

## CIENCIA DE DATOS:

# APRENDE LOS FUNDAMENTOS DE MANERA PRÁCTICA



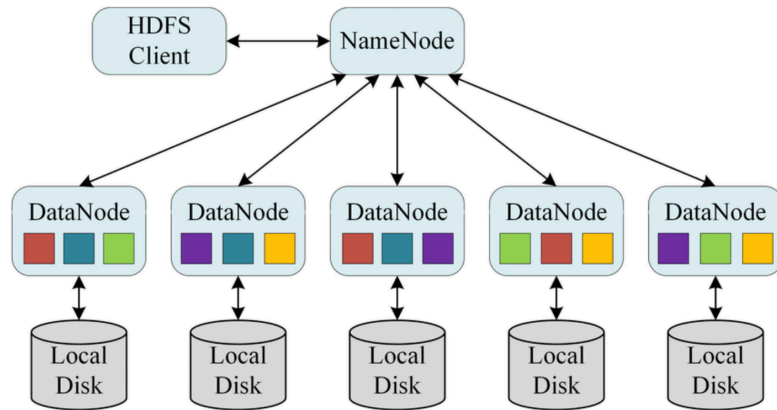
## LABORATORIO 04 LINUX & HDFS

**Juan Chipoco**  
mindquasar@gmail.com

# ÍNDICE

OBJETIVO.....	4
COMANDOS AVANZADOS LINUX.....	5
COMANDOS HDFS.....	10

## OBJETIVO



El objetivo de este laboratorio es profundizar en el sistema de archivos distribuido de Hadoop y sus comandos.

Proporciona conceptos básicos y avanzados de Linux.

## COMANDOS AVANZADOS LINUX

### Introducción de Linux a los usuarios

Aca te mostraremos cómo identificar la cuenta de usuario de un sistema con comandos como quién, quién soy, etc.

Si más de una persona usa un solo sistema, entonces todos pueden tener su propia cuenta de usuario. Aquí, será útil conocer los detalles de la cuenta de usuario.

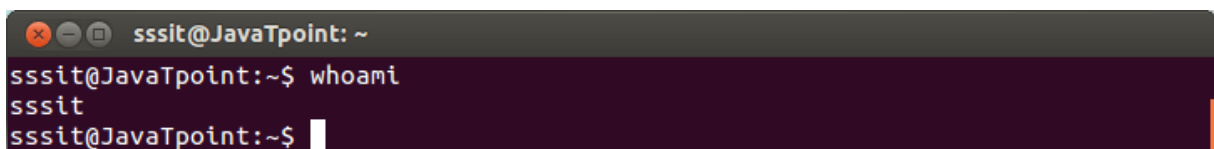
También explica cómo crear una segunda cuenta de usuario y ejecutar un programa con la ayuda de los comandos su y sudo.

quién soy

Le informa sobre el nombre de usuario del sistema.

#### Sintaxis:

1. whoami



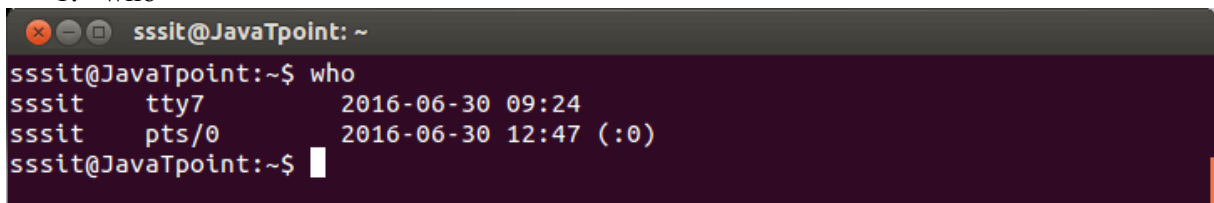
```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ whoami  
sssit  
sssit@JavaTpoint:~$
```

Mire la instantánea anterior, '**sssit**' es el nombre de usuario de nuestro sistema.  
quién

El comando who brinda información sobre los usuarios que iniciaron sesión en el sistema.

