

Informe - Sistema de Ficheros Hadoop HDFS

Laboratorio: Creación de cluster

Autores: Gabriel Jaramillo Cuberos, Mariana Osorio Vásquez, Roberth Santiago Méndez, Juan Esteban Vera

Asignatura: Introducción a los Sistemas Distribuidos

Pontificia Universidad Javeriana - 2025

Resumen

Este informe describe la instalación, configuración y validación de un clúster Hadoop 3.4.2, tanto en modo “Single Node” como en modo “Multi-Node”, siguiendo la guía oficial de Apache Hadoop. El objetivo del laboratorio fue comprender el funcionamiento interno del sistema de ficheros HDFS y su integración con YARN para el procesamiento distribuido de datos.

Se configuraron los archivos esenciales (core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml) y se estableció la comunicación SSH entre nodos. Posteriormente, se formateó el NameNode, se iniciaron los servicios HDFS y YARN, y se validó el clúster mediante comandos de verificación y la ejecución de un trabajo MapReduce de prueba.

El sistema demostró funcionamiento estable, replicación de datos y correcta comunicación entre nodos, evidenciando el rol coordinado de NameNode, DataNodes, ResourceManager y NodeManagers.

Como resultado, el clúster distribuido logró procesar datos en paralelo, cumpliendo los objetivos del laboratorio y permitiendo comprender los principios de la tolerancia a fallos, la replicación de datos y la gestión de recursos en sistemas distribuidos.

Introducción

Este informe documenta el proceso completo de instalación, configuración y validación de un clúster Hadoop 3.4.2 configurado en la forma multi-nodo distribuido en cuatro sistemas de cómputo, desplegado sobre un entorno Linux. El propósito del laboratorio es comprender el funcionamiento interno del sistema de ficheros distribuido de Hadoop (HDFS) y su integración con el framework YARN (Yet Another Resource Negotiator) para la gestión de recursos y ejecución de tareas MapReduce.

Durante la práctica se siguió la guía oficial de Apache Hadoop. El objetivo general es lograr que todos los nodos se comuniquen correctamente, que los servicios HDFS y YARN operaran de forma coordinada, y que se comprenda la arquitectura y funcionamiento del sistema distribuido.

Definiciones

A lo largo del informe se presentan los siguientes términos relevantes al tema del sistema de ficheros Hadoop. Estas definiciones son importantes para comprender los aspectos técnicos del laboratorio.

Hadoop:

Framework de código abierto desarrollado por la Apache Software Foundation. Permite almacenar y procesar grandes volúmenes de datos en sistemas distribuidos mediante clústeres de computadoras que trabajan en paralelo.

HDFS (Hadoop Distributed File System):

Sistema de archivos distribuido de Hadoop. Divide los archivos en bloques que se almacenan en varios nodos del clúster, garantizando redundancia y tolerancia a fallos a través de la replicación de datos.

YARN (Yet Another Resource Negotiator):

Subsistema de Hadoop encargado de gestionar los recursos y planificar las tareas de ejecución. Determina en qué nodo se ejecutará cada proceso y controla el uso de CPU y memoria en todo el clúster.

MapReduce:

Modelo de programación de Hadoop que permite procesar datos de forma paralela en dos etapas:

- Map: transforma y filtra los datos.
- Reduce: combina los resultados parciales y genera la salida final.

NameNode:

Nodo maestro de HDFS que administra los metadatos del sistema de archivos (ubicación de bloques, permisos, estructura de carpetas). No almacena datos directamente, sino que controla a los DataNodes.

DataNode:

Nodo esclavo del clúster encargado de almacenar los bloques reales de los archivos. Se comunica constantemente con el NameNode para reportar su estado y disponibilidad.

ResourceManager:

Componente principal de YARN. Administra los recursos globales del clúster, asigna tareas a los nodos y coordina la ejecución de trabajos MapReduce.

NodeManager:

Proceso que se ejecuta en cada nodo trabajador. Informa al ResourceManager sobre la disponibilidad de recursos y ejecuta las tareas asignadas dentro de contenedores locales.

Arquitectura Master–Slave:

Estructura jerárquica donde un nodo maestro coordina las operaciones del sistema y varios nodos esclavos ejecutan las tareas o almacenan los datos.

