

Administración y monitorización de sistemas Hadoop.

Caso práctico

El Banco Español de Inversiones, BEI, es uno de los principales bancos nacionales de banca privada de España, con más de 100 años de historia.

Su modelo de negocio se ha centrado en grandes inversores, y el principal valor añadido que les aporta es que dispone de un grupo de expertos de gran reputación mundial que recomienda a los clientes qué inversiones realizar para maximizar la rentabilidad.

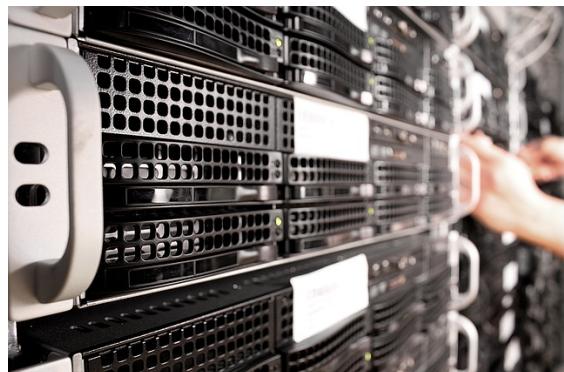
Hasta ahora, su negocio estaba muy focalizado en el cliente y en el gran conocimiento interno, y nunca se había planteado una apuesta por una disruptión tecnológica para aportar mayor valor a los clientes por la tecnología en lugar de por el conocimiento de negocio tradicional. Es por ello que a nivel de arquitectura de sistemas, tiene casi todo su negocio en un sistema mainframe junto con una herramienta de CRM.

Sin embargo, recientemente su consejo de administración ha nombrado nuevos consejeros que pretenden dar una revolución al banco, incorporando la tecnología para ayudar a los clientes y a su grupo de expertos a tomar mejores decisiones con modelos predictivos, o dando funcionalidades en cualquier canal, tanto web como móvil u oficinas.

Dentro de esta nueva estrategia que quiere abordar el banco, han decidido invertir en tecnologías Big Data, y por ello han implantado una plataforma Hadoop a la que han volcado todos los datos del mainframe y el CRM. Tienen un equipo de científicos de datos trabajando ya en el clúster, desarrollando modelos predictivos de los mercados o de valoración, así como analistas de negocio que, mediante herramientas de visualización, realizan su trabajo pudiendo cruzar cualquier dato que exista en el banco, algo que antes era imposible, ya que los datos eran poco accesibles para estos analistas.

El equipo de IT, que está liderado por María Robles, debe manejar ahora todo el clúster, ya que son cada vez más los usuarios que acceden a él para lanzar consultas, ver datos o ejecutar modelos.

María y su equipo están viendo qué capacidades les ofrece Hadoop para la administración de la plataforma.



[Colossus Cloud](#) (Dominio público)

Hadoop es una plataforma que requiere un mayor esfuerzo de monitorización o administración, principalmente por estos motivos:

- ✓ Tiene un **amplio ecosistema de componentes**, cada uno desarrollado de forma independiente, con su propio funcionamiento, versión, sus propios ficheros de configuración o métricas de monitorización.
- ✓ Se ejecuta en un **entorno distribuido**, es decir, en un conjunto de servidores que puede llegar a ser numeroso. Cada servidor tiene su sistema operativo, y por supuesto, su propio hardware que puede tener fallos.
- ✓ Hadoop es una plataforma con un nivel de **madurez inferior** a otras soluciones que tienen una base instalada muy grande y la experiencia de un fabricante que se traduce en una documentación muy rica, unas herramientas de administración potentes, etc.

La administración de un sistema Hadoop, por lo tanto, no es algo trivial. Afortunadamente, existen varias herramientas que facilitan esta labor, entre las que destacan:

- ✓ Las consolas de **monitorización de HDFS** (Namenode UI) y **YARN** (ResourceManager UI), que ofrecen una visión simple de lo que ocurre en cada servicio, permitiendo acceder a bajo nivel hasta ver los logs de las aplicaciones o procesos que se están ejecutando.
- ✓ **Apache Ambari y Cloudera Manager**, que son sin lugar a dudas las herramientas más potentes y a su vez más fáciles de utilizar para la instalación y la administración de plataformas Hadoop, ya que ofrecen funcionalidades desde la visualización de cuadros de mando con cualquier métrica de salud del sistema, la creación de alarmas, la modificación de la configuración del sistema o la administración de usuarios.
- ✓ **Ganglia**, que es una herramienta opensource para monitorizar clústeres de servidores. Aunque no es una herramienta específica para Hadoop, puede ser útil para la monitorización de los sistemas.

En esta unidad vamos a conocer estas herramientas:

- ✓ En primer lugar, conoceremos los interfaces web de HDFS y YARN.
- ✓ A continuación, entraremos en detalle en Apache Ambari, así como Cloudera Manager.
- ✓ Por último, conoceremos cómo utilizar Ganglia para monitorizar plataformas Hadoop.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

1.- Introducción a la administración de Hadoop.

Caso práctico

María Robles, la responsable de IT del Banco Español de Inversiones tiene más de 20 años de experiencia en la administración de todo tipo de sistemas que han utilizado en el banco.

Ahora bien, administrar una plataforma Hadoop es un reto para ella y su equipo, sabe que tienen muchos servidores y muchos servicios en cada servidor, ¿cómo van a poder administrar un sistema tan complejo?



[Csaba Nagy](#) (Dominio público)

Para la administración de una plataforma Hadoop se puede distinguir tres tipos de actividades: la configuración, operación de los servicios y la monitorización.

En este tema nos centraremos en la monitorización, ya que la configuración suele ser más estática, y no requiere grandes intervenciones una vez la plataforma ha sido instalada, y la operación de los servicios son tareas sencillas que consisten en el arranque, parada o reinicio de los mismos. En cualquier caso, ahora veremos cómo se configura o parametriza una plataforma Hadoop, para posteriormente entrar a detalle en la tarea más habitual, la monitorización de la plataforma.

Configuración

La configuración de un clúster Hadoop se basa en ficheros XML asociados a cada servicio de la plataforma, es decir, encontrarás ficheros de configuración de HDFS, YARN, Hive, etc. por separado. Los ficheros de configuración suelen encontrarse en los servidores donde se ejecuta el servicio, habitualmente en los directorios /etc/conf.

A continuación vamos a ver los principales ficheros de configuración de los servicios más importantes:

- ✓ **Hadoop core (HDFS, YARN y MapReduce):** para configurar estos servicios, existen varios ficheros de configuración en los nodos master donde se ejecuta cada servicio:
 - ◆ **core-site.xml:** en este fichero aparece la configuración común de los componentes core, en ella encontraremos algunas propiedades de configuración como:
 - **fs.defaultFS**: que indica el endpoint de HDFS al que deben apuntar los clientes.
 - **hadoop.security.authentication**: tipo de autenticación requerida (Kerberos, simple).
 - ◆ **hdfs-site.xml:** este fichero contiene la configuración específica de HDFS, como por ejemplo las siguientes propiedades:
 - **fs.namenode.name.dir**: directorio local en el servidor donde se aloja la información del Namenode.
 - **dfs.datanode.data.dir**: directorio local en el servidor worker donde se almacenan los bloques de HDFS.
 - **dfs.namenode.http-address**: dirección en la que se va a arrancar una consola web de monitorización de HDFS.
 - ◆ **yarn-site.xml:** este fichero contiene la configuración específica de YARN, como por ejemplo las siguientes propiedades:
 - **yarn.resourcemanager.scheduler.class**: tipo de Scheduler que se va a utilizar para la gestión de las prioridades de las aplicaciones.

- `yarn.resourcemanager.resource-tracker.address`: dirección del ResourceManager al que se conectarán los NodeManager.
 - `yarn.nodemanager.log-dirs`: directorio en los nodos worker en los que se escribirá el log de la ejecución de las aplicaciones.
 - `yarn.resourcemanager.webapp.address`: dirección en la que se va a arrancar una consola web de monitorización de YARN.
- ✓ **Apache Hive**: utiliza un fichero `hive-site.xml` para almacenar la configuración principal del servicio. Este fichero puede contener centenares de parámetros de configuración, aunque los principales son los siguientes:
- ◆ `hive.execution.engine`: indica qué motor se utilizará para la ejecución de consultas, teniendo 3 posibilidades:
 - MapReduce.
 - Tez.
 - Spark.
 - ◆ `hive.server2.enable.doAs`: permite ejecutar las queries con el usuario que hizo la petición, en lugar de con el usuario de sistema con el que se ejecuta Hive.
 - ◆ `hive.server2.thrift.port`: puerto en el que se levantará el servidor Thrift al que se conectarán los clientes de Hive.
- ✓ **Apache Spark**: utiliza el fichero `spark-defaults.conf`, que no es de tipo XML a diferencia de los ficheros de configuración que suelen tener las herramientas del ecosistema Hadoop. Este fichero sirve para parametrizar los valores por defecto, que se pueden modificar mediante código por las aplicaciones. Algunos de los parámetros más importantes que contiene este fichero son los siguientes:
- ◆ `spark.yarn.historyServer.address`: contiene la ruta de YARN donde se ejecuta Spark.
 - ◆ `spark.history.ui.port`: puerto en el que se levantará un interfaz web de monitorización de Spark.
 - ◆ `spark.eventLog.dir`: ruta en la que se almacenarán los ficheros de log con la ejecución de las aplicaciones.

Estos son algunos de los principales ficheros y parámetros de configuración de Spark. Se pueden modificar de dos formas:

- ✓ Accediendo por consola a los ficheros y modificándolos con un editor tipo `vi`.
- ✓ Mediante una herramienta de administración como Apache Ambari o Cloudera Manager.

En cualquier caso, después de la modificación de algún parámetro, suele ser necesario reiniciar el servicio implicado, lo cual puede llevar un tiempo de pérdida de servicio.

Monitorización

En primer lugar, es preciso conocer que la mayoría de los clústers de Hadoop se ejecutan en sistemas Linux y, por lo tanto, es importante conocer los aspectos clave de la monitorización de Linux. Si el sistema Linux sobre el que se ejecuta Hadoop en cada servidor tiene un cuello de botella o un problema de rendimiento, Hadoop no podrá tener un buen funcionamiento.

Las principales variables que se monitorizan en un sistema Linux son:

- ✓ **Uso de CPU**: es normal encontrar picos de consumo de CPU, aunque éstos no deberían ser constantes o durar más de unos segundos. Es importante revisar la media de consumo además, así como los procesos que consumen procesador. El principal comando para monitorizar el uso de CPU es `top`.
- ✓ **Uso de la memoria**: la memoria es uno de los primeros lugares en los que se debe buscar cuando hay problemas de rendimiento. Si tiene memoria (RAM) inadecuada en el servidor, el sistema puede ralentizarse debido al intercambio excesivo de memoria. El intercambio de memoria significa que el sistema está transfiriendo páginas de memoria a dispositivos de disco para liberar memoria para otros procesos. Los comandos que se utilizan principalmente para monitorizar la memoria son `vmstat`, `meminfo` y `free`.
- ✓ **Almacenamiento en disco**: cuando se trata de monitorizar discos se debe buscar dos cosas: en primer lugar, hay que verificar que no se esté quedando sin espacio: las aplicaciones agregan más datos de forma continua y es inevitable que tenga que agregar más espacio de almacenamiento constantemente. En segundo lugar, observar el rendimiento del disco: ¿hay cuellos de botella debido a un rendimiento lento de entrada/salida del disco? Los comandos utilizados para monitorizar el disco son fundamentalmente `iostat` y `sar`.
- ✓ **Tráfico de red**: la red es un componente importante de su sistema: si las conexiones de red son lentas, Hadoop funcionará lentamente. Las estadísticas de red simples, como la cantidad de bytes recibidos y enviados, ayudarán a identificar problemas de red. El principal comando para monitorizar la red es `dstat`.

Además de la monitorización de los sistemas Linux sobre los que se ejecuta, Hadoop dispone de varias herramientas de monitorización o administración que veremos a continuación:

- ✓ Dos interfaces sencillos que proporcionan el ResourceManager y el Namenode.
- ✓ Apache Ambari y Cloudera Manager como herramientas de administración y monitorización más complejas.
- ✓ Ganglia como herramienta de monitorización general para clústers de servidores.

Para saber más

En la web oficial de Apache Hadoop, en la [página de documentación](#), encontrarás el significado de todas las variables de configuración de Hadoop.

En el menú de la izquierda, lo encontrarás en la parte inferior, en la zona "Configuration".

Autoevaluación

¿Por qué es importante monitorizar las métricas de los sistemas operativos Linux sobre los que se ejecuta Hadoop?

- Porque toda la configuración está en ficheros XML.
- Porque Linux es un sistema inestable y puede originar problemas de ejecución a Hadoop.
- Porque Hadoop supone que hay muchos servicios ejecutándose sobre la misma máquina, y podría haber conflicto entre los servicios que se podrían visualizar a nivel de sistema operativo.

Incorrecto: dónde se almacena la información es irrelevante para la monitorización.

Incorrecto: Linux es un sistema muy estable. En caso de que tuviera problemas el sistema operativo, desde luego afectarían a Hadoop, pero en este caso, no podemos considerar que sea una respuesta correcta porque Linux no es un sistema inestable.

Correcto: es importante monitorizar las métricas del sistema operativo ya que al fin y al cabo Hadoop se ejecuta sobre él, pidiéndole los recursos de memoria y procesador, y además, Hadoop implica que cada Linux tenga muchos procesos ejecutándose en la misma máquina.

Solución

- 1. Incorrecto
- 2. Incorrecto
- 3. Opción correcta

2.- Interfaz de HDFS: Namenode UI.

Caso práctico

María Robles, la responsable de IT del Banco Español de Inversiones, BEI, tiene el reto de administrar con su equipo la plataforma Hadoop implantada en el banco, que ya dispone de más de 20 usuarios entre científicos de datos y analistas de negocio.

En primer lugar le preocupa conocer el estado de la capa de almacenamiento, que al fin y al cabo es la base de toda la plataforma. Han ingestado 120 terabytes de datos provenientes del mainframe y de otras herramientas, y le preocupa que pueda haber datos corruptos, ficheros con bloques que no tienen un nivel suficiente de replicación, o simplemente, que se puedan caer nodos sin que su equipo se dé cuenta.

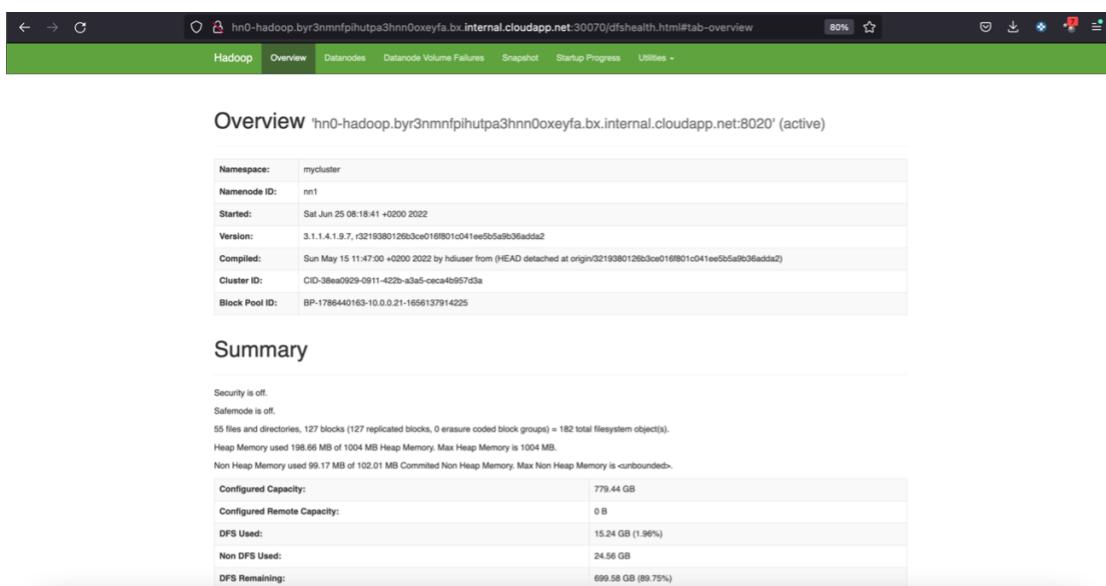
En primer lugar va a estudiar el interfaz que ofrece HDFS por defecto, que se llama Namenode UI.



[ds_30](#) (Dominio público)

HDFS ofrece por defecto una web de administración denominada Namenode UI, o Web UI. Se trata de un servicio web que se arranca en el servidor donde se ejecuta el Namenode, por defecto en el puerto 50070, aunque la ruta se puede configurar en el parámetro `dfs.http.address` del fichero de configuración `hdfs-site.xml`.