#### Sintaxis:

1. who



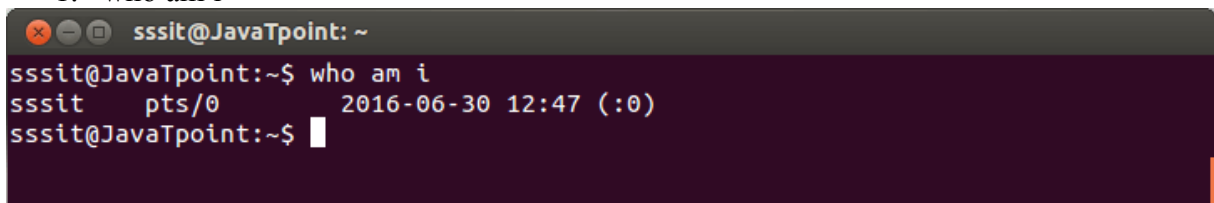
```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ who  
sssit    tty7          2016-06-30 09:24  
sssit    pts/0        2016-06-30 12:47 (:0)  
sssit@JavaTpoint:~$
```

quién soy

Este comando muestra la información sobre el usuario actual únicamente.

#### Sintaxis:

1. who am i



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ who am i  
sssit    pts/0        2016-06-30 12:47 (:0)  
sssit@JavaTpoint:~$
```

Mire la instantánea anterior, en nuestro sistema, el usuario que ha iniciado sesión actualmente es **sssit** .

w

Este comando informa sobre los usuarios que han iniciado sesión y qué están haciendo.

**Sintaxis:**

1. w

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ w  
16:02:31 up 6:37, 2 users, load average: 0.05, 0.12, 0.20  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT  
sssit     tty7                09:24    6:37m   9:56   0.23s gnome-session -  
sssit     pts/0      :0            12:47    0.00s   0.17s   0.00s w  
sssit@JavaTpoint:~$
```

identificación

Este comando informa sobre su identificación de usuario, identificación de grupo principal y una lista de grupos que le pertenecen.

