenberg.org/cache/epub/20417/pg20417.txt
Resolving www.gutenberg.org (www.gutenberg.org)... 152.19.134.47, 2618:28:3090:3000:0:bad:cafe:47
Connecting to www.gutenberg.org (www.gutenberg.org)|152.19.134.47|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 674656 (659K) [text/plain]
Saving to: pg20417.txt.1

pg20417.txt.1          100%[=====] 658.84K   999KB/s    in 0.7s

2025-04-04 01:01:22 (999 KB/s) - 'pg20417.txt.1' saved [674656/674656]

ubuntu_user@hadoop-master:~$ cd ~/gutenberg/
ubuntu_user@hadoop-master:~/gutenberg$ hdfs dfs -mkdir -p /teste
ubuntu_user@hadoop-master:~/gutenberg$ hdfs dfs -put ~gutenberg/pg20417.txt /teste

ubuntu_user@hadoop-master:~/gutenberg$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar wordcount /teste/pg20417.txt /user/ubuntu_user/gutenberg-output/
2025-04-04 01:03:23,431 INFO client.DefaultHttpClient:allowOverProxyProvider: Connecting to ResourceManager at hadoop-master/172.31.10.10:8082
2025-04-04 01:03:23,813 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/ubuntu_user/.staging/job_1743738957991_0002
2025-04-04 01:03:24,532 INFO input.FileInputFormat: Total input files to process : 1
2025-04-04 01:03:26,181 INFO mapreduce.JobSubmitter: number of splits:1
2025-04-04 01:03:27,134 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1743738957991_0002
2025-04-04 01:03:27,410 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-04 01:03:27,410 INFO conf.Configuration: resource-types.xml not found
2025-04-04 01:03:27,410 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-04 01:03:27,409 INFO YarnClientImpl: Submitted application application_1743738957991_0002
2025-04-04 01:03:27,522 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1743738957991_0002/
2025-04-04 01:03:27,523 INFO mapreduce.Job: Running job: job_1743738957991_0002
2025-04-04 01:03:33,630 INFO mapreduce.Job: map 0% reduce 0%
2025-04-04 01:03:33,641 INFO mapreduce.Job: map 100% reduce 0%
2025-04-04 01:03:37,720 INFO mapreduce.Job: map 100% reduce 100%
2025-04-04 01:03:42,773 INFO mapreduce.Job: map 100% reduce 100%
2025-04-04 01:03:42,806 INFO mapreduce.Job: Job job_1743738957991_0002 completed successfully
2025-04-04 01:03:42,918 INFO mapreduce.Job: Counters: 54
File System Counters
  FILE: Number of bytes read=6
  FILE: Number of bytes written=618319
  FILE: Number of read operations=8
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=100
  HDFS: Number of bytes written=0
  HDFS: Number of read operations=8
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=1869
  Total time spent by all reduces in occupied slots (ms)=1792
  Total time spent by all map tasks (ms)=1869
  Total time spent by all reduce tasks (ms)=1792
  Total vcore-milliseconds taken by all map tasks=1869
  Total vcore-milliseconds taken by all reduce tasks=1792
  Total megabyte-milliseconds taken by all map tasks=1913856
  Total megabyte-milliseconds taken by all reduce tasks=1835000
Map-Reduce Framework
  Map input records=0
  Map output records=0
  Map output bytes=0
  Map output materialized bytes=6
  Input split bytes=100
  Combine input records=0
  Combine output records=0
  Reduce input groups=0
  Reduce shuffle bytes=0
  Reduce input records=0
  Reduce output records=0
  Spilled Records=0
  Shuffled Maps=1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=83
  CPU time spent (ms)=748
  Physical memory (bytes) snapshot=520986624
  Virtual memory (bytes) snapshot=5897283584
  Total committed heap usage (bytes)=394264576
  Peak Map Physical memory (bytes)=316571648
  Peak Map Virtual memory (bytes)=2506495488
  Peak Reduce Physical memory (bytes)=284414976
  Peak Reduce Virtual memory (bytes)=2550788096
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=0

```

