

Hadoop

Hadoop: Introducción

Hadoop es un framework de procesamiento de datos distribuido que permite el almacenamiento y procesamiento de conjuntos de datos masivos en clústeres de computadoras. Si bien Hadoop no es una base de datos, su componente principal, HDFS (Hadoop Distributed File System), funciona como un sistema de almacenamiento distribuido que se puede combinar con varias bases de datos NoSQL o motores de consulta como Apache Hive o HBase.



Hadoop era el nombre del elefante de peluche del hijo de Doug Cutting (uno de los creadores)

El origen de Hadoop fue el artículo sobre el [Google File System](#) que se publicó en octubre de 2003.


Este artículo generó otro de Google: "[MapReduce: Simplified Data Processing on Large Clusters](#)"

Hadoop: Historia

- 2002-2004:

 **Nutch:** Doug Cutting y Mike Cafarella desarrollan un motor de búsqueda web distribuido.


- 2004:

 **MapReduce (Google):** Inspirados en el artículo de Google, implementan el modelo para procesar grandes volúmenes de datos.


- 2006:

 **Hadoop nace:** Proyecto de código abierto dentro de Apache Software Foundation.


- 2008:

 **Yahoo adopta Hadoop:** Despliega uno de los mayores clústeres Hadoop del mundo y contribuye a su desarrollo.

- 2008:

 **Fundación de Cloudera:** Primera empresa en ofrecer una distribución comercial de Hadoop, fundada por ex ingenieros de Google y Yahoo.

- 2011:

 **Fundación de Hortonworks:** Empresa nacida de ex empleados de Yahoo, dedicada al desarrollo de Hadoop 100% de código abierto.

- 2018:

- **Fusión Cloudera-Hortonworks:** Se combinan en una única plataforma para fortalecer la adopción empresarial de Hadoop.

Hadoop: Componentes

El framework de Apache Hadoop se compone de los siguientes módulos:

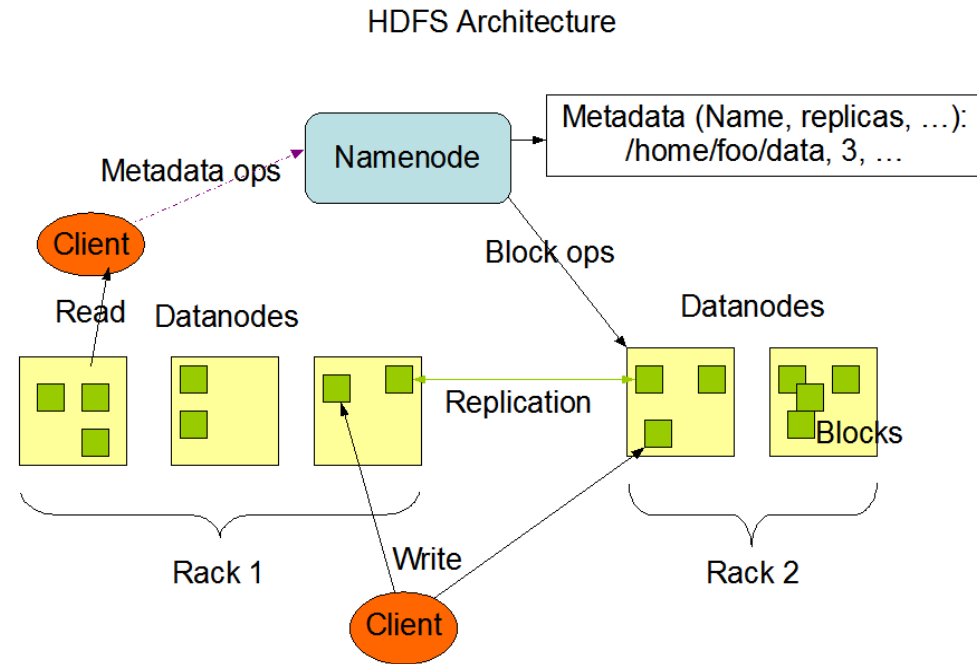
- **Hadoop Common:** contiene bibliotecas y utilidades que necesitan otros módulos de Hadoop.
- **Hadoop Distributed File System (HDFS):** un sistema de archivos distribuido que almacena datos en máquinas comunes, lo que proporciona un ancho de banda agregado muy alto en todo el clúster.
- **Hadoop YARN:** Es una plataforma responsable de administrar recursos informáticos en clústeres y usarlos para programar las aplicaciones de los usuarios.
- **Hadoop MapReduce:** una implementación del modelo de programación MapReduce para el procesamiento de datos a gran escala.
- **Hadoop Ozone:** (introducido en 2020) un almacén de objetos para Hadoop