**Sintaxis:**

1. id

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ id  
uid=1000(sssit) gid=1000(sssit) groups=1000(sssit),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(lpadmin),124(sambashare)  
sssit@JavaTpoint:~$
```

## locate

Puedes usar este comando para **localizar** un archivo, al igual que el comando de búsqueda en Windows. Además, el uso del argumento **-i** junto con este comando hará que no distinga entre mayúsculas y minúsculas, por lo que puedes buscar un archivo incluso si no recuerdas su nombre exacto.

Para buscar un archivo que contenga dos o más palabras, usa un asterisco (\*). Por ejemplo, el comando **locate -i escuela\*nota** buscará cualquier archivo que contenga la palabra «escuela» y «nota», ya sea en mayúsculas o minúsculas.

## find

Similar al comando **locate**, usando **find** también buscas archivos y directorios. La diferencia es que usas el comando **find** para ubicar archivos dentro de un directorio dado.

Como ejemplo, el comando **find /home/ -name notas.txt** buscará un archivo llamado **notas.txt** dentro del directorio de inicio y sus subdirectorios.

Otras variaciones al usar **find** son:

- Para buscar archivos en el directorio actual, **find . -name notas.txt**
- Para buscar directorios, **/ -type d -name notes.txt**

## grep

Otro comando básico de Linux que sin duda es útil para el uso diario es **grep**. Te permite buscar a través de todo el texto en un archivo dado.

Para ilustrar, **grep azul notepad.txt** buscará la palabra azul en el archivo del bloc de notas. Las líneas que contienen la palabra buscada se mostrarán.

### **sudo**

Abreviatura de «**SuperUser Do**» (SuperUsuario hace), este comando te permite realizar tareas que requieren permisos administrativos o raíz. Sin embargo, no es aconsejable usar este comando para el uso diario, ya que podría ser fácil que ocurra un error si haces algo mal.

### **df**

Usa el comando **df** para obtener un informe sobre el uso del espacio en disco del sistema, que se muestra en porcentaje y KB. Si deseas ver el informe en megabytes, escribe **df -m**.

### **du**

Si deseas verificar cuánto espacio ocupa un archivo o un directorio, el comando **du** (Uso del disco, en inglés) es la respuesta. Sin embargo, el resumen de uso del disco mostrará números de bloque de disco en lugar del formato de tamaño habitual. Si deseas verlo en bytes, kilobytes y megabytes, agrega el argumento **-h** a la línea de comando.

### **head**

El comando **head** se usa para ver las primeras líneas de cualquier archivo de texto. De manera predeterminada, mostrará las primeras diez líneas, pero puedes cambiar este número a tu gusto. Por ejemplo, si solo deseas mostrar las primeras cinco líneas, escribe **head -n 5 nombrearchivo.ext**.

### **tail**

Este tiene una función similar al comando **head**, pero en lugar de mostrar las primeras líneas, el comando **tail** mostrará las últimas diez líneas de un archivo de texto. Por ejemplo, **tail -n nombrearchivo.ext**.

### **diff**

Para abreviar diferencia, el comando **diff** compara el contenido de dos archivos línea por línea. Después de analizar los archivos, genera las líneas que no coinciden. Los programadores a menudo usan este comando cuando necesitan hacer modificaciones al programa en lugar de reescribir todo el código fuente.

La forma más simple de usar este comando es **diff archivo1.ext archivo2.ext**

### **tar**

El comando **tar** es el comando más utilizado para guardar múltiples archivos en un **tarball**, un formato de archivo de Linux común que es similar al formato zip, con compresión opcional. Este comando es bastante complejo con una larga lista de funciones, como agregar nuevos archivos a un archivo existente, enumerar el contenido de un archivo, extraer el contenido de un archivo y muchos más.

### **chmod**

**chmod** es otro comando de Linux, utilizado para cambiar los permisos de lectura, escritura y ejecución de archivos y directorios. Como este comando es bastante complicado, puedes leer el [tutorial completo](#) (en inglés) para ejecutarlo correctamente.

### **chown**

En Linux, todos los archivos son propiedad de un usuario específico. El comando **chown** te permite cambiar o transferir la propiedad de un archivo al nombre de usuario especificado. Por

ejemplo, **chown usuariolinux2 archivo.ext** hará que **usuariolinux2** sea el propietario del **archivo.ext**.

### **kill**

Si tienes un programa que no responde, puedes cerrarlo manualmente utilizando el [comando kill](#). Enviará una cierta señal al programa que se está ejecutando mal y le indica a la aplicación que finalice.

Hay un total de sesenta y cuatro señales que puedes usar, pero las personas generalmente solo usan dos señales:

- **SIGTERM (15)**: solicita que un programa deje de ejecutarse y te da algo de tiempo para guardar todo tu progreso. Si no especificas la señal al ingresar el comando kill, se utilizará esta señal.
- **SIGKILL (9)**: obliga a los programas a detenerse inmediatamente. El progreso no guardado se perderá.

Además de conocer las señales, también debes conocer el número de identificación del proceso (PID) del programa que deseas detener (**kill**). Si no conoces el PID, simplemente ejecute el comando **ps ux**.