```

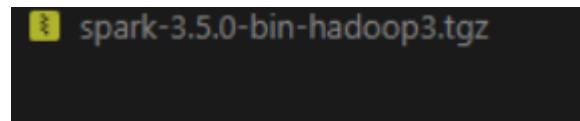
# Configuração e uso do PySpark
# Baixar spark do ClassRoom e colocar no diretório home
tar -xvzf spark-3.5.0-bin-hadoop3.tgz
mv spark-3.5.0-bin-hadoop3 spark
sudo mv spark /usr/local/spark
sudo nano ~/.bashrc

```

```
# Incluir essas linhas:  
export SPARK_HOME="/usr/local/spark/"  
export PATH="$PATH:$SPARK_HOME/bin"  
  
# Carregar variáveis de ambiente  
source ~/.bashrc  
pyspark
```

## Ambiente PySpark iniciado

```
# Criar um contexto de sessão do spark (cria um "programa")  
sc = SparkContext.getOrCreate()  
  
# Variável recebe o caminho que aponta para um arquivo de texto  
file_path = "/teste/pg20417.txt"  
  
# Leitura do arquivo de texto pelo programa spark  
words = sc.textFile(f"{file_path}").flatMap(lambda line: line.split(" "))  
  
# Contagem de palavras utilizando a sintaxe facilitada do pyspark  
wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)  
  
# Salvando arquivo com resultado da execução  
wordCounts.saveAsTextFile(f"/'.join(file_path.split('/')  
[:-1])}/word_count")  
wordCounts.count()
```



```
export HADOOP_HOME="/usr/local/hadoop"  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_YARN_HOME=$HADOOP_HOME  
  
export SPARK_HOME="/usr/local/spark/"  
export PATH="$PATH:$SPARK_HOME/bin"
```

## Welcome to

version 3.5.0

```
Using Python version 3.12.3 (main, Feb 4 2025 14:48:35)
Spark context Web UI available at http://hadoop-master:4040
Spark context available as 'sc' (master = local[*], app id = local-1743739578992).
SparkSession available as 'spark'.
>>> # criar um contexto de sessão do spark (cria um "programa")
>>> sc = SparkContext.getOrCreate()
>>> # variável recebe o caminho que aponta para uma arquivo de texto
>>> file_path = "/teste/pg20417.txt"
>>> # leitura do arquivo de texto pelo programa spark
>>> words = sc.textFile(f"{file_path}).flatMap(lambda line: line.split(" "))
sintaxe facilitada do pyspark
wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)
# salvando arquivo com resultado da execução
wordCounts.saveAsTextFile(f"{'.'.join(file_path.split('/')[-1])}/word_count")