HDFS

Introducción

HDFS es la capa de almacenamiento de Hadoop, donde los archivos se dividen en bloques y se distribuyen en varias máquinas (nodos). Está diseñado para una alta tolerancia a fallas y es ideal para almacenar grandes conjuntos de datos

- **NodeName:** El nodo maestro en HDFS que administra los metadatos de los archivos (por ejemplo, nombres de archivos, ubicaciones de bloques, permisos). No almacena los datos reales, pero sabe dónde se encuentran los bloques de datos en los DataNodes. Solo existe un NameNode activo a la vez, pero se puede configurar una copia de seguridad (NameNode en espera) para alta disponibilidad.
- **DataNode:** Los que realmente almacenan los datos. Cada archivo en HDFS se divide en bloques (tamaño predeterminado 128 MB) y estos bloques se replican en varios DataNodes para lograr tolerancia a fallos. Los DataNodes también informan el estado de los bloques al NameNode.



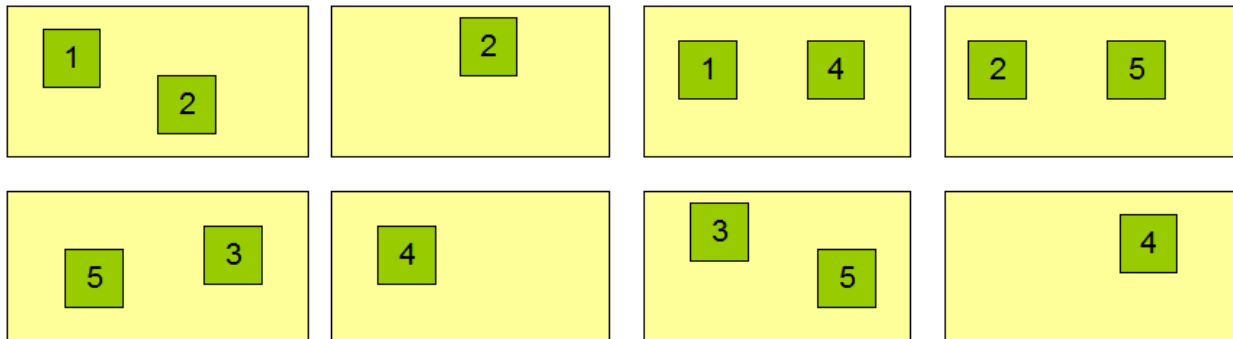
HDFS

Características

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

- Soporta cuotas y permisos de acceso.
- Almacena cada fichero en una secuencia de bloques del mismo tamaño (excepto el último).
- Los bloques son guardados en diferentes DataNodes para tener tolerancia a fallos.
- El número de réplicas se puede definir a nivel de aplicación.
- Está optimizado para realizar pocas escrituras pero muchas lecturas.

HDFS

snapshots

Los **snapshots** en HDFS son una funcionalidad que permite capturar el estado de un directorio en un momento específico en el tiempo. Los snapshots son muy útiles para realizar copias de seguridad y mantener versiones históricas de los datos sin interrumpir las operaciones normales en HDFS.