Después de saber qué señal deseas usar y el PID del programa, ingresa la siguiente sintaxis:  
**kill [opción de señal] PID.**

### **ping**

Usa el comando **ping** para verificar tu estado de conectividad a un servidor. Por ejemplo, simplemente ingresando **ping google.com**, el comando verificará si puedes conectarte a Google y también medirá el tiempo de respuesta.

### **wget**

La línea de comandos de Linux es muy útil: incluso puedes descargar archivos de Internet con la ayuda del comando **wget**. Para hacerlo, simplemente escribe **wget** seguido del enlace de descarga.

### **uname**

El comando **uname**, abreviatura de Nombre de Unix, imprimirá información detallada sobre tu sistema Linux, como el nombre de la máquina, el sistema operativo, el núcleo, etc.

### **top**

Como un terminal equivalente al Administrador de tareas en Windows, el comando **top** mostrará una lista de los procesos en ejecución y la cantidad de CPU que utiliza cada proceso. Es muy útil monitorear el uso de los recursos del sistema, especialmente para saber qué proceso debe terminarse porque consume demasiados recursos.

### **history**

Cuando hayas estado utilizando Linux durante un cierto período de tiempo, notarás rápidamente que puedes ejecutar cientos de comandos todos los días. Como tal, ejecutar el comando **history** es particularmente útil si deseas revisar los comandos que ingresaste anteriormente.

### **zip, unzip**

Usa el comando **zip** para comprimir tus archivos en un archivo zip y use el comando **unzip** para extraer los archivos comprimidos de un archivo zip.

### **hostname**

Si deseas conocer el nombre de tu host/red, simplemente escribe **hostname**. Agregar un **-I** al final mostrará la dirección IP de tu red.



## COMANDOS HDFS

hdfs

hdfs dfs //Lista todos los comandos hdfs

echo Hello World HDFS >> test.txt

ls -alt

cat test.txt

clear

hdfs dfs -ls /

hdfs dfs -ls /user

hdfs dfs -mkdir /prueba

hdfs dfs -mkdir /user/fuentes

hdfs dfs -mkdir /user/jcv

sudo su hdfs

hdfs dfs -ls /

hdfs dfs -mkdir /user/test

hdfs dfs -ls /user

vi /etc/group

vi /etc/passwd

vi /etc/shadow

vi /etc/resolve.conf

hdfs dfs -rm -R /user/jcv

hdfs dfs -ls /user

hdfs dfs -mkdir /user/jcv

hdfs dfs -cp file1.txt /user/jcv/. //Mal

hdfs dfs -put file1.txt /user/jcv //Ok

hdfs dfs -ls /user/jcv

Generar varios archivos locales filex.txt y copiarlos hacia el HDFS

hdfs dfs -put file\* /user/jcv //Ok

hdfs dfs -mkdir /user/jcv/hive

cienciadatos\_uni@cluster-125d-m:~\$ sudo hive

Hive Session ID = b5e844f3-51a5-4d71-9301-41912f964319

hive> show databases;

OK

default

Time taken: 0.043 seconds, Fetched: 1 row(s)

hive>

## Creación de una tabla de base de datos mediante el lenguaje de consulta de Hive (HQL)

Hive es una solución de almacenamiento de datos construida sobre Hadoop. En Hive, los datos se administran en el sistema de archivos distribuido de Hadoop (HDFS). En este esquema, al leer no se requiere verificación de restricciones como se requiere en RDBMS. Está especialmente diseñado para trabajar con un conjunto de datos muy grande. Hive utiliza un lenguaje de consulta conocido como Hive Query Language (HQL).

Permite hacer consultas y analizar grandes cantidades de datos almacenados en HDFS, en la escala de petabytes. Tiene un lenguaje de consulta llamado **HiveQL** o HQL que internamente transforma las consultas SQL en trabajos MapReduce que ejecutan en Hadoop. El lenguaje de consulta HQL es un dialecto de SQL, que no sigue el estándar ANSI SQL, sin embargo es muy similar.

El proyecto comenzó en el 2008 y fue desarrollado por Facebook para hacer que Hadoop se comportara de una manera más parecida a un **data warehouse** tradicional.

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async:
true
Hive Session ID = 2b5ed81c-d20e-410a-af20-266423d50531
hive>
```

```
hive> create database firstdb;
OK
Time taken: 1.492 seconds
hive> show databases;
OK
default
firstdb
Time taken: 0.037 seconds, Fetched: 2 row(s)
hive>
```

```
hive> create table main_table
> (
> id int,
> name string,
> city string
> );
OK
Time taken: 0.515 seconds
hive>
```

```
hive> show tables;
OK
main_table
Time taken: 0.07 seconds, Fetched: 1 row(s)
hive>
```

```
Time taken: 0.07 seconds, Fetched: 1 row(s)
hive> insert into table main_table
> values(101, 'Peter', 'Lima');
Query ID = root_20221221171929_005a5e3a-2cc5-4bf4-9dfa-6de81725e251
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1671640483927_0002)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    1          1          0          0          0          0
Reducer 2 ..... container  SUCCEEDED    1          1          0          0          0          0
-----
VERTICES: 02/02 [=====] 100% ELAPSED TIME: 7.68 s
-----
Loading data to table firstdb.main_table
OK
Time taken: 22.419 seconds
hive>
```

```
Time taken: 11.45 seconds
hive> select * from main_table;
Query ID = root_20221221172215_dcada550-7c1e-46ef-b92d-9b3adde6515f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671640483927_0002)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    1          1          0          0          0          0
-----
VERTICES: 01/01 [=====] 100% ELAPSED TIME: 11.45 s
-----
OK
101      Peter   Lima
Time taken: 13.003 seconds, Fetched: 1 row(s)
hive>
```