Al acceder a esta web se muestra una pantalla de información general:



Overview 'hn0-hadoop.byr3nmnfpihutpa3hnn0oxeyfa.bx.internal.cloudapp.net:8020' (active)

Namespace:	mycluster
Namenode ID:	nn1
Started:	Sat Jun 25 08:18:41 +0200 2022
Version:	3.1.1.4.1.9.7, r3219380126b0ce016fb01c041ee5b5a9b36adda2
Compiled:	Sun May 15 11:47:00 +0200 2022 by hduser from (HEAD detached at origin/r3219380126b0ce016fb01c041ee5b5a9b36adda2)
Cluster ID:	CID-38ea0929-0911-4220-a3a5-ceca4b957d3a
Block Pool ID:	BP-178640163-10.0.0.21-1856137914225

Summary

Configured Capacity:	779.44 GB
Configured Remote Capacity:	0 B
DFS Used:	15.24 GB (1.99%)
Non DFS Used:	24.56 GB
DFS Remaining:	699.58 GB (89.75%)

Iñigo Sanz (Dominio público)

Accediendo a la pestaña de Datanodes, se obtiene la información sobre los diferentes Datanodes, viendo su dirección IP, cuándo se obtuvo un mensaje de check por última vez, la capacidad y el uso de disco, así como algún dato menos importante:

Configured Capacity:	779.44 GB
Configured Remote Capacity:	0 B
DFS Used:	15.24 GB (1.96%)
Non DFS Used:	24.56 GB
DFS Remaining:	699.58 GB (89.75%)
Block Pool Used:	15.24 GB (1.96%)
DataNodes usages% (Min/Median/Max/StdDev):	0.00% / 0.00% / 7.82% / 3.39%
Live Nodes	4 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	Sat Jun 25 08:28:41 +0200 2022
Last Checkpoint Time	Sat Jun 25 08:18:35 +0200 2022

Íñigo Sanz (Dominio público)

Pinchando sobre alguno de los Datanodes que aparecen en el listado, se puede ver el detalle sobre los bloques que contiene, la capacidad, la ruta de HDFS en el disco local, etc.

[hn0-hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30070/dfshealth.html#tab-datanode](#)

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities -

Datanode Information

Legend: ✓ In service ● Down ○ Decommissioned ○ Decommissioned & dead △ In Maintenance & dead

Datanode usage histogram

In operation

Show: 25 entries	Search: <input type="text"/>						
Node	Http Address	Last contact	Last health report	Capacity	Blocks	Block size used	Version
✓en0 hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010 (16.0.0.12.30010)	http://en0.hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010	1a	29m	194.86 GB	0	44 KB (0%)	3.1.1.4.1.8.7
✓en1- hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010 (16.0.0.14.30010)	http://en1-.hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010	1a	29m	194.86 GB	124	15.24 GB (7.82%)	3.1.1.4.1.8.7
✓en2- hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010 (16.0.0.13.30010)	http://en2-.hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010	0s	17m	194.86 GB	0	40 KB (0%)	3.1.1.4.1.8.7
✓en3- hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010 (16.0.0.11.30010)	http://en3-.hadoop/byr3nmnfpihutpa3hnn0oxyeyfa.bx.internal.cloudapp.net:30010	0s	167m	194.86 GB	3	324 KB (0%)	3.1.1.4.1.8.7

Showing 1 to 4 of 4 entries

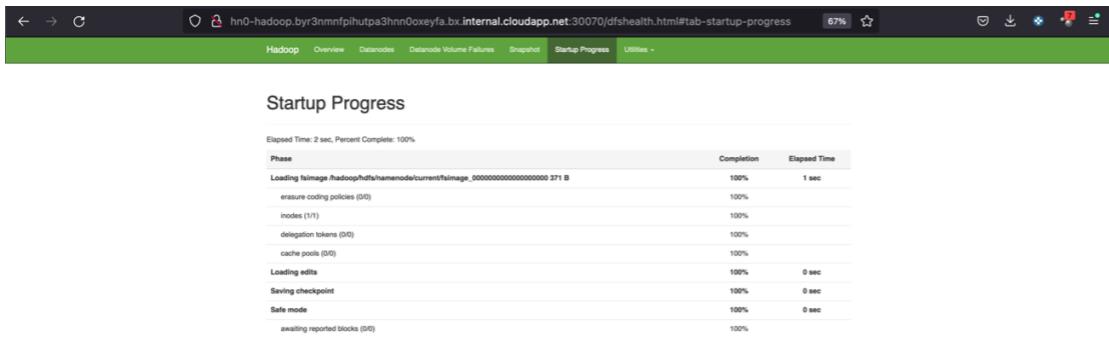
Íñigo Sanz (Dominio público)

En esta misma pantalla, en la pestaña Utilities, se puede acceder a los logs de HDFS en el Datanode:

Íñigo Sanz (Dominio público)

Volviendo a la pantalla principal, se puede encontrar otras pestañas

- ✓ **Datanode Volume Failures:** en la que se muestran los errores en disco que se han detectado en HDFS.
 - ✓ **Snapshot:** en caso de haberse realizado una copia de la estructura de HDFS para poder restaurarlo, en esta pantalla aparecen todos los puntos generados.
 - ✓ **Startup progress:** muestra cómo ha sido el proceso de arranque de HDFS y el estado en el que se encuentra ahora:



Íñigo Sanz (Dominio público)

- Por último, en la pestaña de **Utilidades**, se puede encontrar diferentes funcionalidades, como las siguientes:



Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities -

Browse Directory

File System								Gof	Import	Export	Help
Show	25	- entries							Search:		
□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	Actions		
□	drwx-wx-w	hive	hdfs	0 B	Jun 25 08:20	0	0 B	tmp	View Edit Delete		

Madison 2018

Íñigo Sanz (Dominio público)

✓ Ver los logs de HDFS:

Directory: /logs/			
gc.log-202206250618	11927 bytes	Jun 25, 2022 9:11:51 AM	
hadoop-hdfs-namenode-hail-hadoop.log	349771 bytes	Jun 25, 2022 9:05:45 AM	
hadoop-hdfs-namenode-hail-hadoop.out	7071 bytes	Jun 25, 2022 9:00:34 AM	
hadoop-hdfs-rkfc-hail-hadoop.log	101700 bytes	Jun 25, 2022 6:18:48 AM	

Íñigo Sanz (Dominio público)

Íñigo Sanz (Dominio público)

- ✓ El fichero de configuración de HDFS:

```
<configuration>
  <property>
    <name>mapreduce.jobhistory.jhist.format</name>
    <value>text</value>
    <final>false</final>
    <source>mapred-default.xml</source>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop1:9000</value>
    <final>false</final>
    <source>mapred-default.xml</source>
  </property>
  <property>
    <name>mapreduce.proxyuser.hive.groups</name>
    <value><*></value>
    <final>false</final>
    <source>mapred-site.xml</source>
  </property>
  <property>
    <name>mapreduce.job.block.access.tokens.lifetime</name>
    <value>600</value>
    <final>false</final>
    <source>hdfs-default.xml</source>
  </property>
  <property>
    <name>mapreduce.application.framework.path</name>
    <value>/hadoop-mapreduce/hadoop-mapreduce$ (hdp.version)/mapreduce/tar/gar/file-framework</value>
    <final>false</final>
    <source>mapred-site.xml</source>
  </property>
  <property>
    <name>mapreduce.job.heap.memory-mb.ratio</name>
    <value>0.8</value>
    <final>false</final>
    <source>mapred-default.xml</source>
  </property>
  <property>
    <name>mapreduce.map.log.level</name>
    <value>INFO</value>
    <final>false</final>
    <source>mapred-site.xml</source>
  </property>
  <property>
    <name>dfs.namenode.lazyscrub.file.scrub.interval.sec</name>
    <value>300</value>
    <final>false</final>
    <source>mapred-default.xml</source>
  </property>
```

Íñigo Sanz (Dominio público)

- ✓ Y un volcado de los hilos en ejecución en HDFS para poder analizar errores:

```
Process Thread Dump:
Thread 1185 [IPC Client (1125231526) connection to zk1-hadoop.byr3nmfpilhuptha3hnno0oxyfa.bx.internal.cloudapp.net:8070/stacks
State: TIMED_WAITING
Blocked count: 37
Waited count: 18
Stack:
sun.lang.Object.wait(Native Method)
org.apache.hadoop.ipc.Client$Connection.waitForWork(Client.java:1818)
org.apache.hadoop.ipc.Client$Connection.runClient(Client.java:1863)
Thread 1184 [IPC Client (1125231526) connection to zk1-hadoop.byr3nmfpilhuptha3hnno0oxyfa.bx.internal.cloudapp.net:8070/stacks
State: TIMED_WAITING
Blocked count: 37
Waited count: 18
Stack:
sun.lang.Object.wait(Native Method)
org.apache.hadoop.ipc.Client$Connection.waitForWork(Client.java:1818)
org.apache.hadoop.ipc.Client$Connection.runClient(Client.java:1863)
Thread 1183 [IPC Client (1125231526) connection to zk1-hadoop.byr3nmfpilhuptha3hnno0oxyfa.bx.internal.cloudapp.net:8070/stacks
State: TIMED_WAITING
Blocked count: 37
Waited count: 17
Stack:
sun.lang.Object.wait(Native Method)
org.apache.hadoop.ipc.Client$Connection.waitForWork(Client.java:1818)
org.apache.hadoop.ipc.Client$Connection.runClient(Client.java:1863)
Thread 1099 [IPC Parameter Sending Thread #20]:
State: NEW
Blocked count: 0
Waited count: 41
Stack:
sun.misc.Unsafe.park(Native Method)
java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
java.util.concurrent.SynchronousQueue$TransferStack.awaitIfFull(SynchronousQueue.java:468)
java.util.concurrent.SynchronousQueue$TransferStack.transfer(SynchronousQueue.java:362)
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1073)
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
java.lang.Thread.run(Thread.java:748)
Thread 1098 [IPC Parameter Sending Thread #19]:
State: TIMED_WAITING
Blocked count: 0
Waited count: 35
Stack:
sun.misc.Unsafe.park(Native Method)
java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
java.util.concurrent.SynchronousQueue$TransferStack.awaitIfFull(SynchronousQueue.java:468)
java.util.concurrent.SynchronousQueue$TransferStack.transfer(SynchronousQueue.java:362)
java.util.concurrent.SynchronousQueue$Poller.synchronousQueue$pollSync(SynchronousQueue.java:941)
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1073)
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
java.lang.Thread.run(Thread.java:748)
Thread 1097 [IPC Parameter Sending Thread #218]:
State: TIMED_WAITING
Blocked count: 0
Waited count: 36
Stack:
sun.misc.Unsafe.park(Native Method)
java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
java.util.concurrent.SynchronousQueue$TransferStack.awaitIfFull(SynchronousQueue.java:468)
java.util.concurrent.SynchronousQueue$TransferStack.transfer(SynchronousQueue.java:362)
```

Íñigo Sanz (Dominio público)

Autoevaluación

Indica si las siguientes afirmaciones son correctas sobre Namenode UI

Permite monitorizar el estado de HDFS, pero no permite realizar acciones de administración como para o arrancar el servicio, o modificar la configuración

- Verdadero Falso

Verdadero

Verdadero: no ofrece funcionalidades para administrar el servicio.

Da muy poca información sobre HDFS, sólo la esencial.

Verdadero Falso

Falso

Falso: ofrece mucha información, prácticamente toda la necesaria para monitorizar el servicio HDFS, aunque es verdad que no tiene un formato muy amigable.

3.- Interfaz de YARN: ResourceManager UI.

Caso práctico

María Robles, la responsable de IT del Banco Español de Inversiones, BEI, tiene el reto de administrar con su equipo la plataforma Hadoop implantada en el banco, que ya dispone de más de 20 usuarios entre científicos de datos y analistas de negocio.

Los usuarios lanzan diferentes trabajos en el clúster, desde consultas a Hive bastante pesadas, por ejemplo, para hacer cálculos sobre inversiones pasadas, o tareas Spark para implementar o ejecutar modelos predictivos.

Esta diversidad de usuarios está creando los primeros problemas, y es que a veces, el clúster se ralentiza mucho y ciertas tareas que son bastante importantes, como los reportes al consejo de administración, no se pueden terminar en la ventana de tiempo que deberían.

María y su equipo van a estudiar cómo monitorizar las aplicaciones que se están ejecutando, en primer lugar, para detectar los momentos en los que el clúster está saturado, y en segundo lugar, para poder averiguar qué tareas son las que están ralentizando la plataforma.

Van a empezar a utilizar interfaz que ofrece YARN por defecto, que se llama ResourceManager UI.



Gerd Altmann (Dominio público)

Al igual que en el caso de HDFS, YARN también ofrece un interfaz web de monitorización que se suele desplegar en el nodo donde se ejecuta el servicio ResourceManager. Esta web permite ver el estado de ejecución de las aplicaciones, el estado de recursos del sistema, o ver el detalle de las aplicaciones y los logs que están generando.

Al acceder a la pantalla principal, se muestra un resumen del sistema:

Iñigo Sanz (Dominio público)

En el menú de la izquierda se tienen diferentes opciones, por ejemplo, al pinchar en la opción Nodes se puede ver la información de los nodos worker, pudiendo comprobar cuántos hay, en qué estado se encuentran, la dirección interna del nodo dentro del clúster así como los recursos utilizados y disponibles.

<https://hadoopclusterexamplecidead.azurehdinsight.net/yarnui/hn/cluster/nodes>

Nodes of the cluster

Cluster Metrics																	
Nodes	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Nodes	Nodes	Nodes	Nodes	Nodes	Nodes
2 Cluster Nodes Metrics	0	0	2	0	0 B	192 GB	0 B	0	0	32	0	0	0	0	0	0	0
4 Scheduler Metrics	Active Nodes	Decommissioning Nodes	0	Decommissioned Nodes	0	Lost Nodes	0	Unhealthy Nodes	0	Rebooted Nodes	0	Shutdown Nodes					
Scheduler	Scheduler Type	Scheduling Resource Type			Minimum Allocation		Maximum Allocation			Maximum Cluster Application Priority							
Capacity Scheduler	[memory-mb (unit=MB), vcores]	<memory:3072, vcores:1>			<memory:49152, vcores:8>		0										
Show 20 - entries																	
Node Labels	Rack	Node State		Node Address		Node HTTP Address		Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	Vcores Used	Vcores Avail	Vcores Total	Vcores Reserved
/default-rack	RUNNING	wn0-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30050	wn0-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30060	Sat Jun 25 2022 +0000 2022	0	0 B	48 GB	0	0	0 B	48 GB	0	8	3.1.1.4.1.			
/default-rack	RUNNING	wn1-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30050	wn1-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30060	Sat Jun 25 2022 +0000 2022	0	0 B	48 GB	0	0	0 B	48 GB	0	8	3.1.1.4.1.			
/default-rack	RUNNING	wn2-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30050	wn2-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30060	Sat Jun 25 2022 +0000 2022	0	0 B	48 GB	0	0	0 B	48 GB	0	8	3.1.1.4.1.			
/default-rack	RUNNING	wn3-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30050	wn3-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30060	Sat Jun 25 2022 +0000 2022	0	0 B	48 GB	0	0	0 B	48 GB	0	8	3.1.1.4.1.			

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Íñigo Sanz (Dominio público)

En la opción Applications, se puede ver las aplicaciones que se han ejecutado o las que se están ejecutando en este momento. En este ejemplo se puede ver cómo existe una aplicación en ejecución, que se corresponde con una consulta Hive, y dos aplicaciones que se acaban de ejecutar.