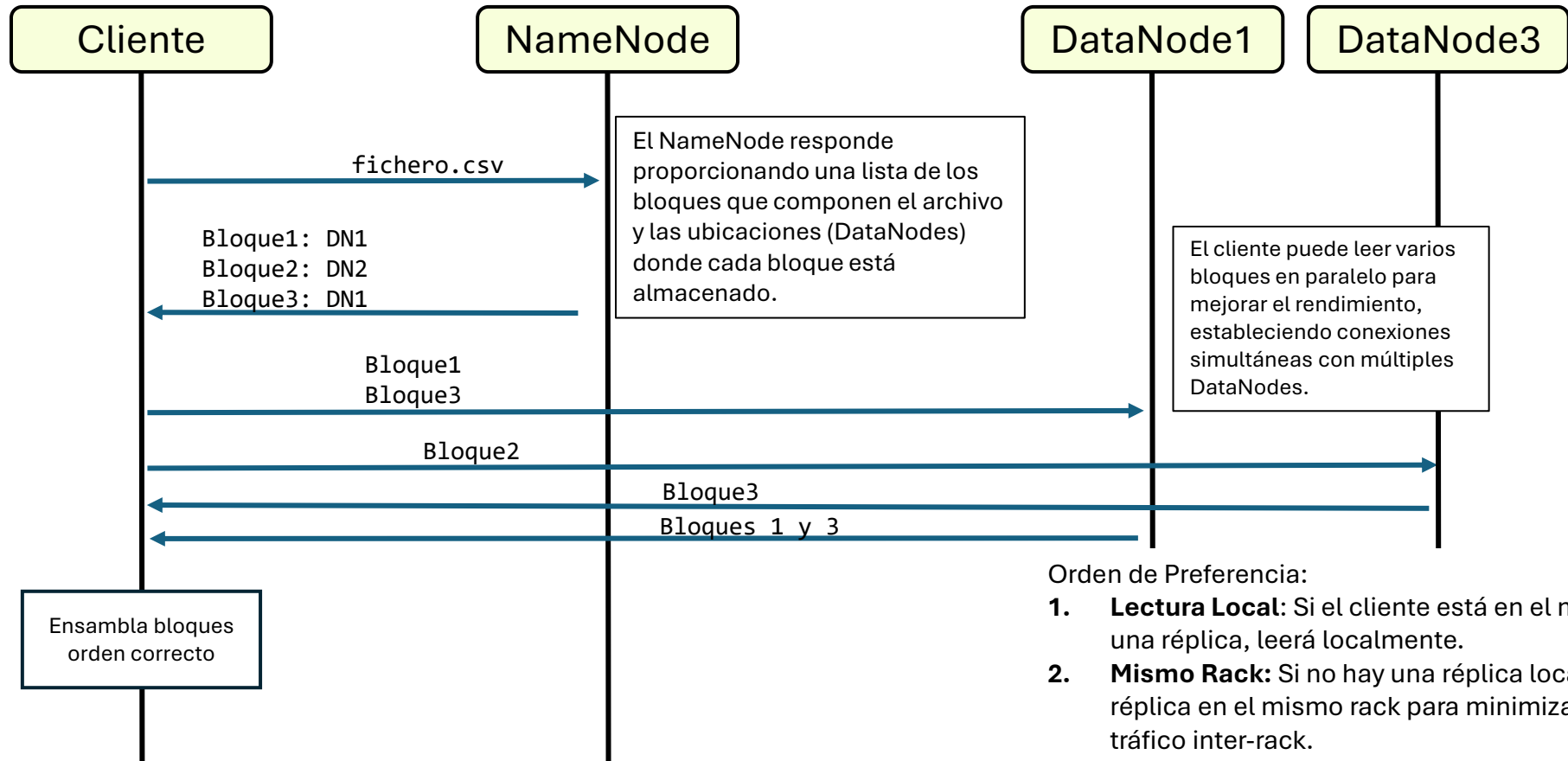


<https://community.cloudera.com/t5/Community-Articles/HDFS-Snapshots-1-Overview/ta-p/247734>

1. **Consistencia en el tiempo:** Un snapshot captura el estado exacto de un directorio en un momento específico
2. **Solo lectura:** Los snapshots son inmutables.
3. **Eficiencia en el uso del espacio:** Los snapshots no duplican los datos en HDFS. Utilizan una técnica llamada **copy-on-write**, donde solo los bloques que se modifican después de crear el snapshot son copiados.
4. **Facilidad para restaurar:** Se puede restaurar el estado de un directorio al estado que tenía en el momento en que se tomó un snapshot.
5. **No afectan el rendimiento:** Los snapshots se pueden crear y eliminar rápidamente sin afectar el rendimiento del sistema ni las operaciones normales de lectura/escritura en HDFS.
6. **Acceso a un snapshot:** Los snapshots se almacenan en un subdirectorio especial llamado `.snapshot` dentro del directorio habilitado. Se puede acceder a los archivos del snapshot como si fueran archivos normales.