Single Node Cluster:

Configuración de Hadoop en la que todos los servicios (NameNode, DataNode, ResourceManager y NodeManager) se ejecutan en un solo equipo. Se utiliza con fines de desarrollo o pruebas.

Multi-Node Cluster:

Configuración distribuida donde cada nodo cumple una función específica (por ejemplo, uno como maestro y varios como esclavos). Es la versión real de un clúster Hadoop de producción.

SSH (Secure Shell) / SSHD:

Protocolo de conexión remota que permite acceder y ejecutar comandos en otros equipos de forma segura. SSHD es el servicio que escucha las conexiones entrantes. En Hadoop, se usa para automatizar la ejecución de servicios sin pedir contraseñas (Passphraseless SSH).

JAVA_HOME:

Variable de entorno que indica la ubicación del JDK (Java Development Kit) instalado. Hadoop la utiliza para ejecutar todos sus procesos Java.

core-site.xml:

Archivo de configuración de Hadoop donde se define el sistema de archivos por defecto (fs.defaultFS) y la dirección del NameNode principal.

hdfs-site.xml:

Archivo donde se especifican los directorios locales que usará cada nodo para almacenar datos y el número de réplicas que tendrá cada bloque (dfs.replication).

mapred-site.xml:

Archivo que define los parámetros del framework MapReduce y su modo de ejecución (mapreduce.framework.name).

workers:

Archivo de texto que contiene la lista de los nodos esclavos (DataNodes/NodeManagers) que forman parte del clúster. Hadoop lo usa para iniciar o detener servicios en ellos automáticamente.

Daemon:

Proceso del sistema que se ejecuta en segundo plano. En Hadoop, cada componente (NameNode, DataNode, ResourceManager, NodeManager) se ejecuta como un daemon independiente.

pdsh:

Herramienta utilizada por Hadoop para ejecutar comandos en paralelo sobre varios nodos. Si la configuración SSH no es correcta, puede generar errores como "Permission denied".

Replication Factor (Factor de replicación):

Número de copias de cada bloque de datos que HDFS mantiene en el clúster. A mayor replicación, mayor tolerancia a fallos, pero también mayor consumo de espacio.

Tolerancia a fallos:

Capacidad del sistema para seguir funcionando aunque uno o más nodos fallen. HDFS logra esto replicando los datos en diferentes nodos y redistribuyendo tareas automáticamente.

Configuración Distribuida:

Acción de copiar los mismos archivos de configuración (core-site.xml, hdfs-site.xml, etc.) a todos los nodos para que trabajen bajo los mismos parámetros.

Heartbeats:

Mensajes periódicos que los DataNodes envían al NameNode para indicar que están activos. Si el NameNode deja de recibirlos, considera al nodo como inactivo y replica sus bloques en otros nodos.

Cluster Capacity (Capacidad configurada):

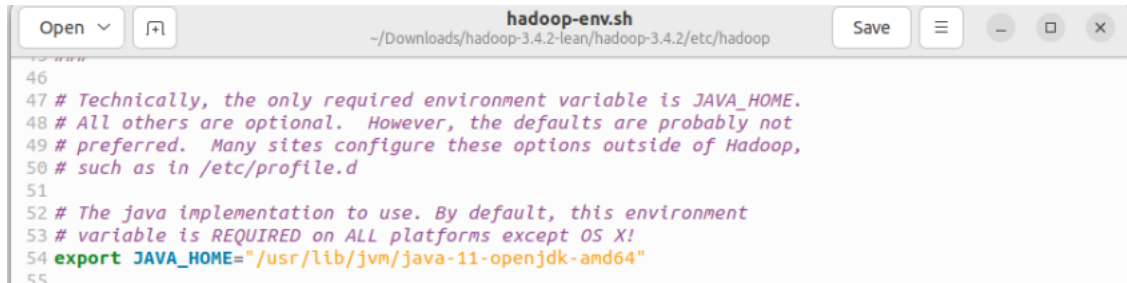
Suma del espacio total disponible en todos los DataNodes del clúster. Se puede visualizar desde la interfaz web del NameNode (<http://master:9870/>).