hive> show databases;

OK

default

firstdb

Time taken: 0.038 seconds, Fetched: 2 row(s)

hive> describe firstdb;

FAILED: SemanticException [Error 10001]: Table not found firstdb

hive> show databases;

OK

default

firstdb

Time taken: 0.066 seconds, Fetched: 2 row(s)

hive> use firstdb;

OK

Time taken: 0.047 seconds

hive> show tables;

OK

main\_table

Time taken: 0.04 seconds, Fetched: 1 row(s)

hive> describe main\_table;

OK

id int

name string

city string

Time taken: 0.086 seconds, Fetched: 3 row(s)

hive>

```
hive> drop table main_table;
OK
Time taken: 0.462 seconds
hive> show tables;
OK
Time taken: 0.063 seconds
hive>
```

```
hdfs fsck /prueba/VirtualBox-7.0.4-154605-OSX.dmg -files -blocks -locations
```

### **Listar Archivos en HDFS**

hdfs dfs -ls /	Lista todos los ficheros y directorios para el path /
hdfs dfs -ls -h /	Lista los ficheros con su tamaño en formato legible
hdfs dfs -ls -R /	Lista todos los ficheros y directorios recursivamente (con subdirectorios)
hdfs dfs -ls /file*	Lista todos los ficheros que cumplen el patrón (ficheros que comienzan con 'file')

### **Leer y Escribir Archivos**

hdfs dfs -text /app.log	Imprime el fichero en modo texto por la terminal
hdfs dfs -cat /app.log	Muestra el contenido del fichero en la salida estándar
hdfs dfs -appendToFile /home/file1 /file2	Añade el contenido del fichero local 'file1' al fichero en hdfs 'file2'

### **Cargar y Descargar Archivos**

hdfs dfs -put /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs
hdfs dfs -put -f /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs y lo sobrescribe en el caso de que ya exista
hdfs dfs -put -l /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs. Fuerza replicación 1 y permite al DataNode persistir los datos de forma perezosa.
hdfs dfs -put -p /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs. Mantiene los tiempos de acceso, de modificación y propietario original
hdfs dfs -get /file1 /home/	Copia el fichero 'file1' de hdfs al sistema de ficheros local
hdfs dfs -moveFromLocal /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs y luego lo borra del sist. ficheros local

### **Gestión de Archivos**

hdfs dfs -cp /hadoop/file1 /hadoop1	Copia el fichero al directorio destino en hdfs
hdfs dfs -cp -p /hadoop/file1 /hadoop1	Copia el fichero al directorio destino en hdfs conservando tiempos de acceso y de modificación, propietario y modo
hdfs dfs -rm /hadoop/file1	Elimina el fichero 'file1' de hdfs y lo envía a la papelera
hdfs dfs -rm -r /hadoop	
hdfs dfs -rm -R /hadoop	
hdfs dfs -rmr /hadoop	Elimina el directorio y su contenido en hdfs
hdfs dfs -rm -skipTrash /file1	Elimina el fichero sin dejarlo en la papelera
hdfs dfs -mkdir /hadoop2	Crea un directorio en hdfs
hdfs dfs -touchz /hadoop3	Crea un fichero en hdfs con tamaño 0

### **Gestión de Permisos**

hdfs dfs -checksum /hadoop/file1  
hdfs dfs -chmod 775 /hadoop/file1  
hdfs dfs -chmod -R 755 /hadoop  
hdfs dfs -chown hadoop:hadoop /file1  
hdfs dfs -chown -R hadoop:hadoop /file1  
hdfs dfs -chgrp hadoop /file1

Muestra la información checksum del fichero  
Cambia los permisos del fichero en hdfs  
Cambia los permisos de los ficheros recursivamente  
Cambia el propietario y el grupo del fichero  
Cambia el propietario y el grupo recursivamente  
Cambia el grupo del fichero

### ***Comandos de Administración HDFS***

hdfs dfs -df /hadoop  
hdfs dfs -df -h /hadoop  
en formato legible  
hadoop version  
hdfs fsck /  
hdfs dfsadmin -safemode leave  
hdfs namenode -format

Muestra la capacidad y el espacio libre y usado del sistema de ficheros  
Muestra la capacidad y el espacio libre y usado del sistema de ficheros  
Muestra la versión de hadoop  
Comprueba el estado de salud del sistema de ficheros  
Deshabilita el modo seguro del NameNode  
Formatea el NameNode