HDFS

Flujo lectura

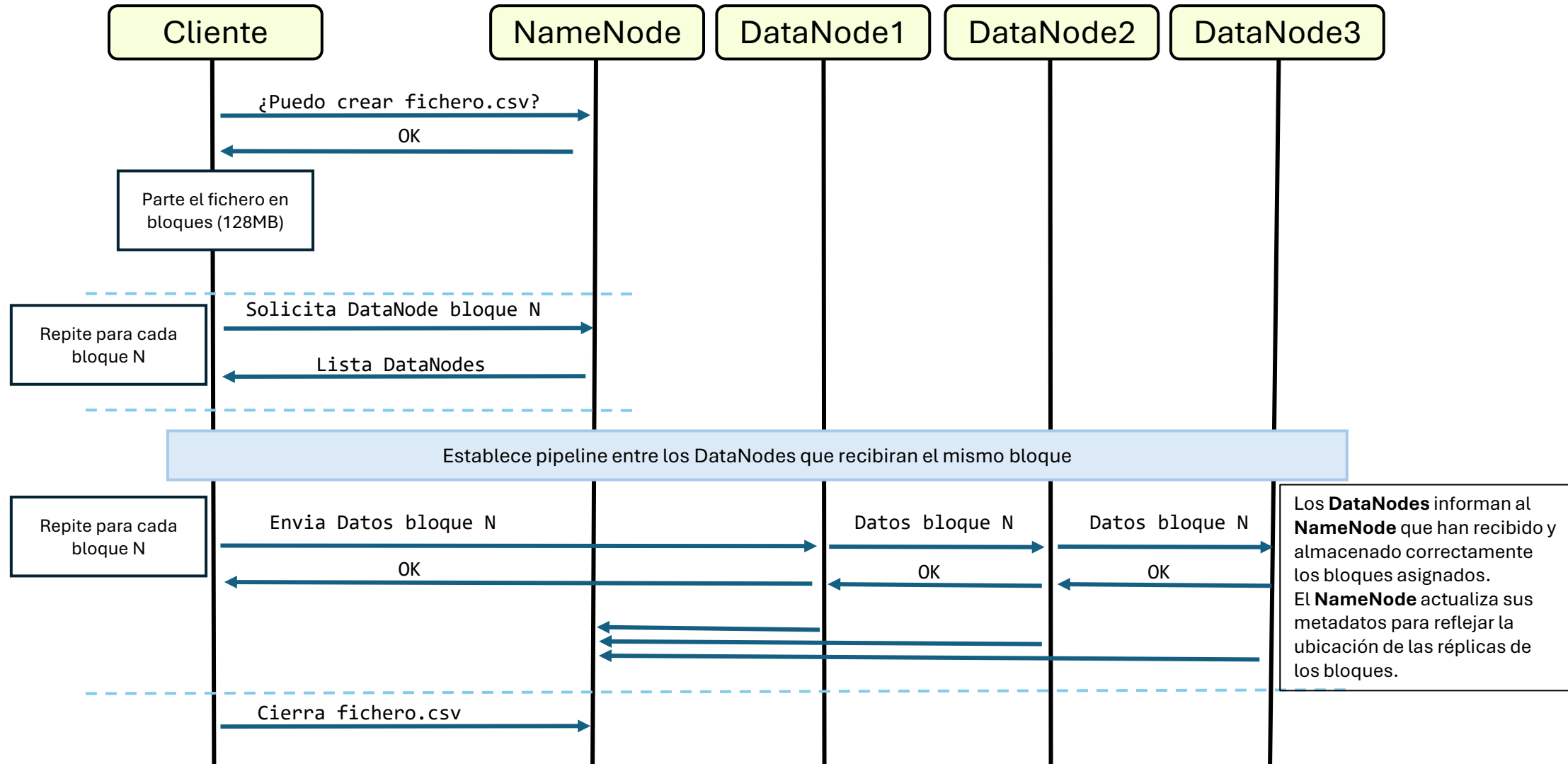


Orden de Preferencia:

1. **Lectura Local:** Si el cliente está en el mismo nodo que una réplica, leerá localmente.
2. **Mismo Rack:** Si no hay una réplica local, buscará una réplica en el mismo rack para minimizar la latencia y el tráfico inter-rack.
3. **Rack Diferente:** Como última opción, leerá desde un DataNode en un rack diferente.