<https://hadoopclusterexamplecidead.azurehdinsight.net/yarnui/hn/cluster/apps>

All Applications

Cluster Metrics																		
Nodes	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Nodes	Nodes	Nodes	Nodes	Nodes	Nodes	
3 Cluster Nodes Metrics	0	1	2	1	3 GB	192 GB	0 B	0	1	32	0	0	0	0	0	0	0	
4 Scheduler Metrics	Active Nodes	Decommissioning Nodes	0	Decommissioned Nodes	0	Lost Nodes	0	Unhealthy Nodes	0	Rebooted Nodes	0	Shutdown Nodes						
Scheduler	Scheduler Type	Scheduling Resource Type			Minimum Allocation		Maximum Allocation			Maximum Cluster Application Priority								
Capacity Scheduler	[memory-mb (unit=MB), vcores]	<memory:3072, vcores:1>			<memory:49152, vcores:8>		0											
Show 20 - entries																		
User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU	Allocated Memory MB	Reserved CPU	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking URL	BlockedNodes
application_1656137970815_0003	Nva	HIVE-4630da54-f1c0-43b1-accf-c5693835f5b	TEZ	default	0 Sat Jun 25 12:05:17 +0000 2022	N/A	RUNNING	UNDEFINED	1	1	3072	0	0	1.6	1.6	ApplicationMaster	0	
application_1656137970815_0002	Nva	HIVE-4630da54-f1c0-43b1-accf-c5693835f5b	TEZ	default	0 Sat Jun 25 10:26:50 +0000 2022	Sat Jun 25 10:26:50 +0000 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0	History	0	
application_1656137970815_0001	Nva	HIVE-4630da54-f1c0-43b1-accf-c5693835f5b	TEZ	default	0 Sat Jun 25 09:23:00 +0000 2022	Sat Jun 25 09:23:00 +0000 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0	History	0	

Showing 1 to 3 of 3 entries

First Previous 1 Next Last

Íñigo Sanz (Dominio público)

Haciendo clic sobre una aplicación, se puede consultar su detalle de ejecución. Por ejemplo, entrando en el detalle de la segunda aplicación ejecutada, se obtiene la siguiente información:

https://hadoopclusterexamplecidead.azurehdinsight.net/yarnui/hn/cluster/app/application_1656137970815_0002

Application application_1656137970815_0002

Application Overview																	
User	hive	Name	hive	Application Type	TEZ	Application Tags	userdriven,hive_20202625072243,2807d5e-2c2f-4580-aabc-4d795e723c88	Application Priority	0 (Higher Integer value indicates higher priority)	YarnApplicationState	FINISHED	Queued					
FinalStatus Reported by AM:	SUCCEEDED	Started:	Sat Jun 25 09:26:50 +0000 2022	Elapsed:	5mins, 44sec	Tracking URL:	History	Log Aggregation Status:	SUCCEEDED	Application Timeout (Reported by AM):	Uncertain	Diagnostics:	Session timed out, lastDAGCompletionTime=1656145629569 ms, sessionTimeoutInterval=300000 ms Session stats submittedDAGs=1, successfulDAGs=1, failedDAGs=0, killedDAGs=0	Total Resource Preempted:	<memory:0, vcores:0>		
Application Node Label expression:	<Not set>	AM container Node Label expression:	<DEFAULT_PARTITION>	Number of Non-AM Containers Preempted:	0	Total Number of AM Containers Preempted:	0	Resource Preempted from Current Attempt:	<memory:0, vcores:0>	Number of Non-AM Containers Preempted:	0	Aggregate Resource Allocation:	1125967 MB-seconds, 366 vcore-seconds	Aggregate Preempted Resource Allocation:	0 MB-seconds, 0 vcore-seconds		

Showing 20 - entries

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattempt_1656137970815_0002_000001	Sat Jun 25 10:26:50 +0000 2022	http://wn2-hadoop.byr3nnmfpihuppa3hnn0oxyeyfa.bx.internal.cloudapp.net:30060	Logs	0	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Íñigo Sanz (Dominio público)

Se puede ver que la aplicación necesitó 5 minutos y 44 segundos para ejecutarse, que lo lanzó el usuario "hive" (es el usuario con el que se lanzan las consultas de Hive por defecto). Asimismo, se puede comprobar que sólo necesitó ejecutarse en un nodo, el nodo wn2, y pinchando sobre el enlace "Logs" se puede consultar el log de ejecución por si fuera necesario analizar un fallo que hubiera podido ocurrir en dicha ejecución:

Log Type: dag_1656137970815_0002_1.dot
Log Upload Time: Sat Jun 25 08:32:36 +0000 2022
Log Length: 1485

```
graph TD; subgraph "from_hivesampletable_where_c_DESC_Stage_1_"; select1["select_from_hivesampletable_where_c_DESC_Stage_1_"]; end; subgraph "Map_1[MapReduceProcessor]"; map1["Map_1[MapReduceProcessor]"]; end; subgraph "Reduce_1[ReduceProcessor]"; reduce1["Reduce_1[ReduceProcessor]"]; end; subgraph "from_hivesampletable_where_c_DESC_Stage_1_Reducer_2_"; select2["select_from_hivesampletable_where_c_DESC_Stage_1_Reducer_2_"]; end; subgraph "Reduce_2[ReduceProcessor]"; reduce2["Reduce_2[ReduceProcessor]"]; end; subgraph "from_hivesampletable_where_c_DESC_Stage_1_Reducer_2_out_Reducer_2_"; select3["select_from_hivesampletable_where_c_DESC_Stage_1_Reducer_2_out_Reducer_2_"]; end; subgraph "Reduce_3[ReduceProcessor]"; reduce3["Reduce_3[ReduceProcessor]"]; end; subgraph "from_hivesampletable_where_c_DESC_Stage_1_Map_1_hivesampletable_2_"; select4["select_from_hivesampletable_where_c_DESC_Stage_1_Map_1_hivesampletable_2_"]; end; subgraph "Map_2[MapInputLegacy]"; map2["Map_2[MapInputLegacy]"]; end; subgraph "from_hivesampletable_where_c_DESC_Stage_1_Map_1_Reducer_2_out_Reducer_2_"; select5["select_from_hivesampletable_where_c_DESC_Stage_1_Map_1_Reducer_2_out_Reducer_2_"]; end; subgraph "Reduce_4[ReduceProcessor]"; reduce4["Reduce_4[ReduceProcessor]"]; end;
```

Log Type: directory.info
Log Upload Time: Sat Jun 25 08:32:36 +0000 2022
Log Length: 21061
Showing 4096 bytes of 21061 total. Click here for the full log.
hadoop 244918 May 15 18:12 ./texel/lib/commons-beanutils-1.9.4.jar
2883718 32 ->r--r--x 1 yarn hadoop 38674 May 18:14 /texel/lib/hadoop-conf-1.8.1.jar
2883719 16 -r--r--x 1 yarn hadoop 891164 May 18:14 /texel/lib/hadoop-yarn-service-1.8.1.jar
2883679 872 -r--r--x 1 yarn hadoop 891164 May 18:17 /texel/lib/hadoop-yarn-line-pluginstorage-3.1.1.4.1.9.7.jar
2883746 8 -r--r--x 1 yarn hadoop 4467 May 18:15 /texel/lib/hadoop-mapline-1.8.jar
2883747 20 -r--r--x 1 yarn hadoop 3997 May 18:15 /texel/lib/hadoop-mapline-1.8.1.jar
2883748 78 -r--r--x 1 yarn hadoop 76983 May 18:14 /texel/lib/guice-servlet-4.0.jar
2883774 132 -r--r--x 1 yarn hadoop 134380 May 18:11 /texel/lib/hadoop-jpc-jewel-2.1.0.jar
2883775 29 -r--r--x 1 yarn hadoop 165540 May 18:12 /texel/lib/jpc-jewel-2.1.0.1.jar
2883877 164 -r--r--x 1 yarn hadoop 165540 May 18:12 /texel/lib/jersey-1.9.jar
2883878 10 -r--r--x 1 yarn hadoop 95880 May 18:12 /texel/lib/jersey-client-3.1.1.4.1.9.7.jar
2883879 96 -r--r--x 1 yarn hadoop 95880 May 18:12 /texel/lib/jersey-servlet-3.1.1.4.1.9.7.jar
2883763 68 -r--r--x 1 yarn hadoop 65928 May 18:12 /texel/lib/jersey-1.8.jar
2883764 79 -r--r--x 1 yarn hadoop 75232 May 18:12 /texel/lib/jersey-client-common-3.1.1.4.1.9.7.jar
2883782 70 -r--r--x 1 yarn hadoop 75232 May 18:14 /texel/lib/jersey-2.5.1.jar
2883751 18 -r--r--x 1 yarn hadoop 15329 May 18:14 /texel/lib/jackson-jaxrs-annotation-2.18.0.jar
2883752 29 -r--r--x 1 yarn hadoop 15329 May 18:14 /texel/lib/jackson-jaxrs-annotation-2.18.0.1.jar
2883752 38 -r--r--x 1 yarn hadoop 33176 May 18:14 /texel/lib/jackson-jaxrs-base-2.18.0.jar
2883753 10 -r--r--x 1 yarn hadoop 33180 May 18:14 /texel/lib/jackson-jaxrs-base-2.18.0.1.jar
2883872 128 -r--r--x 1 yarn hadoop 33180 May 18:13 /texel/lib/jetty-annotations-9.4.24.v20180905.jar
2883747 16 -r--r--x 1 yarn hadoop 14151 May 18:13 /texel/lib/jersey-guice-1.19.jar
2883748 40 -r--r--x 1 yarn hadoop 436680 May 18:12 /texel/lib/jersey-guice-1.19.1.jar
2883748 428 -r--r--x 1 yarn hadoop 436680 May 18:12 /texel/lib/jersey-1.9.1.1.4.1.9.7.jar
2883776 1372 -r--r--x 1 yarn hadoop 1408944 May 18:19 /texel/lib/jackson-databind-2.18.0.jar
2883664 64 -r--r--x 1 yarn hadoop 58168 May 18:17 /texel/lib/commons-codec-1.4.jar
2883666 68 -r--r--x 1 yarn hadoop 58168 May 18:17 /texel/lib/commons-codec-1.4.jar
2883711 224 -r--r--x 1 yarn hadoop 226672 May 18:14 /texel/lib/jersey-client-3.1.1.4.1.9.7.jar
2883711 226 -r--r--x 1 yarn hadoop 226672 May 18:14 /texel/lib/jersey-client-3.1.1.4.1.9.7.jar
2883685 28 -r--r--x 1 yarn hadoop 27875 May 18:17 /texel/lib/jackson-ec-1.9.2.jar
2883740 2340 -r--r--x 1 yarn hadoop 27875 May 18:17 /texel/lib/jackson-ec-1.9.2.jar
2883784 816 -r--r--x 1 yarn hadoop 834918 May 18:17 /texel/lib/zookeeper-3.4.4.1.5.1.7.jar
2883785 4 -r--r--x 1 yarn hadoop 1465 May 15 18:09 /texel/LICENSE-BSD-3-Clause
2883807 10 -r--r--x 1 yarn hadoop 58168 May 18:17 /texel/lib/jersey-client-3.1.1.4.1.9.7.jar
2883886 4 -r--r--x 1 yarn hadoop 16 Jun 25 08:26 /default_container_executor_session.sh.crc
2883887 42996 -r--r--x 1 yarn hadoop 43182431 Jun 25 08:26 /hive-exec-3.1.2.4.1.9.7-98024dfe7a37d38090073aae199e0d8c1547ffed25e.jar
2883887 179 Jun 25 08:26 /container_tokens
broken symbols find -L -naxdepth 5 -type l -lsl

Log Type: launch_container.sh
Log Upload Time: Sat Jun 25 08:32:36 +0000 2022
Log Length: 1000
Showing 4096 bytes of 5420 total. Click here for the full log.
LOCAL_USER_ID=65 "/mnt/resource/hadoop/yarn/local/usercache/hive"/
"/mnt/resource/cfs/hadoop/yarn/log/applications/_1656137970815_0002_01_000001"
export USER="hive"
export LOGNAME="hive"

Íñigo Sanz (Dominio público)

El resto de opciones que aparecen dentro de la opción “Applications” sólo sirven para filtrar las aplicaciones que se muestran según su estado: en ejecución, finalizadas, paradas, etc.

En cuanto a la opción “Scheduler”, muestra el estado de las colas de ejecución que se han configurado en YARN, mostrando la capacidad de cada cola, su ocupación, etc. En el siguiente ejemplo, por ejemplo, se puede ver que hay dos colas configuradas, default y joblauncher:

NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
3	0	0	3	0	0 B	192 GB	0 B	0	32	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
4	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory_mb (uni=M), vcores]	<memory_3072, vCores_1>	<memory_49152, vCores_8>	0

Dump scheduler log: 1 min

Application Queues

Legend: Capacity **Used** Used (over capacity) Max Capacity Users Requesting Resources Auto Created Queues

- Queue: root
- + Queue: default
- + Queue: joblauncher

0.0% used
0.0% used
0.0% used

Show: 20 - entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																			

Showing 0 to 0 of 0 entries

Aggregate scheduler counts

Total Container Allocations(count)	Total Container Releases(count)	Total Fulfilled Reservations(count)	Total Container Preemptions(count)
8	8	0	0

Last scheduler run

Time	Allocations(count - resources)	Reservations(count - resources)	Releases(count - resources)
Sat Jun 25 10:14:02 +0000 2022	0 - <memory_0, vCores_0>	0 - <memory_0, vCores_0>	0 - <memory_0, vCores_0>

Last Preemption

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Last Reservation

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Last Allocation

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Last Release

Time	Container Id	Node Id	Queue
Sat Jun 25 10:11:03 +0000 2022	container_1656137970815_0003_01_000001	N/A	N/A

wn3-hadoop.bn3mrnf0hutaa3hnn0covevbx Internal cloudagent.net:30050

root.default...

Iñigo Sanz (Dominio público)

Haciendo clic sobre una cola, se puede ver su detalle, donde por ejemplo podemos comprobar que la cola joblauncher tiene una capacidad máxima del 50% de los recursos de YARN:

The screenshot shows the 'NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED' section of the YARN cluster metrics. It highlights the 'joblauncher' queue status, which includes the following details:

- Queue State:** RUNNING
- Used Capacity:** <memory:0, vCores:0> (0.0%)
- Configured Capacity:** <memory:0, vCores:0>
- Configured Max Capacity:** unlimited
- Effective Capacity:** <memory:9830, vCores:1> (5.0%)
- Effective Max Capacity:** <memory:98304, vCores:16> (50.0%)
- Absolute Used Capacity:** 0.0%
- Absolute Configured Capacity:** 5.0%
- Absolute Configured Max Capacity:** 50.0%
- Used Resources:** <memory:0, vCores:0>
- Configured Max Application Master Resources:** 33.0
- Max Application Master Resources:** <memory:33792, vCores:1>
- Used Application Master Resources:** <memory:0, vCores:0>
- Max Application Master Resources Per User:** <memory:33792, vCores:1>
- Num Schedulable Applications:** 0
- Num Non-Schedulable Applications:** 0
- Num Containers:** 0
- Max Applications:** 500
- Max Applications Per User:** 500
- Configured Minimum User Limit Percent:** 100%
- Configured User Limit Factor:** 10.0
- Accessible Node Labels:** *
- Ordering Policy:** FifoOrderingPolicy
- Preemption:** disabled
- Intra-queue Preemption:** disabled
- Default Node Label Expression:** <DEFAULT_PARTITION>
- Default Application Priority:** 0

Íñigo Sanz (Dominio público)

Por último, en la opción “Tools” disponemos de diferentes utilidades al igual que en el interfaz del Namenode:

- ✓ Ver la configuración de YARN.
- ✓ Ver los ficheros de log.
- ✓ Ver los hilos de ejecución en un momento dado para depurar errores.
- ✓ Ver las métricas actuales en formato JSON.

Autoevaluación

¿Cuál de las siguientes funcionalidades ofrece el ResourceManager UI?

- Ver el total de memoria y núcleos de proceso que YARN puede utilizar y su consumo actual
- Parar tareas que están consumiendo muchos recursos.
- Ver qué aplicaciones se están ejecutando y cuántos recursos está consumiendo cada una.
- Ver qué nodos worker hay en el clúster ejecutando YARN y en qué estado se encuentran.

[Mostrar retroalimentación](#)

Solución

1. Correcto
2. Incorrecto
3. Correcto
4. Correcto

4.- Apache Ambari.

Caso práctico

El equipo de IT del Banco Español de Inversiones, BEI, ya puede monitorizar HDFS y las tareas que se ejecutan en el clúster Hadoop y que son lanzadas por diferentes usuarios, tanto usuarios de negocio como aplicaciones o científicos de datos.

Tenían un problema por el que a veces el clúster tenía un rendimiento muy pobre, y gracias al interfaz de usuario de ResourceManager, con el que pudieron identificar la aplicación que consumía muchos recursos y que hacía que el resto de aplicaciones se ejecutaran de una forma muy lenta. Esta aplicación resultó ser unas consultas que un analista de negocio estaba lanzando con Hive, en las que unía múltiples tablas y que generaba consultas que tardaban varias horas en terminar. Pudieron detectarlo con ayuda de los logs que pudieron ver desde el interfaz del ResourceManager, y ayudaron a este analista a optimizar sus consultas.

Ahora bien, necesitan una herramienta que les permita unificar toda la monitorización y sobre todo, poder administrar el sistema, es decir, reiniciar servicios, parar o arrancar nodos, instalar componentes, cambiar la configuración, etc.

Saben que Apache Ambari es una herramienta específica para este propósito, y van a probarla.



Gerd Altmann (Dominio público)

Apache Ambari tiene como objetivo simplificar la administración de Hadoop para el aprovisionamiento, la administración y el monitoreo de clústeres. Ambari proporciona una interfaz de usuario web de administración de Hadoop intuitiva y fácil de usar respaldada por sus API RESTful.