Preparación - Single node cluster

La configuración del cluster en un solo nodo cuenta con una serie de requisitos para su funcionamiento. Antes de iniciar la configuración del cluster, se realizaron los pasos para cumplir con los requisitos mínimos de software y comunicación entre los nodos:

1. Verificación de Requisitos de Software: se confirmó la instalación de Java(JDK) y SSH/SSHD en el nodo base:

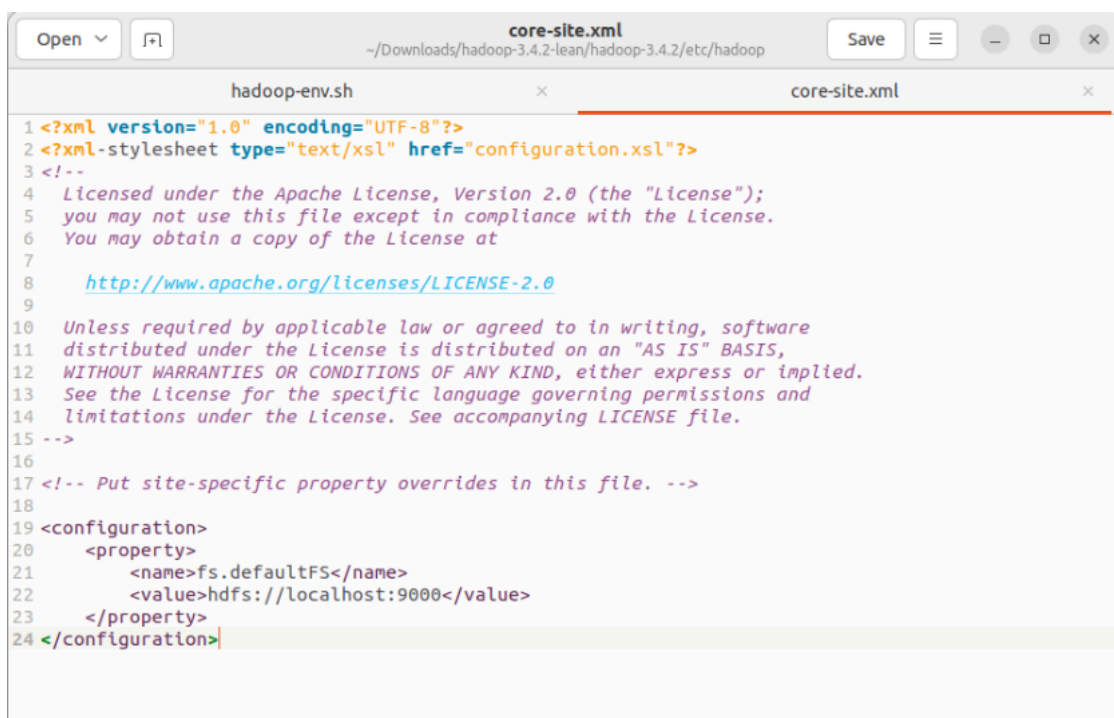
- a. Se identificó la ruta del JDK y se estableció la variable de entorno JAVA_HOME en el archivo etc/hadoop/hadoop-env.sh.



```
46
47 # Technically, the only required environment variable is JAVA_HOME.
48 # All others are optional. However, the defaults are probably not
49 # preferred. Many sites configure these options outside of Hadoop,
50 # such as in /etc/profile.d
51
52 # The java implementation to use. By default, this environment
53 # variable is REQUIRED on ALL platforms except OS X!
54 export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
55
```

Figura 1. Establecimiento de ruta JAVA_HOME

- b. Configuración de HDFS Se modificaron los archivos de configuración para apuntar el sistema de archivos por defecto a localhost:9000 y establecer la replicación en 1 (propio del modo pseudo-distribuido)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8 http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://localhost:9000</value>
23   </property>
24 </configuration>
```

Figura 2. Configuración de HDFS

2. Configuración de SSH sin Frase de Contraseña (Passphraseless SSH): se configuró el acceso SSH sin contraseña en el nodo para que los scripts de Hadoop puedan iniciar y detener los daemons localmente de forma automatizada.

- b. Ejecución y verificación de daemons: se iniciaron los servicios principales de HDFS. Se confirmó que la solución del problema de pdsh previa permitió el inicio sin errores de "Permission denied".

```
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [NGEN363]
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$
```

Figura 5. Ejecución y verificación de daemons

- c. Se confirmó que todos los daemons de HDFS están activos y corriendo.