HDFS

Flujo escritura

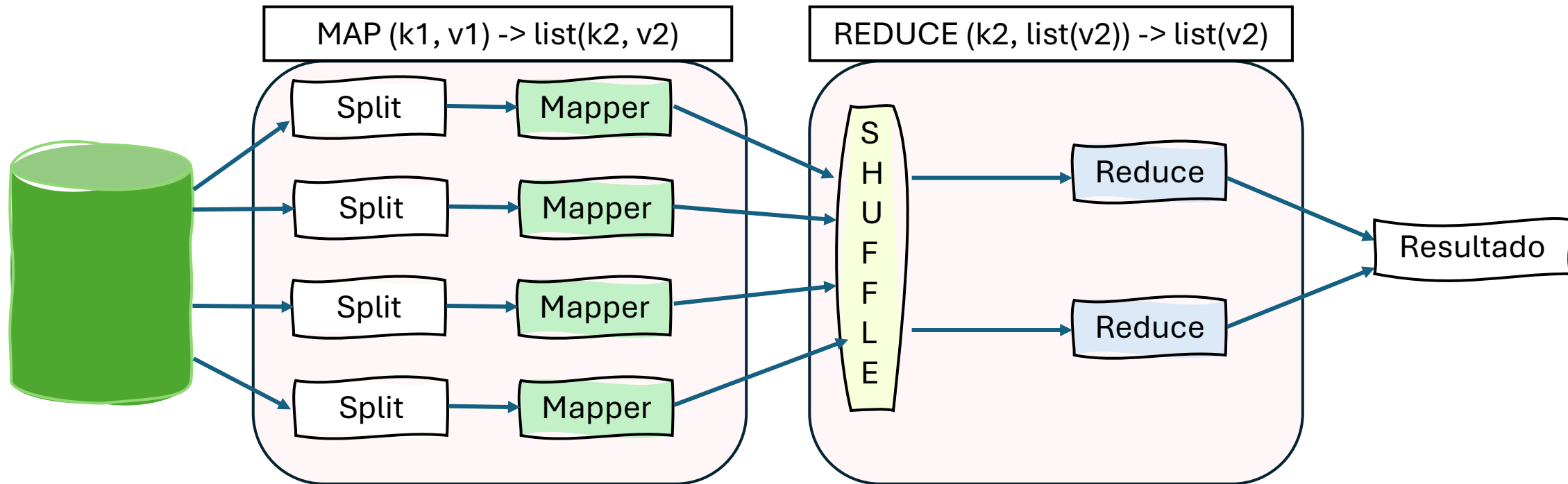


Map Reduce

Modelo de programación para el procesamiento de grandes datasets utilizando computación distribuida.

Este modelo fue publicado originalmente en 2014 por Google Research:

<https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf>

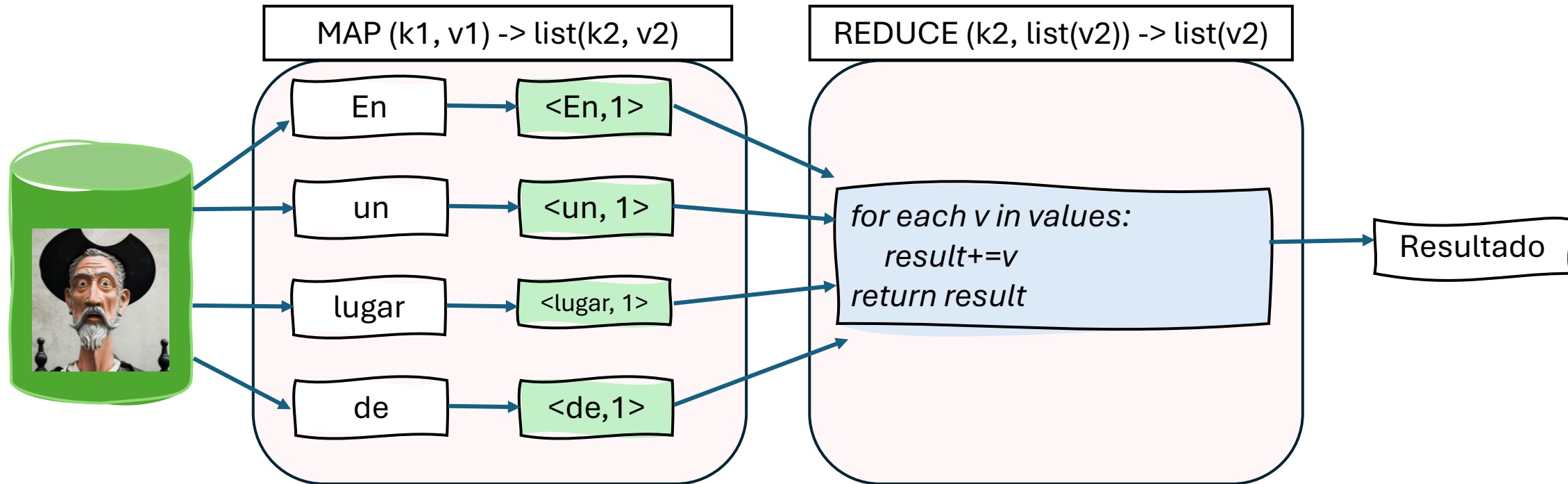


MapReduce: Contando palabras

Vamos a contar las palabras de un documento utilizando MapReduce de forma distribuida