Ambari permite a los administradores del sistema:



Apache Ambari

[Apache Software Foundation](#) (Apache License)

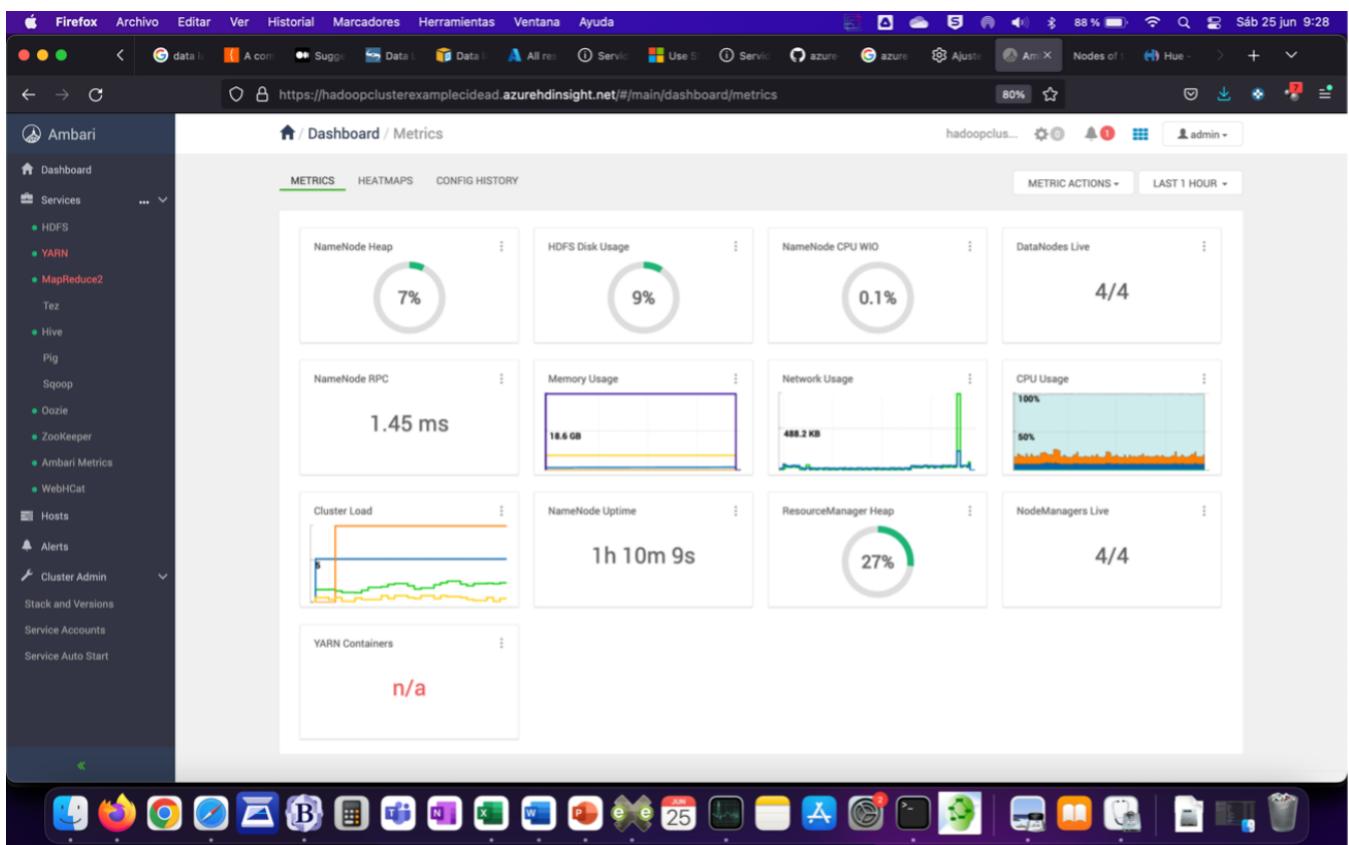
- ✓ **Instalar** un clúster de Hadoop:
 - ◆ Ambari proporciona un asistente paso a paso para instalar los servicios de Hadoop en cualquier número de hosts.
 - ◆ Ambari maneja la configuración de los servicios de Hadoop para el clúster.
- ✓ **Administrar** un clúster Hadoop, ya que proporciona funcionalidades para iniciar, detener y reconfigurar los servicios de Hadoop en todo el clúster, así como los diferentes nodos o servidores que lo componen.
- ✓ **Monitorizar** un clúster Hadoop, ya que:
 - ◆ Ambari proporciona cuadros de mando para monitorear la salud y el estado del clúster de Hadoop.
 - ◆ Ambari permite definir alertas y notificará cuando se cumpla la condición de la alerta (por ejemplo, un nodo deja de funcionar, el espacio restante en el disco es bajo, etc.).

Además, Ambari dispone de un API con el que las aplicaciones pueden utilizar los servicios indicados anteriormente, para poder ejecutar cualquier funcionalidad de administración de forma automática.

A continuación vamos a ver las principales pantallas y funcionalidades de Ambari para la administración y monitorización de plataformas Hadoop:

Pantalla inicial

Al acceder a Ambari, se obtiene un cuadro de mando general con las principales métricas.

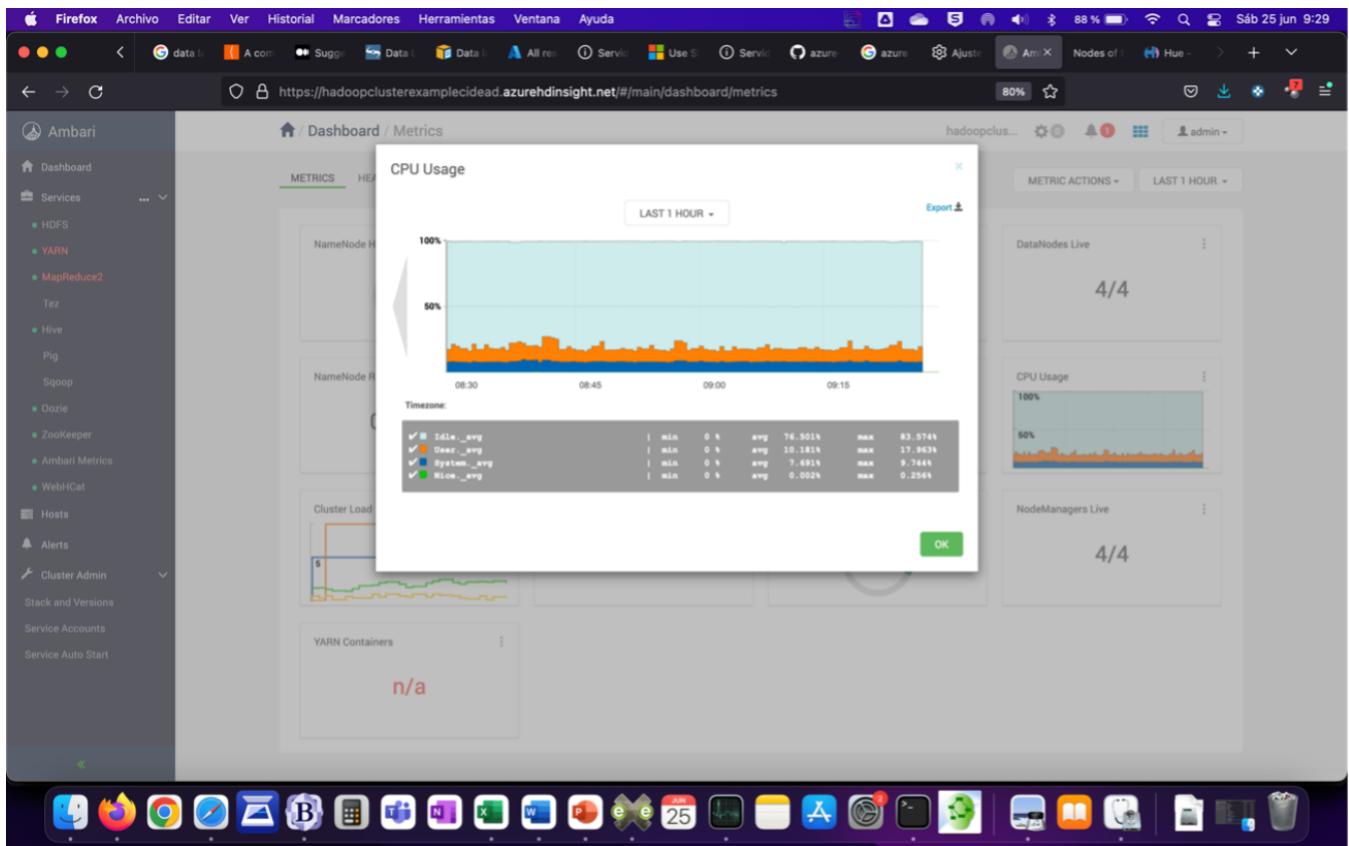


Iñigo Sanz (Dominio público)

Se puede ver algunas métricas importantes:

- ✓ El espacio usado por HDFS del total de espacio disponible, un 9%.
- ✓ El uso de CPU y memoria del Namenode, 0,1% y 7% respectivamente.
- ✓ El número de datanodes y cuántos están en servicio: 4.
- ✓ El uso de red.
- ✓ El uso de CPU general.
- ✓ El número de Nodemangers y cuántos están activos: 4.
- ✓ Algun dato adicional.

Se puede hacer zoom sobre una métrica para ver el detalle:

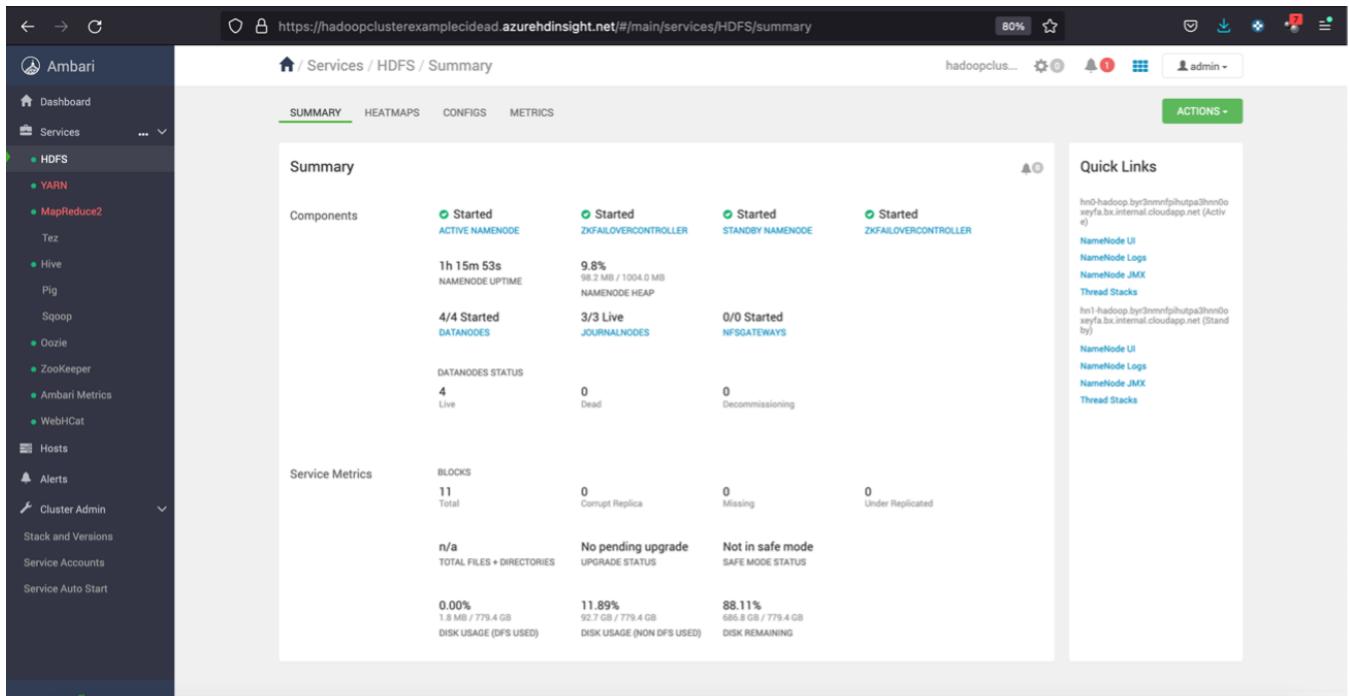


Íñigo Sanz (Dominio público)

En el menú de la izquierda se puede seleccionar qué tipo de acción realizar, se divide en servicios (HDFS, YARN, etc.), hosts (servidores del clúster), alertas, y otras acciones de administración.

Servicios

Entrando en servicios, podemos acceder a cada uno de los servicios o componentes del clúster para administrarlo (parar, arrancar, etc.), ver el estado o cambiar la configuración. Por ejemplo, accedemos a HDFS, y vemos en la primera pantalla un cuadro de mando con el resumen del estado de HDFS.



Íñigo Sanz (Dominio público)

Sobre las métricas que habíamos visto en el cuadro de mando principal, en esta pantalla se añaden algunas importantes como el número total de bloques (11), el estado de las réplicas, por si hubiera algún bloque corrupto o con un número de réplicas más bajos, etc.

En la pestaña de configuración, se puede modificar toda la configuración de HDFS de un modo visual, se puede ver un cuadro general o en la pestaña "Advanced" se puede ver toda la configuración disponible. Algunos parámetros de configuración son:

- ✓ Las rutas en el disco local del Namenode y Datanodes donde se enganchará HDFS.
- ✓ Tamaños de memoria de disponible para HDFS en los servidores.
- ✓ El factor de replicación por defecto (1 en nuestro ejemplo).
- ✓ Tiempo máximo de respuesta de los mensajes de status a partir del cual un nodo se marca como parado o defectuoso.

https://hadoopclusterexamplecidead.azurehdinsight.net/#/main/services/HDFS/configs

Ambari

Dashboard

Services

HDFS

YARN

MapReduce2

Tez

Hive

Pig

Sqoop

Oozie

ZooKeeper

Ambari Metrics

WebHCat

Hosts

Alerts

Cluster Admin

Stack and Versions

Service Accounts

Service Auto Start

NameNode

NameNode hosts: hn0-hadoop.byr3nmnfpihutpa3hnn0oxyfa.bx.internal.cloudapp.net and 1 other

NameNode new generation size: 200 MB

NameNode maximum new generation size: 200 MB

NameNode permanent generation size: 128 MB

NameNode maximum permanent generation size: 256 MB

DataNode

DataNode hosts: wn0-hadoop.byr3nmnfpihutpa3hnn0oxyfa.bx.internal.cloudapp.net and 3 others

DataNode directories permission: 750

General

DISCARD SAVE

Íñigo Sanz (Dominio público)

https://hadoopclusterexamplecidead.azurehdinsight.net/#/main/services/HDFS/configs

Ambari

Dashboard

Services

HDFS

YARN

MapReduce2

Tez

Hive

Pig

Sqoop

Oozie

ZooKeeper

Ambari Metrics

WebHCat

Hosts

Alerts

Cluster Admin

Stack and Versions

Service Accounts

Service Auto Start

General

WebHDFS enabled: Enabled

Hadoop maximum Java heap size: 1024 MB

Access time precision: 0

HDFS Maximum Checkpoint Delay: 21600 seconds

Reserved space for HDFS: 1073741824 bytes

Block replication: 1

NFS Gateway

NFSGateway host: No host assigned

NFSGateway maximum Java heap size: 1024 MB

NFSGateway dump directory: /tmp/.hdfs-nfs

Allowed hosts: * rw

Advanced core-site

Advanced hadoop-env

Advanced hadoop-metrics2.properties

DISCARD SAVE

Íñigo Sanz (Dominio público)

The screenshot shows the Ambari web interface for managing a Hadoop cluster. The left sidebar has a dark theme with white icons and text. It includes links for Dashboard, Services (with HDFS selected), YARN, MapReduce, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. Below these are Hosts, Alerts, Cluster Admin (with Stack and Versions, Service Accounts, and Service Auto Start), and a back arrow. The main content area is titled 'HDFS' and shows a list of configuration files with dropdown arrows: Advanced core-site, Advanced hadoop-env, Advanced hadoop-metrics2.properties, Advanced hadoop-policy, Advanced hdfs-log4j, Advanced hdfs-site, Advanced ranger-hdfs-audit, Advanced ranger-hdfs-plugin-properties, Advanced ranger-hdfs-policymgr-ssl, Advanced ranger-hdfs-security, Advanced ssl-client, Advanced ssl-server, Advanced viewfs-mount-table, Custom core-site, Custom hadoop-metrics2.properties, and Custom hadoop-policy. At the bottom right are 'DISCARD' and 'SAVE' buttons. The URL in the address bar is https://hadoopclusterexamplecidead.azurehdinsight.net/#/main/services/HDFS/configs.

Íñigo Sanz (Dominio público)

Hay infinidad de variables de configuración. Antes de modificar alguna es conveniente consultar la guía oficial para entender qué implicaciones tiene o para qué sirve el parámetro. En caso de modificar algún parámetro, se requerirá reiniciar los servicios.

En la última pestaña, “Metrics” se puede configurar un cuadro de mando a medida con las métricas que nos resulten interesantes, aunque las que aparecen cargadas por defecto suelen ser las habituales para poder entender la salud de HDFS.

Por otro lado, en la parte superior derecha, se puede ver un botón “Actions” con el que podemos manejar el servicio para parar el servicio, reiniciarlo, etc.

Por último, en la parte derecha aparecen los enlaces a las consolas de administración generales del Namenode, así como a los logs.

Accediendo al servicio YARN obtenemos la misma estructura de pestañas de HDFS, con métricas y configuraciones específicas de YARN, así como las acciones para parar el servicio, reiniciarlo, etc. Por ejemplo, el cuadro de mando general muestra información sobre el estado del ResourceManager y los Nodemangers, y en la parte inferior indica cuántas aplicación hay ejecutándose en este momento en YARN (1) o cuántos contenedores se están utilizando (2).

The screenshot shows the Ambari interface for the YARN service. The left sidebar lists various services like HDFS, YARN, MapReduce2, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. The main content area has tabs for SUMMARY, HEATMAPS, CONFIGS, and METRICS. The SUMMARY tab displays the following data:

Component	Status	Count
Timeline Service V1.5	Started	1
Timeline Service V1.5	Stopped	1
Standby ResourceManager	Started	1
Active ResourceManager	Started	1
NodeManagers	4/4 Started	4
YARN Clients	2 Installed	2

RESOURCEMANAGER UPTIME: 1h 27m 7s

NODEMANAGERS STATUS:

Status	Count
Active	4
Lost	0
Unhealthy	0
Rebooted	0
Decommissioned	0

RESOURCEMANAGER HEAP: 113.8 MB / 910.5 MB

Service Metrics:

Category	Count
CONTAINERS	2 Allocated
APPLICATIONS	1 Submitted

YARN Features (Enabled): Node Labels, Pre-emption, Docker Runtime.

Íñigo Sanz (Dominio público)

En cuanto a la configuración, el formato de presentación es similar, con una pantalla simplificada denominada "Settings" así como una pantalla con todos los parámetros de configuración, denominada "Advanced".

The screenshot shows the Ambari interface for the YARN service configurations. The left sidebar lists various services. The main content area has tabs for SUMMARY, HEATMAPS, CONFIGS, and METRICS. The CONFIGS tab displays the following configuration sections:

SETTINGS (selected) and **ADVANCED**

Memory (Node): Minimum Container Size (Memory) is set to 256MB. Maximum Container Size (Memory) is set to 49152 MB.