```
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$ jps
2573415 Jps
2572322 SecondaryNameNode
2570728 NameNode
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$
```

Figura 6. Verificar daemons

- d. Se validó la funcionalidad de HDFS creando el directorio raíz del usuario.

```
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$ bin/hdfs dfs -mkdir -p /user/estudiante
estudiante@NGEN363:~/Downloads/hadoop-3.4.2-lean/hadoop-3.4.2$
```

Figura 7. Revisión de errores en HDFS.

4. Prueba funcional (MapReduce):

- Se ejecutó un trabajo de MapReduce de ejemplo para asegurar que el sistema es capaz de procesar datos distribuidos.
- Se copian archivos XML de configuración del sistema local al HDFS, dentro de un directorio de entrada (/input).
- Se ejecutó el ejemplo grep para buscar un patrón (dfs[a-z.]*) en los archivos.
- Se mostró la salida del trabajo MapReduce desde HDFS.
- Se accedió a la interfaz web del NameNode para verificar el estado del cluster y del sistema de archivos, como se muestra en la siguiente imagen:

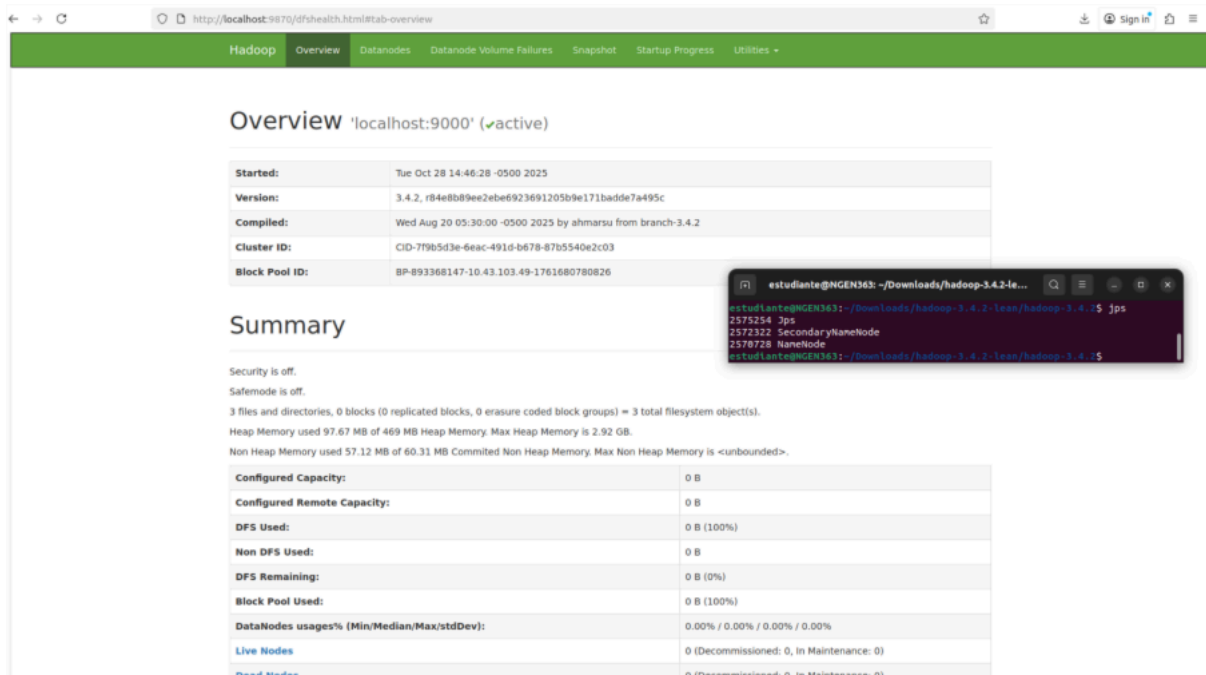


Figura 8. Cluster single node.

Clúster distribuido

El sistema se expandió a una configuración de 4 nodos para lograr una operación completamente distribuida (Multi-Nodo), siguiendo una arquitectura Master-Slave:

Rol de Hadoop	Nodo
NameNode (NN) & ResourceManager (RM)	Master
DataNode (DN) & NodeManager (NM)	Slave 1
DataNode (DN) & NodeManager (NM)	Slave 2
DataNode (DN) & NodeManager (NM)	Slave 3

1. Configuración de red y SSH Multi-Nodo:

- Se aseguró la comunicación de red entre los 4 nodos.

- b. La clave pública de SSH se copió desde el Nodo Master a todos los nodos Slave para permitir la centralización de los daemons.

2. Archivos de Configuración Distribuidos:

- a. Se distribuyeron los archivos de configuración clave a todos los nodos.

etc/hadoop/core-site.xml

- b. Se cambió la referencia de localhost a la dirección IP del Nodo Master (master), permitiendo a los DataNodes externos saber a dónde reportarse, usando las siguientes líneas en el archivo:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>
```