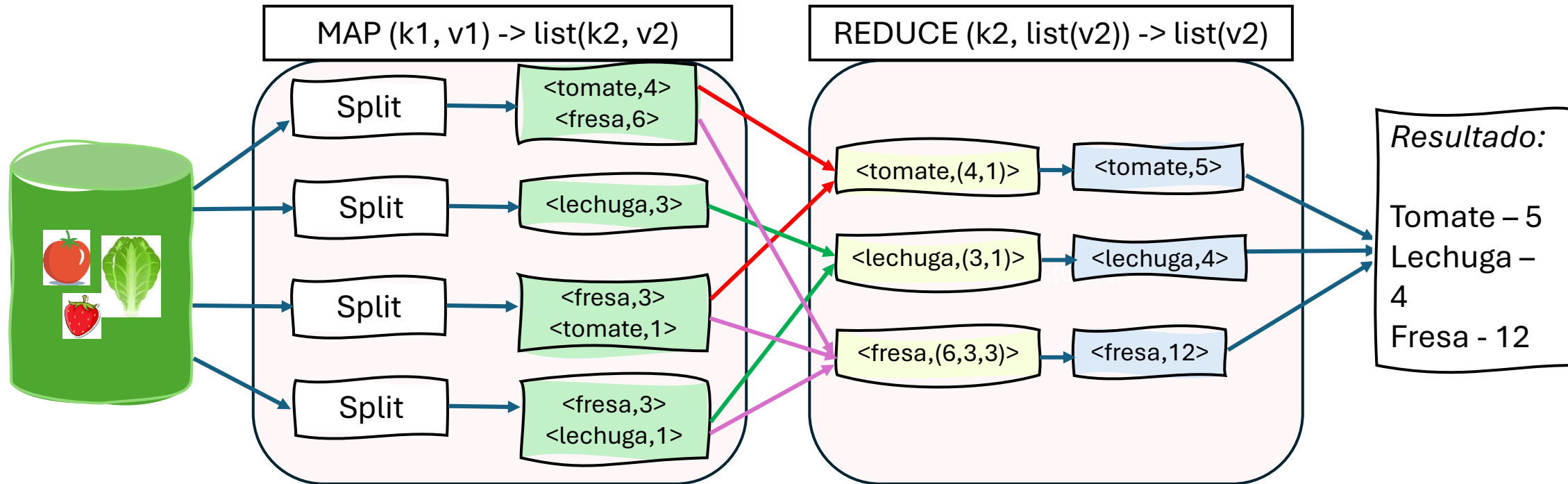
```
map(String key, String value):  
    EmitIntermediate(w, "1");
```

```
reduce(String key, Iterator values):  
    int result = 0;  
    for each v in values:  
        result += ParseInt(v);  
    Emit(AsString(result));
```



Map Reduce: Sumando pesos

Vamos a sumar el peso de diferentes hortalizas



YARN

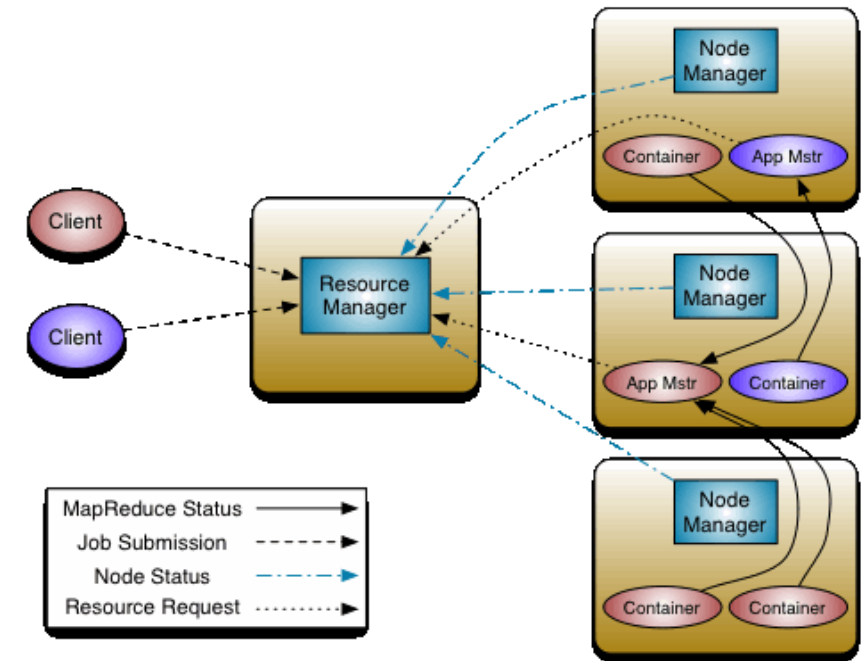
YARN (Yet Another Resource Negotiator) es la capa de gestión de recursos de Hadoop, responsable de asignar recursos del sistema (CPU, memoria, etc.) a las aplicaciones que se ejecutan en el ecosistema Hadoop.