CPU (Node): CPU Scheduling and Isolation is set to Disabled.

YARN Features (Enabled): Node Labels, Pre-emption, Docker Runtime.

Buttons at the bottom right: DISCARD and SAVE.

Íñigo Sanz (Dominio público)

Ambari

Dashboard Services ...

HDFS YARN MapReduce2 Tez Hive Pig Sqoop Oozie ZooKeeper Ambari Metrics WebHCat Hosts Alerts Cluster Admin Stack and Versions Service Accounts Service Auto Start

CPU

Node CPU Scheduling and Isolation: Disabled

Container Minimum Container Size (VCores): 1

Percentage of physical CPU allocated for all containers on a node: 80%

Maximum Container Size (VCores): 8

Number of virtual cores: 2

GPU

GPU Scheduling and Isolation: Disabled

Container Maximum Container Size (GPU): 1

Absolute path of nvidia-smi on NodeManagers: [redacted]

DISCARD SAVE

This screenshot shows the Ambari interface for managing YARN configurations. On the left, there's a sidebar with various service links like HDFS, YARN, and MapReduce2. The main area is titled 'CPU' and 'GPU'. Under 'CPU', it shows 'Node CPU Scheduling and Isolation' set to 'Disabled', and three sliders for 'Minimum Container Size (VCores)', 'Percentage of physical CPU allocated for all containers on a node', and 'Maximum Container Size (VCores)'. Under 'GPU', it shows 'GPU Scheduling and Isolation' set to 'Disabled', and a slider for 'Container Maximum Container Size (GPU)'.

Íñigo Sanz (Dominio público)

Ambari

Dashboard Services / YARN / Configs

SUMMARY HEATMAPS CONFIGS METRICS ACTIONS

Version: 2

SETTINGS ADVANCED

Resource Manager

ResourceManager hosts: hn0-hadoop.byr3nmnfpihutpa3hnn0xseyfa.bx.internal.cloudapp.net and 1 other

ResourceManager Java heap size: 1024 MB

yarn.acl.enable: false

yarn.admin.acl: [redacted]

Enable Log Aggregation: [checkboxes]

Node Manager

NodeManager hosts: wn0-hadoop.byr3nmnfpihutpa3hnn0xseyfa.bx.internal.cloudapp.net and 3 others

NodeManager Java heap size: 1024 MB

yarn.nodemanager.aux-services: mapreduce_shuffle,spark2_shuffle,{{timeline_collector}}

DISCARD SAVE

This screenshot shows the Ambari interface for managing YARN configurations, specifically under the 'CONFIGS' tab. It displays two sections: 'Resource Manager' and 'Node Manager'. In 'Resource Manager', it lists 'ResourceManager hosts' (hn0-hadoop.byr3nmnfpihutpa3hnn0xseyfa.bx.internal.cloudapp.net and 1 other), sets 'ResourceManager Java heap size' to 1024 MB, and shows configuration for 'yarn.acl.enable' (set to false) and 'yarn.admin.acl' (with a redacted value). In 'Node Manager', it lists 'NodeManager hosts' (wn0-hadoop.byr3nmnfpihutpa3hnn0xseyfa.bx.internal.cloudapp.net and 3 others), sets 'NodeManager Java heap size' to 1024 MB, and shows configuration for 'yarn.nodemanager.aux-services' (set to mapreduce_shuffle,spark2_shuffle,{{timeline_collector}}).

Íñigo Sanz (Dominio público)

The screenshot shows the Ambari interface for managing YARN configurations. On the left, there's a sidebar with various service options like HDFS, YARN, MapReduce2, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. The main area is divided into two sections: Resource Manager and Node Manager. In the Resource Manager section, there are fields for ResourceManager hosts (hn0-hadoop.byr3nmnfpihutpa3hnn0oxyfa.bx.internal.cloudapp.net and 1 other), ResourceManager Java heap size (1024 MB), yarn.acl.enable (false), and yarn.admin.acl. In the Node Manager section, there are fields for NodeManager hosts (hn0-hadoop.byr3nmnfpihutpa3hnn0oxyfa.bx.internal.cloudapp.net and 3 others), NodeManager Java heap size (1024 MB), yarn.nodemanager.aux-services (mapreduce_shuffle,spark2_shuffle,{{timeline_collector}}), YARN NodeManager Local directories (/mnt/resource/hadoop/yarn/local), and YARN NodeManager Log directories (/mnt/resource/hadoop/yarn/log). At the bottom right, there are 'DISCARD' and 'SAVE' buttons.

Íñigo Sanz (Dominio público)

Algunos parámetros de configuración importantes son:

- ✓ El tamaño máximo de los contenedores: 49152 megabytes y 1 vCore en el ejemplo.
- ✓ Los recursos máximos que puede tomar YARN de los distintos nodos worker: 80% de la CPU como máximo y 48 gigabytes de RAM.
- ✓ La ruta de los ficheros de log en los Nodemanager.

Al igual que en el caso de HDFS, se dispone de una última pestaña para montar cuadros de mando a medida con las métricas que más nos interese monitorizar, aunque vienen cargadas las más importantes:

The screenshot shows the Ambari Metrics page for the YARN service. The left sidebar is identical to the previous configuration page. The main area has tabs for SUMMARY, HEATMAPS, CONFIGS, and METRICS, with METRICS selected. It displays several performance metrics in a grid: Memory Utilization (4%, 2%), CPU Utilization (5%, 2%), Bad Local Disks (n/a), Container Failures (1%, 0.5%), App Failures (1%, 0.5%), Pending Apps (1 Apps, 0.5 Apps), Cluster Memory (4%, 2%), Cluster Disk (200 Mbps, 100 Mbps), Cluster Network (1K), and Cluster CPU (10%, 5%). There are 'ACTIONS' and 'LAST 1 HOUR' buttons at the top right. At the bottom, it says 'Licensed under the Apache License, Version 2.0. See third-party tools/resources that Ambari uses and their respective authors.'

Íñigo Sanz (Dominio público)

Por último, como ocurría en HDFS, tenemos opciones para arrancar, parar o reiniciar YARN, por ejemplo, tras aplicar algún cambio de configuración, así como los enlaces a las consolas de monitorización por defecto de YARN.

The screenshot shows the Ambari interface for the YARN service. On the left sidebar, 'YARN' is selected. The main content area displays the 'SUMMARY' tab for YARN. It shows the following components and their states:

- Components**: Started (TIMELINE SERVICE V1.5), Stopped (TIMELINE SERVICE V1.5), Started (STANDBY RESOURCEMANAGER), Started (ACTIVE RESOURCEMANAGER).
- 4/4 Started NODEMANAGERS**
- 2 Installed YARN CLIENTS**
- 1h 36m 12s RESOURCEMANAGER UPTIME**
- NODEMANAGERS STATUS**: 4 Active, 0 Lost, 0 Unhealthy, 0 Rebooted, 0 Decommissioned.
- 228.6 MB / 910.5 MB RESOURCEMANAGER HEAP**
- Service Metrics**: CONTAINERS (0 Allocated, 0 Pending, 0 Reserved), APPLICATIONS (1 Submitted, 0 Running, 0 Pending, 1 Completed).

A right-hand sidebar titled 'ACTIONS' includes options like Start, Stop, Refresh YARN Capacity Scheduler, and Restart All. Below it are links for ResourceManager UI, ResourceManager logs, ResourceManager JMX, and Thread Stacks.

Íñigo Sanz (Dominio público)

El resto de servicios, como Hive, Tez, Pig, Sqoop, etc. tienen interfaces más sencillas, con menos elementos de monitorización, ya que sus tareas se ejecutan en YARN, y por lo tanto, se monitorizarán desde esa consola. Disponen, además, de toda la configuración para poder realizar modificaciones, así como acciones para parar, arrancar o reiniciar los servicios.

Por ejemplo, en el caso de Hive, nos encontramos este cuadro de mando general:

The screenshot shows the Ambari interface for the Hive service. On the left sidebar, 'Hive' is selected. The main content area displays the 'SUMMARY' tab for Hive. It shows the following components and their states:

- Components**: Started (HIVE METASTORE), Started (HIVE METASTORE), Started (HIVESERVER2), Started (HIVESERVER2).
- 6 Installed HIVE CLIENTS**
- HIVESERVER2 JDBC URL**: jdbc:hive2://zk1-hadoop.byr3nnmfphutpa3hnn0oxyefabx.internal.cloudapp.net:2181,zk2-hadoop.byr3nnmfphutpa3hnn0oxyefabx.internal.cloudapp.net:2181,zk5-hadoop.byr3nnmfphutpa3hnn0oxyefabx.internal.cloudapp.net:2181/,serviceDiscoveryMode=zooKeeper,zooKeeperNamespace=hiveserver2

A right-hand sidebar titled 'Quick Links' includes a link to 'Hive Dashboard (Grafana)'.

Íñigo Sanz (Dominio público)

Como puedes ver, indica el estado del servicio, pero no muestra datos de rendimiento, ya que sus operaciones, como hemos indicado, se realizan en YARN. En cuanto a la configuración, se dispone del mismo tipo de pantalla, con una ficha simplificada y una pestaña avanzada con toda la configuración disponible:

The screenshot shows the Ambari interface for managing Hive configurations. The left sidebar lists services like HDFS, YARN, MapReduce2, Tez, and Hive. The main area is titled 'Services / Hive / Configs' and shows the 'CONFIGS' tab selected. It includes sections for 'Interactive Query', 'Security', and 'Optimization'. In the 'Interactive Query' section, there's a toggle for 'Enable Interactive Query (requires YARN pre-emption)' set to 'No'. The 'Security' section shows 'HiveServer2 Authentication' set to 'None' and 'Use SSL' set to 'False'. The 'Optimization' section has a 'Tez Container Size' slider set to 4096 MB. At the bottom are 'DISCARD' and 'SAVE' buttons.

Íñigo Sanz (Dominio público)

This screenshot shows the same Ambari interface for Hive configurations, but with different sections visible. It includes the 'Hive Metastore' and 'General' sections. Under 'Hive Metastore', it lists 'Hive Metastore hosts' as 'hn0-hadoop.byr3nnnfpihutpa3hnn0seyfa.bx.internal.cloudapp.net and 1 other'. The 'General' section contains various configuration parameters like 'datanucleus.cache.level2.type' (set to 'none'), 'hive.compactor.check.interval' (set to '300'), 'hive.compactor.delta.num.threshold' (set to '10'), and 'hive.compactor.delta.pct.threshold' (set to '0.1f'). At the bottom are 'DISCARD' and 'SAVE' buttons.

Íñigo Sanz (Dominio público)

Hosts

En cuanto a la opción de Hosts, muestra el estado de los distintos servidores. Es decir, no se tiene una visión de servicio, que puede estar ejecutándose en múltiples servidores, sino una visión de los servidores uno a uno.

Al acceder a la opción, se muestra todos los servidores del clúster:

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
hn0-hadoop.by3nmnfp...	10.0.0.21	/default-rack	4 (4)	31.36GB	<div style="width: 10%;"> </div>	8.07	HDInsight-4.1.9.7	20 Components
hn1-hadoop.by3nmnfp...	10.0.0.22	/default-rack	4 (4)	31.36GB	<div style="width: 10%;"> </div>	2.17	HDInsight-4.1.9.7	16 Components
wn0-hadoop.by3nmnfp...	10.0.0.12	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	1.00	HDInsight-4.1.9.7	7 Components
wn1-hadoop.by3nmnfp...	10.0.0.14	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.42	HDInsight-4.1.9.7	7 Components
wn2-hadoop.by3nmnfp...	10.0.0.13	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.47	HDInsight-4.1.9.7	7 Components
wn3-hadoop.by3nmnfp...	10.0.0.11	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.12	HDInsight-4.1.9.7	7 Components
zk1-hadoop.by3nmnfp...	10.0.0.5	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	1.09	HDInsight-4.1.9.7	4 Components
zk2-hadoop.by3nmnfp...	10.0.0.6	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	0.95	HDInsight-4.1.9.7	4 Components
zk5-hadoop.by3nmnfp...	10.0.0.10	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	0.91	HDInsight-4.1.9.7	4 Components

Íñigo Sanz (Dominio público)

Se puede ver que nuestro clúster tiene 9 servidores. Para cada servidor se puede ver la dirección IP interna, la memoria disponible, el uso de disco, la media de CPU utilizada, así como el número de servicios que está ejecutando. En caso de existir alguna incidencia, además, se marcará con un ícono como puede verse en el segundo servidor, que indica que está habiendo una incidencia con el servicio Timeline Server, de YARN.

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
hn0-hadoop.by3nmnfp...	10.0.0.21	/default-rack	4 (4)	31.36GB	<div style="width: 10%;"> </div>	9.40	HDInsight-4.1.9.7	20 Components
hn1-hadoop.by3nmnfp...	10.0.0.22	/default-rack	4 (4)	31.36GB	<div style="width: 10%;"> </div>	4.83	HDInsight-4.1.9.7	16 Components
wn0-hadoop.by3nmnfp...	10.0.0.12	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.47	HDInsight-4.1.9.7	7 Components
wn1-hadoop.by3nmnfp...	10.0.0.14	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.44	HDInsight-4.1.9.7	7 Components
wn2-hadoop.by3nmnfp...	10.0.0.13	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.31	HDInsight-4.1.9.7	7 Components
wn3-hadoop.by3nmnfp...	10.0.0.11	/default-rack	8 (8)	62.81GB	<div style="width: 10%;"> </div>	0.31	HDInsight-4.1.9.7	7 Components
zk1-hadoop.by3nmnfp...	10.0.0.5	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	0.68	HDInsight-4.1.9.7	4 Components
zk2-hadoop.by3nmnfp...	10.0.0.6	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	0.64	HDInsight-4.1.9.7	4 Components
zk5-hadoop.by3nmnfp...	10.0.0.10	/default-rack	2 (2)	3.84GB	<div style="width: 10%;"> </div>	0.98	HDInsight-4.1.9.7	4 Components

Íñigo Sanz (Dominio público)

Haciendo clic sobre el nombre de un servidor, podemos conocer el detalle de los servicios que está ejecutando, así como su estado. Por ejemplo, al entrar en el primer servidor, que es un nodo maestro de HDFS, que también hace de frontera, con los interfaces de todos los servicios instalados, nos muestra la siguiente información:

The screenshot shows the Ambari interface for a Hadoop cluster. On the left, a sidebar lists services like HDFS, YARN, and MapReduce2. The main area displays a table of components with columns for Status, Name, Type, and Action. Several components are marked as Master, including Timeline Service V1.5 / YARN, History Server / MapReduce2, Hive Metastore / Hive, HiveServer2 / Hive, Metrics Collector / Ambari Metrics, Grafana / Ambari Metrics, Active NameNode / HDFS - mycluster, Oozie Server / Oozie, Standby ResourceManager / YARN, WebHCat Server / WebHCat, Metrics Monitor / Ambari Metrics, ZKFailoverController / HDFS, HDFS Client / HDFS, Hive Client / Hive, MapReduce2 Client / MapReduce2, Oozie Client / Oozie, and Pig Client / Pig. To the right, there's a section titled 'Host Metrics' with five charts: CPU Usage, Disk Usage, Load, Memory Usage, Network Usage, and Processes. A 'HOST ACTIONS' dropdown is visible at the top right.

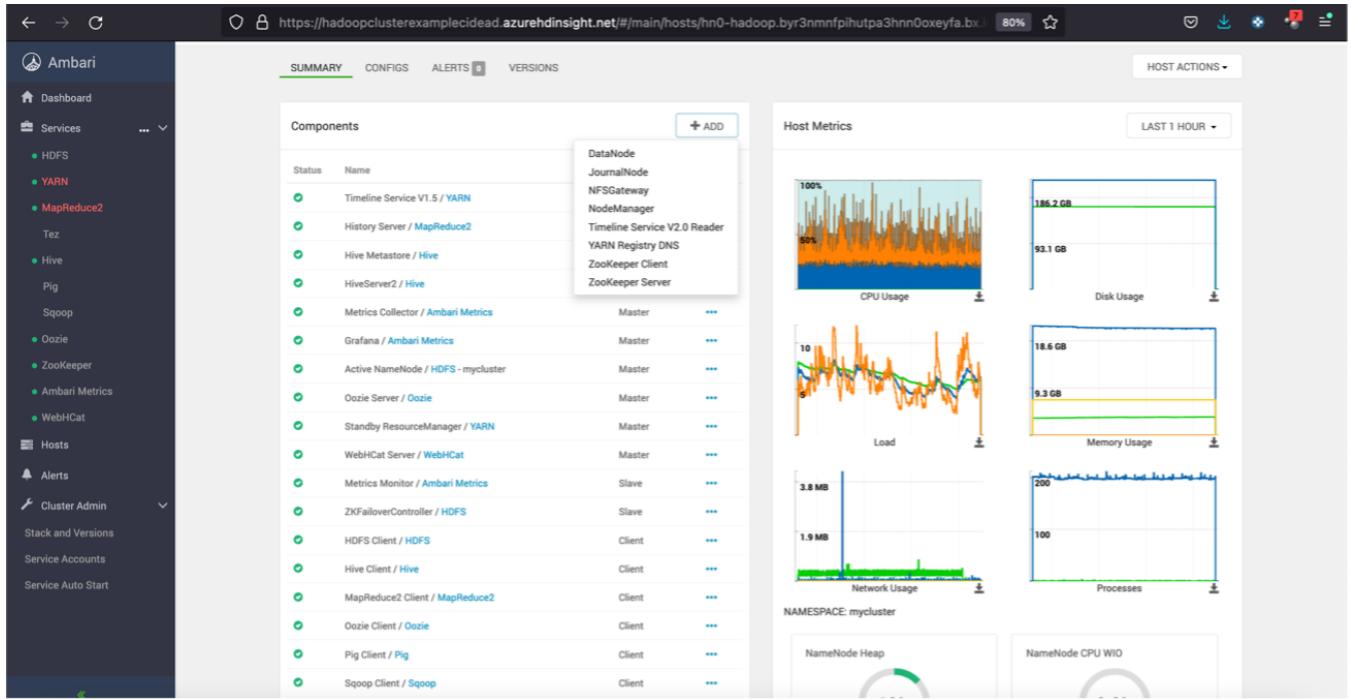
Íñigo Sanz (Dominio público)

Para cada servicio, pinchando en las opciones que aparecen en la parte derecha, se puede parar, reiniciar o mover el servicio a otro servidor.