etc/hadoop/hdfs-site.xml

-
- c. Se aumentó el factor de replicación a 3, adecuado para un clúster con tres DataNodes para redundancia, usando las siguientes líneas en el archivo:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
</configuration>
```

etc/hadoop/workers (Solo en el Nodo Master)

- d. Se incluyeron los nombres de host o IPs de los tres nodos Slave para que los scripts de inicio y detención los administren.

3. Ejecución del clúster distribuido

Formato y Reinicio: Después de distribuir los archivos de configuración, se realizó un último \$ bin/hdfs namenode -format en el Master, seguido por el inicio de los servicios:

- \$ sbin/start-dfs.sh
- \$ sbin/start-yarn.sh

4. Validación del clúster distribuido

Se accedió a la interfaz web del NameNode (<http://master:9870/>) para confirmar el estado final del clúster. Como resultado, demostrado en la *Figura 9*, la interfaz muestra el clúster funcionando correctamente. El valor de Configured Capacity es de 135.71 GB, lo que indica que los DataNodes se han registrado correctamente y reportan su espacio de disco. Además, la sección Live Nodes reporta 2 nodos activos (Master y un Slave) o (dos Slaves), confirmando el éxito en la comunicación y en la formación del clúster distribuido.

Overview 'master:9000' (✓active)

Started:	Thu Oct 30 11:21:43 -0500 2025
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 03:22:00 -0500 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-b79eb84b-0341-445e-bf3a-8b2a25b88766
Block Pool ID:	BP-2130398472-10.43.103.236-1761841264030

Summary

Security is off.

Safemode is off.

3 files and directories, 1 blocks (1 replicated blocks, 0 erasure coded block groups) = 4 total filesystem object(s).

Heap Memory used 120.32 MB of 291 MB Heap Memory. Max Heap Memory is 2.92 GB.

Non Heap Memory used 59.59 MB of 62 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	135.71 GB
Configured Remote Capacity:	0 B
DFS Used:	52.58 KB (0%)
Non DFS Used:	39.23 GB
DFS Remaining:	89.51 GB (65.95%)
Block Pool Used:	52.58 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	2 (Decommissioned: 0, In Maintenance: 0)

Figura 9. Cluster.

Análisis de resultados

El proceso permitió comprobar que la arquitectura de Hadoop se basa en la cooperación entre procesos distribuidos. La coordinación entre NameNode y DataNodes aseguró la distribución y tolerancia a fallos del almacenamiento. El uso de SSH sin contraseña y la sincronización del archivo workers resultaron esenciales para la automatización de tareas.

Por lo tanto, tras la prueba del clúster:

- Se verificó la importancia de definir correctamente las rutas de entorno (JAVA_HOME, HADOOP_HOME).
- YARN demostró su capacidad de asignar recursos dinámicamente.
- La práctica consolidó la comprensión de conceptos como replicación, sincronización, comunicación entre procesos y balanceo de carga.

Conclusiones

El sistema de ficheros Hadoop HDFS ofrece una solución eficiente para el almacenamiento distribuido, asegurando redundancia mediante replicación.

La implementación del clúster permitió comprender la arquitectura maestro-esclavo de Hadoop y la comunicación mediante daemons en red.

El ejercicio reforzó los conocimientos de sistemas distribuidos y administración de recursos compartidos.

La correcta configuración de cada componente (SSH, XMLs, variables, directorios) fue clave para alcanzar un funcionamiento estable.

Se logró un entorno funcional capaz de ejecutar tareas de procesamiento paralelo, cumpliendo todos los objetivos académicos propuestos.

Referencias

<https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>