Resource Manager: Gestiona y asigna recursos del clúster entre diferentes aplicaciones. Recibe solicitudes de recursos de aplicaciones y asigna recursos disponibles según políticas.

Node Manager: Se ejecuta en cada nodo trabajador y es responsable de monitorear el uso de recursos (CPU, memoria) de los contenedores (donde se ejecutan las tareas de cálculo). Informa el estado del uso de recursos al ResourceManager.

ApplicationMaster: El ApplicationMaster es una entidad específica para cada aplicación que se ejecuta en el clúster. Su responsabilidad principal es gestionar el ciclo de vida de la aplicación desde la solicitud inicial de recursos hasta su finalización. Hay un ApplicationMaster por aplicación en ejecución.

Container: Un contenedor es la unidad básica de recursos que YARN asigna para ejecutar una tarea específica dentro de una aplicación. El contenedor está formado por una cantidad definida de CPU, memoria y, potencialmente, otros recursos como GPU, que se asignan dinámicamente.



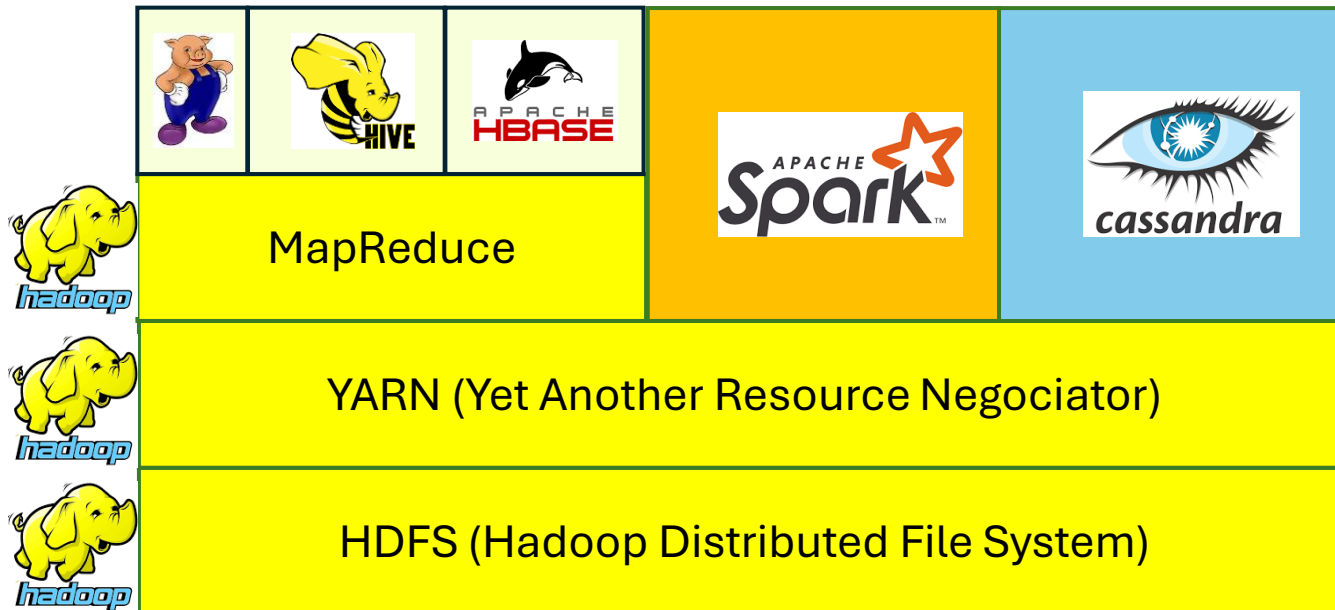
Stack Hadoop

Mientras que MapReduce es solo una de las aplicaciones que puede ejecutarse en YARN, este último puede gestionar otros motores de procesamiento como Apache Spark o Cassandra.

Ejecución de MapReduce en YARN

YARN actúa como la capa de gestión de recursos sobre la cual se ejecuta MapReduce. YARN asigna los recursos necesarios para las tareas de map y reduce. MapReduce es una aplicación que utiliza el sistema de gestión de YARN para realizar el procesamiento distribuido.