This screenshot is similar to the previous one, showing the Ambari Summary page. However, a context menu is open over the 'Timeline Service V1.5 / YARN' entry in the component list. The menu options include 'Restart', 'Stop', 'Move', and 'Turn On Maintenance Mode'. The rest of the interface and metrics section are identical to the first screenshot.

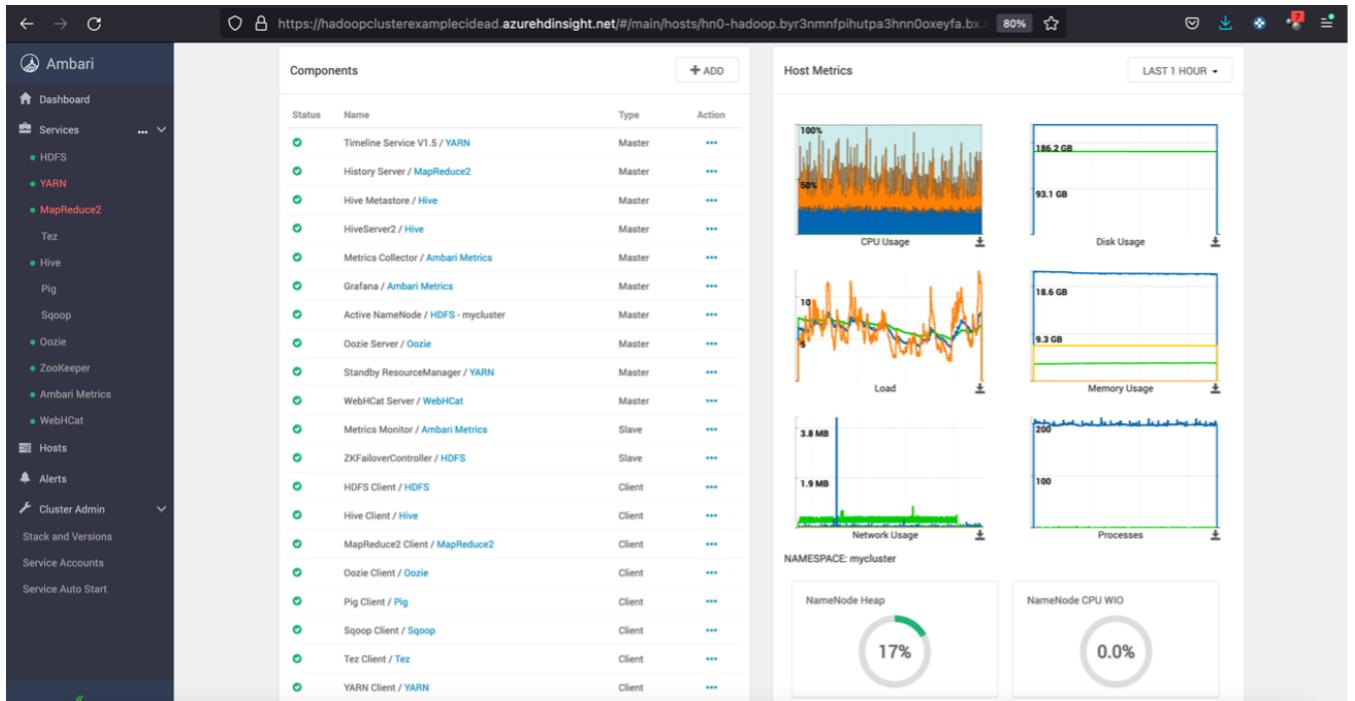
Íñigo Sanz (Dominio público)

Además, se puede añadir un servicio al servidor, por ejemplo, si quisieramos que este nodo maestro además fuera un Datanode (no recomendable):



Íñigo Sanz (Dominio público)

En la parte derecha podemos ver diferentes métricas del estado del servidor, donde puede apreciarse que no tiene grandes problemas en cuanto al uso de recursos:



Íñigo Sanz (Dominio público)

En la siguiente pestaña, de configuración, se puede visualizar y modificar la configuración de los servicios de este servidor:

The screenshot shows the Ambari interface for managing a Hadoop cluster. The left sidebar lists services such as HDFS, YARN, MapReduce2, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. The main content area is titled 'Hosts / hn0-hadoop.byr3nmnfpihupta3hnn0oxeyfa.bx... / Configs'. It displays configuration settings for 'HDFS' under the 'CONFIGS' tab. The 'SETTINGS' tab is selected for 'NameNode', showing parameters like 'NameNode hosts' (hn0-hadoop.byr3nmnfpihupta3hnn0oxeyfa.bx.internal.cloudapp.net and 1 other), 'NameNode new generation size' (200 MB), 'NameNode maximum new generation size' (200 MB), 'NameNode permanent generation size' (128 MB), and 'NameNode maximum permanent generation size' (256 MB). The 'ADVANCED' tab is also visible. Below this, the 'DataNode' section shows 'DataNode hosts' (hn0-hadoop.byr3nmnfpihupta3hnn0oxeyfa.bx.internal.cloudapp.net and 3 others).

Íñigo Sanz (Dominio público)

En la siguiente pestaña, de alertas, se puede ver si ha habido algún error significativo, o si por el contrario el servidor está funcionando correctamente.

The screenshot shows the Ambari interface for managing a Hadoop cluster. The left sidebar lists services such as HDFS, YARN, MapReduce2, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. The main content area is titled 'Hosts / hn0-hadoop.byr3nmnfpihupta3hnn0oxeyfa.bx... / Alerts'. It displays a list of alerts for various services. The table includes columns for 'Service', 'Alert Definition Name', 'Status', and 'Response'. Most entries show 'OK' status for over 2 hours. Some notes include 'Capacity Used: [33.76%, 44.9 GB], Capacity Total: [133.0 GB], path=/usr/h...', 'hdp-select reported the following versions: 4.1.9.7', 'TCP OK - 0.000s response on port 6188', 'Metrics Collector has not been auto-started and is running normally.', 'HTTP 200 response in 0.000s', 'Ambari Monitor is running on hn0-hadoop.byr3nmnfpihupta3hnn0oxeyfa...', 'TCP OK - 0.000s response on port 61310', '4 CPU, load 49.7%', and 'HTTP 200 response in 0.000s'. The bottom right shows 'Items per page: 10 - 1 - 10 of 36'.

Íñigo Sanz (Dominio público)

En el caso de que accedamos a un nodo worker, muestra menos servicios, como es lógico, porque no tiene interfaces o servicios maestro instalados, pero el resto de estructura de pestañas es similar:

Host Metrics

- CPU Usage: 100% (50% usage)
- Disk Usage: 279.3 GB (186.2 GB free)
- Load: 1.5 (fluctuating between 1 and 2)
- Memory Usage: 55.8 GB (37.2 GB free)
- Network Usage: 7.6 MB (1.9 MB average)
- Processes: 150 (50 active)

Íñigo Sanz (Dominio público)

Alertas

En la opción de alertas, muestra un listado de todas las comprobaciones periódicas, indicando si hay algún fallo en alguna de ellas:

Status	Alert Definition Name	Service	Last Status Changed	State
OK (2)	NameNode Client RPC Processing Latency (Hourly)	HDFS	2 hours ago	Enabled
OK (2)	ZooKeeper Failover Controller Process	HDFS	2 hours ago	Enabled
OK (2)	NameNode Directory Status	HDFS	2 hours ago	Enabled
OK (2)	HDFS Capacity Utilization	HDFS	2 hours ago	Enabled
OK (2)	NameNode RPC Latency	HDFS	2 hours ago	Enabled
OK (2)	NameNode Last Checkpoint	HDFS	2 hours ago	Enabled
OK (2)	ResourceManager Web UI	YARN	2 hours ago	Enabled
OK (2)	ResourceManager CPU Utilization	YARN	2 hours ago	Enabled
OK (2)	ResourceManager RPC Latency	YARN	2 hours ago	Enabled
OK (2)	NodeManager Health Summary	YARN	2 hours ago	Enabled

Íñigo Sanz (Dominio público)

Administración del clúster

En la opción de Administración del clúster, se puede gestionar aspectos como los componentes y versiones de los componentes utilizados, los usuarios de sistema operativos de los diferentes servicios, u opciones para arranque automático de los servicios.

Pantalla de componentes y versiones:

The screenshot shows the Ambari interface for managing a Hadoop cluster. The left sidebar is dark blue with white text, showing navigation links like Dashboard, Services, Hosts, Alerts, Cluster Admin, and Service Accounts. The 'Cluster Admin' section is expanded, showing 'Stack and Versions' which is currently selected. The main content area has a light gray background and displays a table titled 'Stack'. The table has columns for Service, Version, Status, and Description. All services listed are in 'Installed' status. The table includes entries for HDFS, YARN, MapReduce2, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. Below the table, there is a note from 'Íñigo Sanz (Dominio público)'.

Service	Version	Status	Description
HDFS	3.1.1	Installed	Apache Hadoop Distributed File System
YARN	3.1.1	Installed	Apache Hadoop NextGen MapReduce (YARN)
MapReduce2	3.1.1	Installed	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.9.1	Installed	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	3.1.0	Installed	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
Pig	0.16.0	Installed	Scripting platform for analyzing large datasets
Sqoop	1.5.0	Installed	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.3.1	Installed	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJS library.
ZooKeeper	3.4.6	Installed	Centralized service which provides highly reliable distributed coordination
Infra Solr	0.1.0	Installed	Core shared service used by Ambari managed components.
Ambari Metrics	0.1.0	Installed	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
Spark2	2.4.4	Installed	Apache Spark 2.3 is a fast and general engine for large-scale data processing.
Zeppelin Notebook	0.8.0	Installed	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
Credential Service	0.1	Installed	Credential Service 0.1
Delegation Token Service	0.1	Installed	Delegation Token 0.1
HDInsight Data Disks	1.0.0.0	Installed	Azure HDInsight Data Disks
HDInsight Kafka Tools	1.0.0.0	Installed	Azure HDInsight Kafka tools

Íñigo Sanz (Dominio público)

Pantalla de usuarios de sistema operativo de los servicios:

The screenshot shows the Ambari interface for managing service accounts. The left sidebar is identical to the previous one. The 'Service Accounts' section is currently selected. The main content area displays a table titled 'Service Users and Groups'. The table has columns for Name and Value. It lists various users and groups, such as Ambari Metrics User (ams), Smoke User (ambari-qa), Hadoop Group (hadoop), HDFS User (hdfs), Proxy User Group (users), Hive User (hive), Mapreduce User (mapred), Oozie User (oozie), Sqoop User (sqoop), Tez User (tez), WebHCat User (hive), Yarn ATS User (yarn-ats), Yarn User (yarn), and ZooKeeper User (zookeeper). Below the table, there is a note from 'Íñigo Sanz (Dominio público)'.

Name	Value
Ambari Metrics User	ams
Smoke User	ambari-qa
Hadoop Group	hadoop
HDFS User	hdfs
Proxy User Group	users
Hive User	hive
Mapreduce User	mapred
Oozie User	oozie
Sqoop User	sqoop
Tez User	tez
WebHCat User	hive
Yarn ATS User	yarn-ats
Yarn User	yarn
ZooKeeper User	zookeeper

Íñigo Sanz (Dominio público)

Inicio automático de los servicios:

Service	Components	Auto Start
HDFS	Metrics Collector	<input checked="" type="checkbox"/>
	Grafana	<input checked="" type="checkbox"/>
	Metrics Monitor	<input checked="" type="checkbox"/>
	DataNode	<input checked="" type="checkbox"/>
	HDFS Client	<input checked="" type="checkbox"/>
Hive	JournalNode	<input checked="" type="checkbox"/>
	NameNode	<input checked="" type="checkbox"/>
	ZKFailoverController	<input checked="" type="checkbox"/>
MapReduce2	Hive Client	<input checked="" type="checkbox"/>
	Hive Metastore	<input checked="" type="checkbox"/>
	HiveServer2	<input checked="" type="checkbox"/>
Oozie	History Server	<input checked="" type="checkbox"/>
	MapReduce2 Client	<input checked="" type="checkbox"/>
Oozie	Oozie Client	<input checked="" type="checkbox"/>
	Oozie Server	<input checked="" type="checkbox"/>

Íñigo Sanz (Dominio público)

Autoevaluación

Indica si las siguientes afirmaciones son verdaderas o falsas.

Ambari permite simplificar la instalación de un clúster Hadoop, ofreciendo una funcionalidad para hacer la instalación a modo de asistente

Verdadero Falso

Verdadero

Verdadero: Ambari dispone de un asistente para ayudar en la instalación y configuración inicial de un clúster.

Ambari me permite definir alertas para recibir notificaciones en caso de que se caiga un servidor del clúster.

Verdadero Falso

Verdadero

Verdadero: Ambari tiene una funcionalidad para definir alertas basada en reglas.

Con Ambari se puede saber el porcentaje de CPU utilizado en los servidores, el porcentaje de memoria o de disco.

Verdadero Falso

Verdadero

Verdadero: Ambari ofrece muchas métricas, algunas de las cuales son las que se indican en la pregunta.



4.1.- Ejemplo: parada de nodos worker

Como sabes, Hadoop es capaz de sobreponerse a la parada o caída de los nodos worker, dando servicio igualmente:

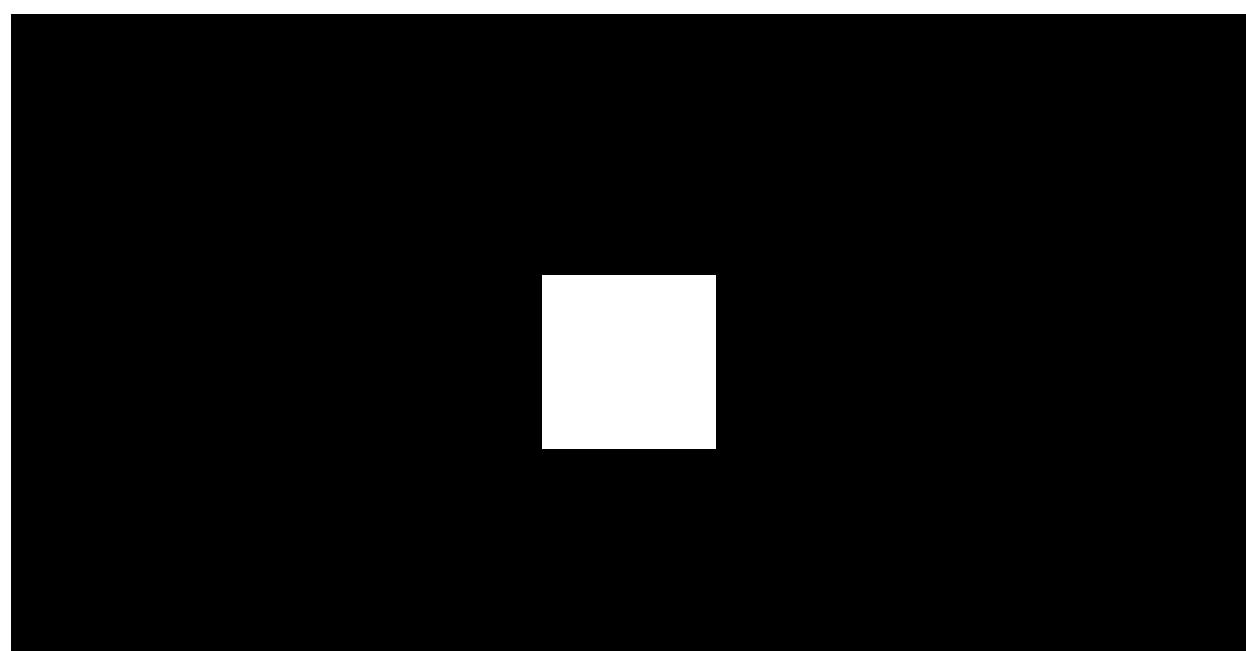
- ✓ En el caso de HDFS, los datos de los nodos que se paran se cogen de las réplicas que están en otros nodos del clúster.
- ✓ En el caso de YARN, los trabajos se distribuyen entre los servidores que están activos.

A continuación se va a mostrar un vídeo en el que, utilizando Apache Ambari, se van a parar la mitad de los nodos worker, simulando una caída por fallo de los servidores, y veremos cómo seguimos siendo capaces de ver los datos y de ejecutar aplicaciones, en este caso, consultas Hive, sin pérdida de servicio.

Situación inicial:

- ✓ Disponemos de un clúster con 4 nodos worker.
- ✓ Estamos realizando consultas en Hive a una tabla que contiene datos que están en un fichero en la ruta /hive/warehouse/managed/hivesampletable.