wordCounts.count()
```

```
>>> wordCounts.count()
```

18020

### 3. Resultados e Discussão

A implementação do ambiente distribuído permitiu responder diversas questões técnicas e aprofundar o conhecimento em várias áreas:

### 3.1 Configuração de Sistemas

Foi possível compreender os procedimentos para a instalação e configuração adequada do Ubuntu Server, incluindo a sincronização de tempo e ajuste do fuso horário, elementos críticos para o funcionamento coordenado de sistemas distribuídos.

## 3.2 Implementação do Hadoop

O laboratório proporcionou conhecimento prático sobre a instalação e configuração do ecossistema Hadoop, incluindo:

- Instalação de dependências como Java e ferramentas auxiliares
  - Configuração de variáveis de ambiente
  - Ajuste de arquivos de configuração específicos do Hadoop
  - Estabelecimento de comunicação segura entre os nós através de SSH

### 3.3 Configuração de Rede e Cluster

Os experimentos demonstraram a importância da configuração adequada da rede para sistemas distribuídos:

- Definição de IPs estáticos para cada nó
- Configuração de nomes de host e DNS local via arquivo `/etc/hosts`
- Estratégias para comunicação eficiente entre nós

### 3.4 Operações no HDFS

Foram executadas e compreendidas operações fundamentais no sistema de arquivos distribuído:

- Criação de diretórios e manipulação de arquivos
- Transferência de dados entre o sistema de arquivos local e o HDFS
- Monitoramento e verificação do estado do sistema

### 3.5 MapReduce e Processamento Distribuído

Os laboratórios permitiram a implementação de exemplos de processamento distribuído:

- Execução de jobs MapReduce para cálculo de  $\pi$
- Implementação de contagem de palavras (wordcount) em documentos
- Monitoramento de aplicações através do YARN

### 3.6 PySpark e Análise de Dados

A utilização do PySpark demonstrou o potencial para análise de dados em grande escala:

- Configuração e uso da API Python para o Spark
- Implementação de algoritmos para contagem de palavras em documentos
- Armazenamento e recuperação de resultados processados

### 3.7 Gestão do Cluster

Os procedimentos de inicialização e encerramento adequados do cluster foram compreendidos:

- Inicialização e parada coordenada dos serviços
- Utilização de ferramentas como `sshpass` para automação de tarefas
- Monitoramento do estado do cluster

## 4. Curiosidades e Dúvidas Respondidas

Durante os laboratórios, várias curiosidades e dúvidas foram respondidas, incluindo:

### 4.1 Configuração do Ubuntu Server

Durante a preparação do ambiente, aprendemos a importância crítica da sincronização de tempo em sistemas distribuídos. Sem uma sincronização precisa, os nós do cluster podem enfrentar problemas de coordenação, resultando em falhas de processamento ou inconsistência de dados.

### 4.2 Arquitetura do HDFS

Compreendemos como o HDFS divide e replica dados através de múltiplos nós, oferecendo tanto redundância quanto paralelismo. A divisão entre NameNode (para metadados) e DataNodes (para armazenamento) demonstra uma separação clara de responsabilidades.

#### 4.3 Configuração e Instalação do Hadoop

Aprendemos os passos necessários para instalar e configurar o Hadoop em um cluster distribuído, incluindo a definição de variáveis de ambiente e o ajuste dos arquivos de configuração para refletir corretamente a topologia do cluster.

#### 4.4 Operações Básicas no HDFS

Descobrimos como criar diretórios, transferir arquivos e listar dados no HDFS usando comandos como `hdfs dfs -put` e `hdfs dfs -get`, permitindo o gerenciamento eficiente dos dados no sistema distribuído.

#### 4.5 Execução de Jobs MapReduce

Experimentamos a execução de exemplos de MapReduce, como o cálculo de π e contagem de palavras, e aprendemos a monitorar aplicações usando o YARN, obtendo insights sobre como trabalhos complexos são divididos e executados em paralelo.

#### 4.6 Configuração e Uso do PySpark

Exploramos como instalar e configurar o PySpark em um ambiente Hadoop, e como executar scripts para processamento de dados, aproveitando a sintaxe intuitiva do Python para operações distribuídas complexas.

#### 4.7 Gerenciamento de Cluster

Aprendemos técnicas para iniciar e parar os serviços do Hadoop de forma adequada, garantindo a integridade do sistema e dos dados durante o ciclo de vida do cluster.

#### 4.8 Comunicação e Sincronização entre Nós

Compreendemos como configurar SSH para comunicação sem senha entre os nós do cluster e como usar ferramentas como `pdsh` para execução paralela de comandos, facilitando a administração do sistema.

#### 4.9 Monitoramento e Depuração

Descobrimos como verificar processos em execução usando `jps` e como interpretar logs e saídas de jobs MapReduce, habilidades essenciais para manutenção e otimização do sistema.

### 5. Conclusões

Os laboratórios proporcionaram uma compreensão prática abrangente dos sistemas distribuídos baseados no ecossistema Hadoop. A experiência não apenas esclareceu dúvidas técnicas, mas também forneceu uma visão mais profunda de como esses sistemas funcionam na prática, desde a configuração inicial até a execução de tarefas complexas de processamento de dados.

A implementação passo a passo permitiu o desenvolvimento de habilidades técnicas essenciais para a administração e uso de sistemas distribuídos, demonstrando a viabilidade e o poder dessas tecnologias.

para processamento de grandes volumes de dados.

O conhecimento adquirido sobre HDFS, MapReduce e PySpark constitui uma base sólida para aplicações futuras em ambientes de Big Data, onde o processamento distribuído é fundamental para lidar com a escala crescente dos conjuntos de dados modernos.