Otras aplicaciones como Pig, Hive o Hbase pueden beneficiarse de MapReduce para realizar sus operaciones.



Instalación

1. Instalar Java 8
2. Ir a <https://hadoop.apache.org/> y descargar la última versión
3. Configurar variables de entorno
4. Configurar los ficheros
 - \$HADOOP_HOME/etc/hadoop/hdfs-site.xml.
 - \$HADOOP_HOME/etc/hadoop/core-site.xml
5. Formatea NameNode para inicializar el sistema de archivos HDFS.
6. Inicia todos los servicios: *start-all.sh*

NameNode Web UI (HDFS)

http://myubuntu:9870/

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Overview 'myubuntu.home:9000' (✔active)

Started:	Wed Sep 18 10:31:09 +0200 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaee760
Compiled:	Mon Mar 04 07:35:00 +0100 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-f7863fdb-32cc-4883-89c2-ca22cc979c5e
Block Pool ID:	BP-639621154-127.0.1.1-1726183267040

Summary

Security is off.

Safemode is off.

170 files and directories, 74 blocks (74 replicated blocks, 0 erasure coded block groups) = 244 total filesystem object(s).

Heap Memory used 208.23 MB of 322.5 MB Heap Memory. Max Heap Memory is 1.72 GB.

Non Heap Memory used 56.39 MB of 57.75 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Visualiza el estado del sistema de archivos distribuido (HDFS).

Puedes ver estadísticas de almacenamiento, detalles sobre bloques y el estado general de los DataNodes.

ResourceManager Web UI (YARN)

http://myubuntu:8088/



▼ Cluster

[About](#)
[Nodes](#)
[Node Labels](#)
[Applications](#)
 [NEW](#)
 [NEW_SAVING](#)
 [SUBMITTED](#)
 [ACCEPTED](#)
 [RUNNING](#)
 [FINISHED](#)
 [FAILED](#)
 [KILLED](#)

[Scheduler](#)

► Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps C
0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes
<u>1</u>	<u>0</u>

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Min
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024

Show 20 ▼ entries						
ID ▼	User	Name	Application Type	Application Tags	Queue	Application Priority

Showing 0 to 0 of 0 entries

El ResourceManager maneja los recursos del clúster y la planificación de trabajos YARN. Su interfaz web permite visualizar el estado de los nodos, las aplicaciones en ejecución y el historial de trabajos.

DataNode Web UI (HDFS)

http://myubuntu:9864/

DataNode on myubuntu:9866

Cluster ID:	CID-f7863fdb-32cc-4883-89c2-ca22cc979c5e
Started:	Wed Sep 18 10:31:12 +0200 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaee760

Block Pools

Namenode Address	Namenode HA State	Block Pool ID	Actor State	Last Heartbeat Sent	Last Heartbeat Response	Last Block Report	Last Block Report Size (Max Size)
myubuntu.home:9000	active	BP-639621154-127.0.1.1-1726183267040	RUNNING	1s	1s	27 minutes	776 B (128 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hadoop/data/hdfs/datanode	DISK	93.68 MB	8.14 GB	0 B	0 B	74

La interfaz web del DataNode te permite ver información específica de cada DataNode, como el estado del almacenamiento y el rendimiento.

NodeManager Web UI (YARN)

http://myubuntu:8042/



NodeManager information

▸ ResourceManager	NodeManager information	
▸ NodeManager		
▼ Tools		
Configuration		
Local logs		
Server stacks		
Server metrics		
	Total Vmem allocated for Containers	12.60 GB
	Vmem enforcement enabled	true
	Total Pmem allocated for Container	6 GB
	Pmem enforcement enabled	true
	Total VCores allocated for Containers	2
	Resource types	memory-mb (unit=Mb), vcores
	NodeHealthyStatus	true
	LastNodeHealthTime	Wed Sep 18 08:59:21 UTC 2024
	NodeHealthReport	
	NodeManager started on	Wed Sep 18 08:31:19 UTC 2024
	NodeManager Version:	3.4.0 from bd8b77f398f626bb7791783192ee7a5dfaee760 by root source checksum 934da0c5743762b7851cfcff9f8ca2 on 2024-03-04T06:54Z
	Hadoop Version:	3.4.0 from bd8b77f398f626bb7791783192ee7a5dfaee760 by root source checksum f7fe694a3613358b38812ae9c31114e on 2024-03-04T06:35Z

El NodeManager gestiona los recursos locales en cada nodo del clúster YARN. Su interfaz web muestra información sobre los contenedores que está ejecutando y el uso de recursos.