Veamos cómo Hadoop es capaz de sobreponerse a este tipo de situaciones:



00:00

02:45

Íñigo Sanz (Dominio público)

En el vídeo podrás encontrar los siguientes momentos relevantes:

- ✓ 24": se para manualmente uno de los 4 nodos worker.
- ✓ 1'20": se lanza la consulta Hive con un nodo parado y sigue funcionando.
- ✓ 1'40": se para manualmente otros de los 4 nodos worker, quedando sólo 2 activos.
- ✓ 2'29": se lanza la consulta con dos nodos parados y sigue funcionando.

5.- Cloudera Manager.

Caso práctico

El equipo de IT del Banco Español de Inversiones, BEI, comenzó a utilizar Apache Ambari para la administración de su cluster Hadoop, ya que inicialmente tenía una distribución de Hortonworks. Sin embargo, tras la compra de Hortonworks por parte de Cloudera, y la continuación de las versiones, hace unos meses tomaron la decisión de migrar todo el clúster a Cloudera.

El cambio no supuso un esfuerzo grande, pero a partir de ahora tienen que utilizar Cloudera Manager para administrar el clúster en lugar de Apache Ambari. En principio parece que no habría muchos cambios, pero veamos si son aplicaciones similares.



Gerd Altmann (Dominio público)

Cloudera Manager fue la primera herramienta de administración de Hadoop propiamente dicha. Fue desarrollada por Cloudera e incluida en su distribución de Hadoop, denominada Cloudera CDH, desde su origen en el año 2009.

En ese momento, no había ninguna herramienta alternativa en el ecosistema Apache, ya que Apache Ambari se lanzó en el año 2013.

Cloudera Manager siguió ofreciéndose en las distribuciones Cloudera y se sigue incluyendo hoy en día. De hecho, la desaparición de Hortonworks y del resto de distribuciones ha hecho que Apache Ambari se haya descontinuado, ya que la única distribución comercial de Hadoop, la de Cloudera, incluye Cloudera Manager en lugar de Ambari.

Sin embargo, ambas herramientas son muy similares en cuanto a funcionalidad. Las dos ofrecen capacidad de instalación, administración y monitorización de clústers, e incluyen un interfaz de usuario con una experiencia prácticamente idéntica.

Veamos algunas pantallas y funcionalidades de Cloudera Manager en un clúster de demostración de Cloudera.

La organización de opciones del menú de Cloudera Manager es muy parecida a la de Apache Ambari, separando los servicios (denominados Clústeres en Cloudera Manager) de los servidores donde está desplegado Hadoop (denominado Hosts). Al acceder a la primera pantalla, se ve un resumen de los servicios instalados en el clúster, así como las principales métricas.

Cloudera Manager

Búsqueda

Estado Todos los problemas de estado Configuración Todos los comandos recientes

cdptrialuser27-datalake

Cloudera Runtime 7.2.8 (Remesas)

2 hosts 1

Hive Metastore

atlas

hbase

hdfs 2

kafka

knox

ranger

solr

zookeeper

Gráficos

30m 1h 2h 6h 12h 1d 7d 30d

CPU de clúster

IO de disco del clúster

IO de red del clúster

Íñigo Sanz (Dominio público)

Haciendo clic en la opción “Clústeres”, se puede ver los diferentes servicios de instalados:

Cloudera Manager

Búsqueda

Clústeres

Hosts

Diagnóstico

Gráficos

Administración

cdptrialuser27-datalake

Cloudera Runtime 7.2.8 (Remesas)

Hive Metastore

atlas

hbase

hdfs

kafka

knox

ranger

solr

zookeeper

Hosts

Roles

Plantillas de host

Remesas

Enviar datos de diagnóstico

Informes

Informe de utilización

Pools de servicios estáticos

Cloudera Management Service

Íñigo Sanz (Dominio público)

Seleccionando uno de ellos, por ejemplo, Hive, se tiene una primera pantalla, denominada “Estado” con la información general del servicio: resumen del estado, historial, así como las principales métricas de monitorización.

The screenshot shows the Cloudera Manager web interface for the 'cdptrialuser27-datalake' cluster. The main page displays the 'Hive Metastore' service with a green checkmark icon and the status 'Estado'. Below the service status, there are sections for 'Pruebas de estado' (with a summary card for the Hive Metastore Server), 'Resumen de estado' (listing 'Hive Metastore Server' and 'Hosts' both in 'good health'), 'Historial de estados' (a log entry for a successful state transition on June 23 at 3:52:41 AM), and 'Gráficos' (two charts: 'Núcleos de CPU usados' and 'Estado'). The left sidebar contains navigation links for Clusters, Hosts, Diagnóstico, Gráficos, Administración, Remesas, Comandos en ejecución, Asistencia técnica, and trial27_admin. The bottom of the screen shows the Mac OS X dock with various application icons.

Íñigo Sanz (Dominio público)

Se puede navegar entre las pestañas, por ejemplo, accediendo a la pestaña Instancias se puede ver los servidores donde se encuentra desplegado el servicio:

The screenshot shows the same Cloudera Manager interface, but the 'Instancias' tab is selected under the 'Hive Metastore' service. The page displays a table of instances. The table has columns for Estado, Tipo de rol, Estado, Nombre de host, Estado de activación, and Grupo de roles. One instance is listed: 'Hive Metastore Server' (Rol: Hive Metastore, Estado: Iniciado, Host: cdptrialuser27-datalake-master0.cdptrial.akee-2isk.cloudera.site, Activado, Grupo de roles: Hive Metastore Server Default Group). The table includes filters and sorting options. The left sidebar and Mac OS X dock are also visible.

Íñigo Sanz (Dominio público)

En la pestaña "Configuración", se puede ver y modificar los parámetros de configuración del servicio.

The screenshot shows the Cloudera Manager interface for managing a service named 'cdptrialuser27-datalake'. The main navigation bar includes links for Correo, Calendario, Lease Trial User, Cloudera Management C..., Configuración - Hive Met..., Almacenamiento y proce..., and Introducción a Apache H... The left sidebar has sections for Clústeres, Hosts, Diagnóstico, Gráficos, Administración, Remesas, Comandos en ejecución, Asistencia técnica, and trial27_admin. The central pane is titled 'Hive Metastore' and shows the 'Configuración' tab selected. It features a search bar and a 'Filtros' section with dropdown menus for 'ÁMBITO' and 'CATEGORÍA'. A large table lists various configurations with their values and descriptions. At the bottom right of the table is a 'Guardar cambios(CTRL+S)' button.

Íñigo Sanz (Dominio público)

Asimismo, existe un botón “Acciones” que ofrece la posibilidad de detener, reiniciar o realizar otras actividades sobre el servicio.

This screenshot shows the 'Estado - Hive Metastore...' page. The 'Acciones' dropdown menu is open, displaying options like 'Iniciar', 'Detener', 'Reiniciar', and 'Entrar en modo de mantenimiento'. Below the actions, there's a 'Pruebas de estado' section with a green checkmark next to 'Estado de Hive Metastore Server'. The 'Resumen de estado' section shows 'Hive Metastore Server' and 'Hosts' both with a green checkmark and '1 con estado satisfactorio'. The 'Historial de estados' section shows a log entry from 'jun. 23 3:52:41 AM' stating 'Estado de Hive Metastore Server Satisfactorio 1 con estado satisfactorio'. On the right side of the page, there are two charts: one for 'Eventos y alertas importantes' and another for 'Estado de los servicios'. The bottom navigation bar includes links for Correo, Calendario, Lease Trial User, Cloudera Management C..., Configuración - Hive Met..., Almacenamiento y proce..., and Introducción a Apache H... The left sidebar is identical to the previous screenshot.

Íñigo Sanz (Dominio público)

A continuación se muestra las pantallas de administración del servicio HDFS, donde se puede comprobar que las opciones son similares a las aparecidas en Hive, o idénticas a las que se ofrecen en Apache Ambari.

Por ejemplo, la pantalla inicial de Estado muestra las métricas fundamentales de HDFS, así como el estado de los servicios:

The screenshot shows the Cloudera Manager interface for a cluster named 'cdptrialuser27-datalake'. The main navigation bar includes 'Safari', 'Archivo', 'Edición', 'Visualización', 'Historial', 'Marcadores', 'Ventana', 'Ayuda', and several tabs for monitoring services like 'Correo', 'Calendario', 'Lease Trial User', 'Cloudera Management C...', 'Estado - hdfs - Cloudera...', 'Almacenamiento y proce...', and 'Introducción a Apache H...'. The left sidebar has sections for 'Búsqueda', 'Clústeres' (selected), 'Hosts', 'Diagnóstico', 'Gráficos', 'Administración', 'Remesas', 'Comandos en ejecución', 'Asistencia técnica', and 'trial27_admin'. The main content area is titled 'cdptrialuser27-datalake' and 'hdfs'. It features a 'Resumen de HDFS' section with a chart showing 'Capacidad configurada' at 5,1 MiB/236,0 GiB. Below it is a 'Pruebas de estado' section with a button to 'Mostrar 7 correctos'. To the right are three line graphs: 'Capacidad HDFS' (blue line, constant at ~236G), 'Total de Bytes leídos en DataNodes' (blue line, fluctuating between 0.5b/s and 1b/s), and 'Total de Bytes escritos en DataNodes' (blue line, fluctuating between 0.93b/s and 1b/s). At the bottom is a 'Resumen de estado' table listing services: Balancer (Ninguno 1), DataNode (1 con estado satisfactorio), Gateway (Ninguno 1), NameNode (1 con estado satisfactorio), and SecondaryNameNode (1 con estado satisfactorio). The status bar at the bottom shows various Mac OS X icons.

Íñigo Sanz (Dominio público)

This screenshot shows the 'Historial de estados' (History) tab in the Cloudera Manager interface. The left sidebar is identical to the previous screenshot. The main content area lists historical events for the 'hdfs' service. The first event is 'Pruebas básicas de HDFS Incorrecto' (incorrect) on Jun 21 8:49:11 AM, with one entry labeled 'con estado incorrecto'. The second event is '5 con estado satisfactorio' (satisfactory) on Jun 21 8:49:06 AM, with one entry labeled 'con estado todavía preocupante'. The third event is '1 con estado preocupante' (worrisome) on Jun 21 8:48:46 AM, with four entries labeled 'con estado desconocido' (unknown) and one labeled 'con estado satisfactorio'. The fourth event is '8 con estado desactivado' (disabled) on Jun 21 8:47:10 AM. To the right of the history list are four line graphs: 'Total de Bytes leídos en DataNodes' (blue line, fluctuating between 0.5b/s and 1b/s), 'Total de Bytes escritos en DataNodes' (blue line, fluctuating between 0.93b/s and 1b/s), 'Total de Bloques leídos en DataNodes' (blue line, fluctuating between 0.01 and 0.1 blocks/second), and 'Total de Bloques escritos en DataNodes' (blue line, fluctuating between 0.01 and 0.1 blocks/second). The status bar at the bottom shows Mac OS X icons.

Íñigo Sanz (Dominio público)

En la pestaña de instancias, se puede comprobar que HDFS se encuentra desplegado en diferentes nodos, con un Namenode y un Datanode.

Estado	Tipo de rol	Nombre de host	Estado de activación	Grupo de roles
N/D	Balancer	cdptrialuser27-datalake-master0.cdptrial.akee-2isk.cloudera.site	Activado	Balancer Default Group
Iniciado	DataNode	cdptrialuser27-datalake-master0.cdptrial.akee-2isk.cloudera.site	Activado	DataNode Default Group
N/D	Gateway	cdptrialuser27-datalake-master0.cdptrial.akee-2isk.cloudera.site	Activado	Gateway Default Group
Iniciado	NameNode (Activo)	cdptrialuser27-datalake-master0.cdptrial.akee-2isk.cloudera.site	Activado	NameNode Default Group

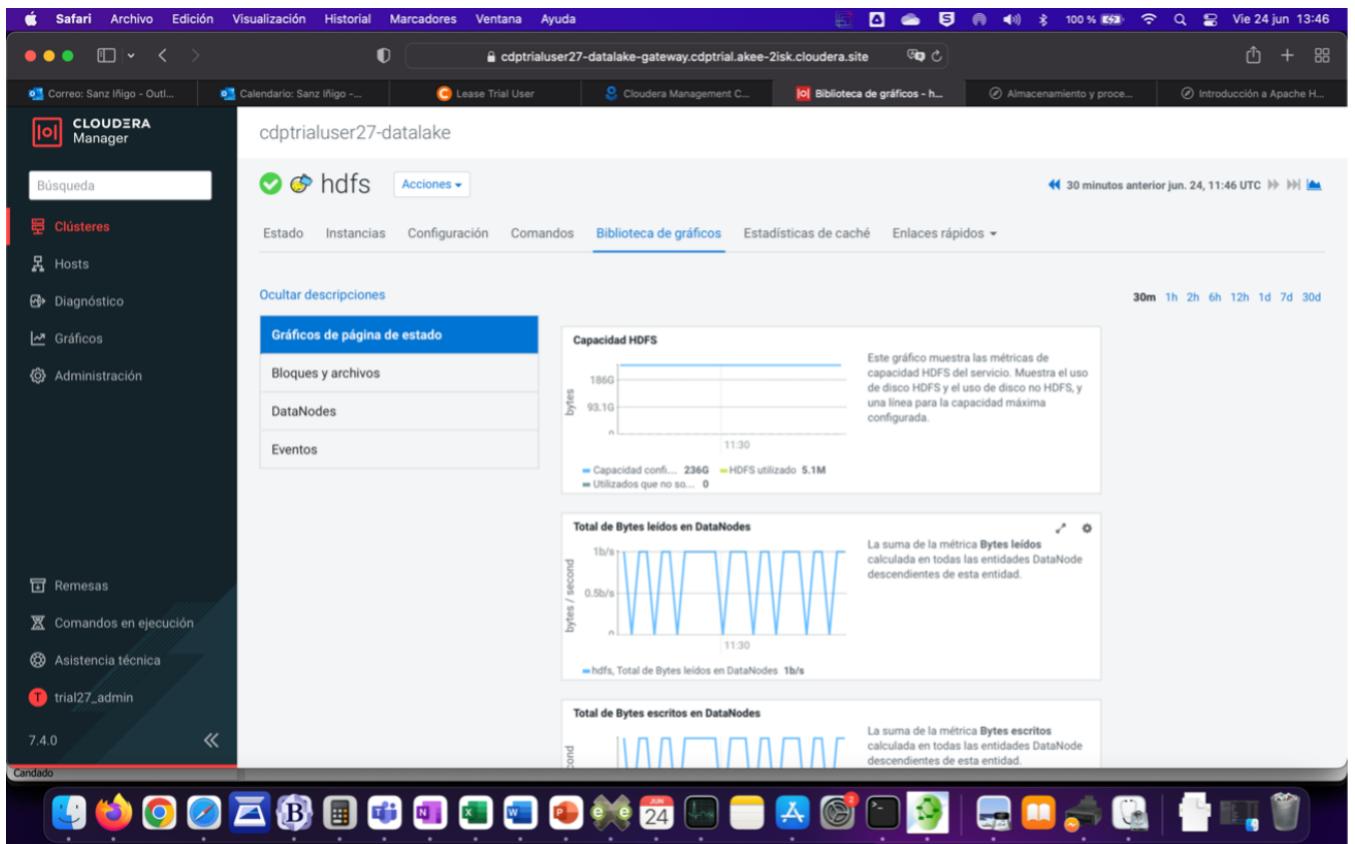
Íñigo Sanz (Dominio público)

En cuanto a la configuración, es similar a Hive, apareciendo las opciones relacionadas con HDFS:

Ambito	Configuración
dfs (Todo el servicio)	zookeeper_service
Balancer	zookeeper
DataNode	kms_service
Gateway	Requiere autorización adicional
HttpFS	none
JournalNode	
NFS Gateway	
NameNode	object_store_service
SecondaryNameNode	none
Failover Controller	

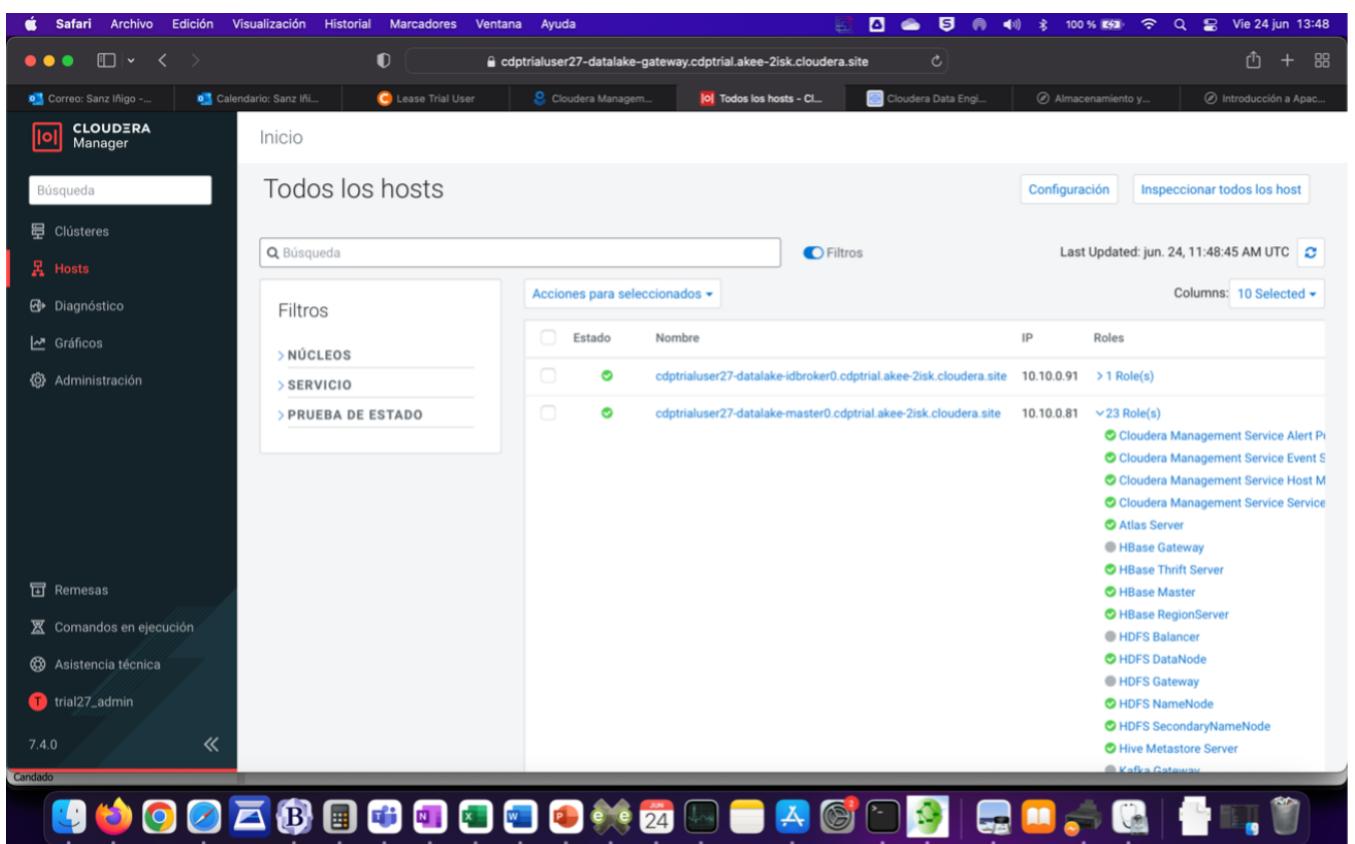
Íñigo Sanz (Dominio público)

La pestaña de “Biblioteca de gráficos” muestra la información de las métricas agrupadas por tipo:



Íñigo Sanz (Dominio público)

En cuanto a la opción de menú de Hosts, muestra los servidores que se están utilizando para el clúster. En este ejemplo, que se está utilizando un entorno de demostración de Cloudera, tenemos un clúster con dos servidores únicamente.



Íñigo Sanz (Dominio público)

En la opción de “Diagnóstico”, se muestran los mensajes de estado y las diferentes comprobaciones que se han realizado recientemente sobre los diferentes servicios y nodos del clúster:

The screenshot shows the Cloudera Manager interface on a Mac OS X desktop. The window title is 'cdptrialuser27-datalake-gateway.cdptrial.akee-2isk.cloudera.site'. The sidebar on the left has a dark theme with white icons and text. The 'Diagnóstico' option is selected. The main content area is titled 'Eventos' and shows a list of system logs. At the top of the log list, it says 'No hay ningún filtro seleccionado.' and 'Agregar un filtro.'. Below this, there are several log entries with details like date, level (IMPORTANT, INFORMATIONAL), service (hdfs, Cloudera Manager), role (SecondaryNameNode, Service Monitor), host (cdptrialuser27-datalake-master0), category (LOG_MESSAGE, AUDIT_EVENT), and message content. The interface includes a search bar at the top right and a toolbar with various icons at the bottom.

Íñigo Sanz (Dominio público)

En la opción “Gráficos” permite generar cuadros de mando a medida, y en la opción de Administración se dispone de opciones para la gestión de usuarios, versiones de los componentes, etc.

Para saber más

Si deseas obtener más información de Cloudera Manager, puedes acceder a la web oficial en [este enlace](#).

Asimismo, en la página de Cloudera tienes muchos vídeos de demostración de diferentes funcionalidades de Cloudera Manager. Puedes verlos en [este enlace](#).

Autoevaluación

Indica si las siguientes afirmaciones son verdaderas o falsas.

Cloudera Manager permite arrancar y parar los servicios del clúster.

Verdadero Falso

Verdadero

Verdadero: permite administrar todos los servicios de Hadoop instalados.

Cloudera Manager es una herramienta de trabajo para los data scientists.

Verdadero Falso

Falso

Falso: Cloudera Manager es una herramienta de trabajo para los administradores de sistemas.

6.- Ganglia.

Caso práctico

El equipo de IT del Banco Español de Inversiones, BEI, ha utilizado tradicionalmente otro sistema de monitorización de los diferentes clústers que había para otras aplicaciones, como los clústers de servidores web o de un grid de HPC.

Dado que conocen bien esa herramienta, que se llama Ganglia, se preguntan si podrían utilizarla para el clúster Hadoop, ya que en su día a día dominan su uso y podrían implementar cuadros de mando específicos para Hadoop, integrándolos en los existentes y centralizando la monitorización en una única herramienta.

Veamos cómo es Ganglia y si permite monitorizar clústers Hadoop.



[Thomas Ulrich](#) (Dominio público)

Ganglia es una herramienta opensource que permite la recogida de métricas de un sistema y su monitorización. No es una herramienta específica de Hadoop, ya que su propósito es ayudar en la monitorización de cualquier tipo de clúster, pero puede resultar útil en entornos en los que no se disponga de Apache Ambari o Cloudera Manager.

Ganglia se puede ejecutar, por lo tanto, en los nodos del clúster, de modo que Hadoop pueda enviar los datos de métricas a los agentes de Ganglia o recogerlos directamente del sistema operativo en el que se ejecutan los servicios de Hadoop en cada nodo. Además, Ganglia se puede integrar con Nagios para montar un sistema de alertas sobre las métricas recogidas en Ganglia.

Ganglia recopila métricas como el uso de la CPU y el espacio libre en el disco y también puede ayudar a monitorizar los nodos que están caídos.

Arquitectura de Ganglia

Hay cuatro componentes principales en un sistema de monitoreo de Ganglia:

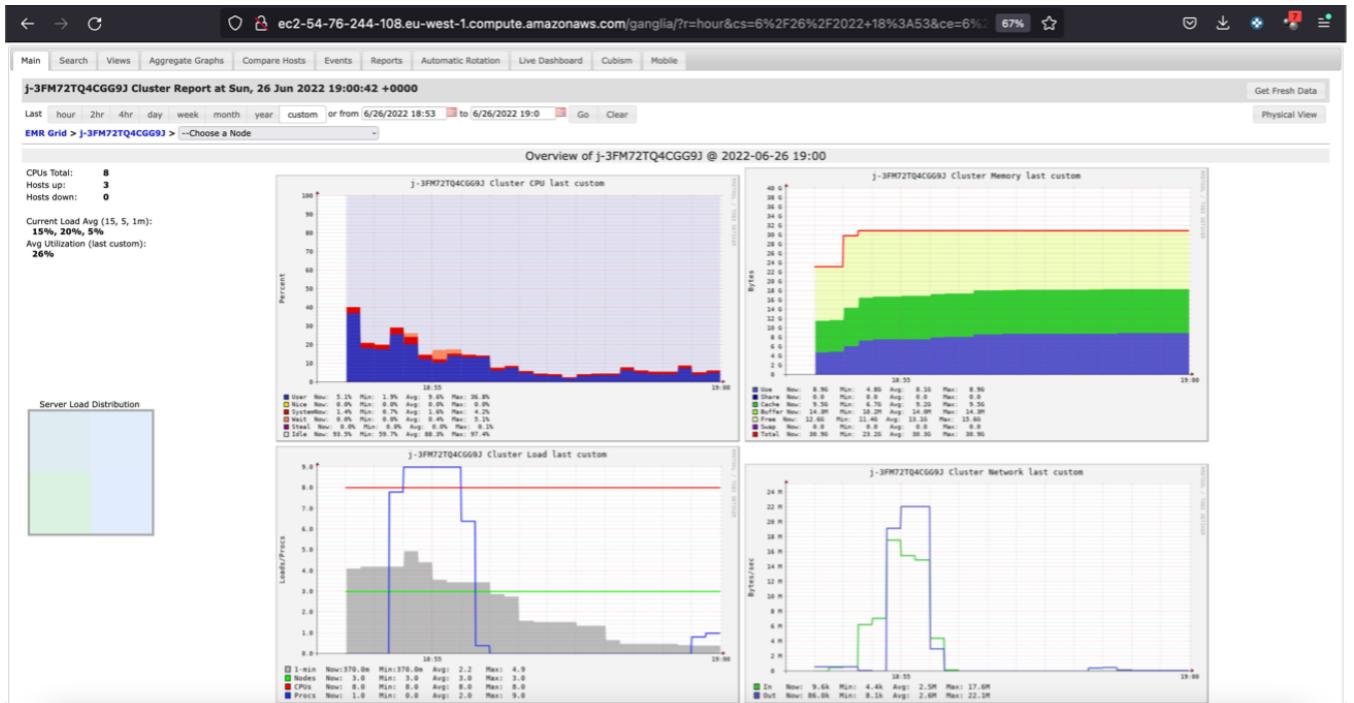
- ✓ **gmond**: es un demonio que se ejecuta en cada nodo del clúster, cuyo trabajo es recopilar los datos de métricas de cada nodo. Cada nodo ejecuta el demonio gmond y el nodo recibirá métricas del resto de los nodos del clúster (todos los nodos se comunican con todos los nodos). Esto significa que el proceso de recogida de métricas (gmetad, que se explica a continuación) necesita solo un nodo para obtener las métricas del clúster y también en caso de un fallo de un nodo, Ganglia sigue proporcionando servicio.
- ✓ **gmetad**: este es el demonio que sondea los nodos en busca de datos de las métricas. Puede obtener un volcado de métricas para todo el clúster desde cualquier nodo del clúster. El demonio gmetad crea tablas denominadas RRD para almacenar los datos de métricas.
- ✓ **RRDtool**: este componente almacena los datos de las métricas recogidas por el demonio gmetad en cada nodo.
- ✓ **gweb**: esta es la interfaz web para visualizar las métricas recopiladas por el sistema de monitoreo de Ganglia a través de los datos almacenados en las bases de datos de RRD. Puede ver métricas específicas mediante gráficos y también crear gráficos personalizados profundizando en los detalles de una métrica o host específico. El proceso gweb es en realidad un programa PHP que se ejecuta en un servidor web Apache.

Funcionalidad de Ganglia

La instalación de Ganglia consiste en instalar los agentes en cada uno de los nodos y levantar gweb, el interfaz web, para la visualización de las métricas.

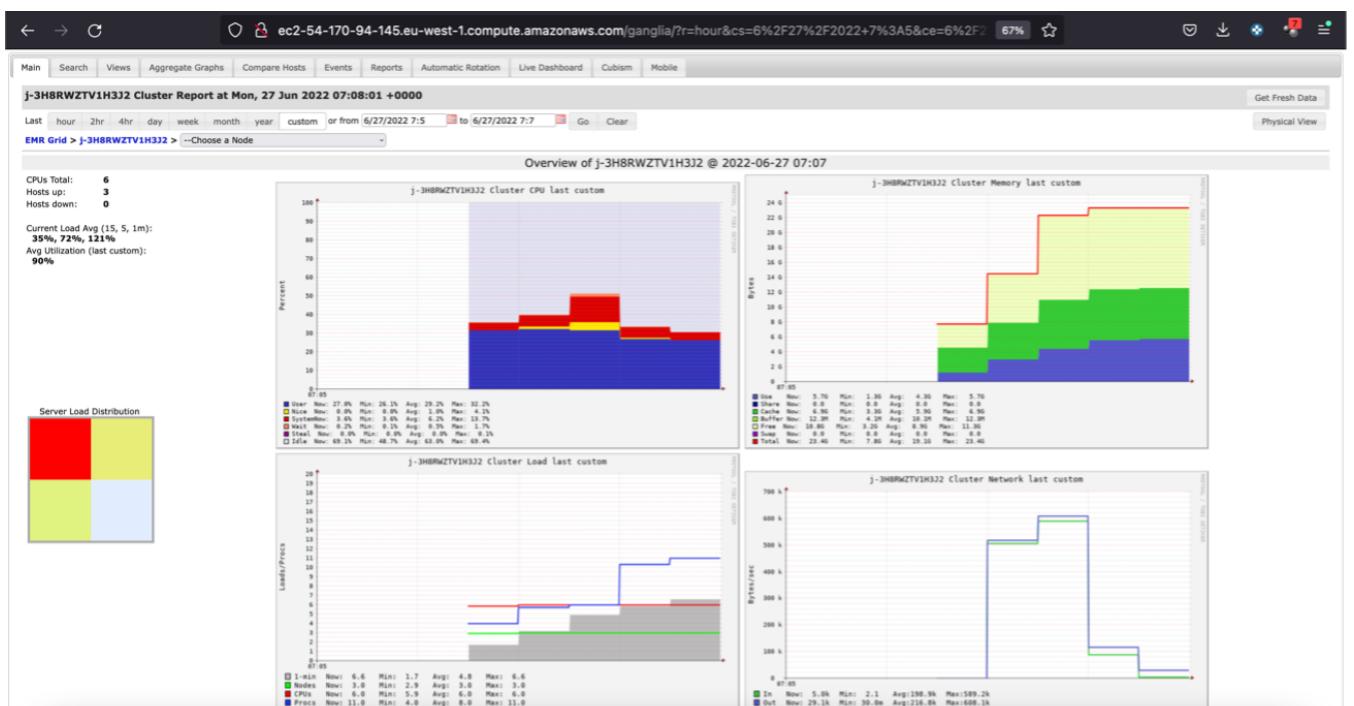
A continuación vamos a conocer las principales pantallas y funcionalidades que ofrece gweb, y para ello, vamos a visualizar cómo sería alguna pantalla de Ganglia para un clúster de 3 nodos de Hadoop en Amazon Web Services.

Al acceder a la pantalla principal, se muestran las principales métricas agregadas de los 3 nodos:



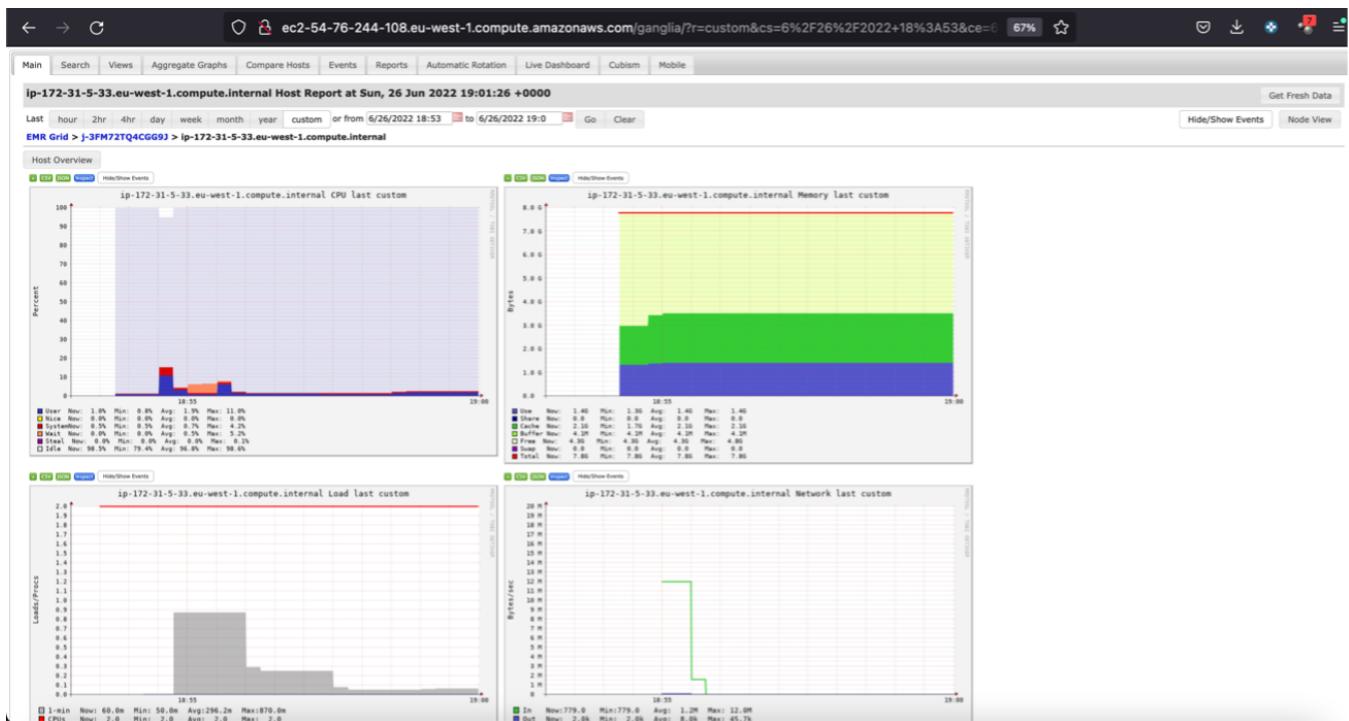
Iñigo Sanz (Dominio público)

Se puede ver que muestra como gráfico la carga de CPU, de memoria o red. En caso de que algún nodo tenga alguna métrica por encima de los umbrales establecidos, se muestra en el mapa de la izquierda en rojo.

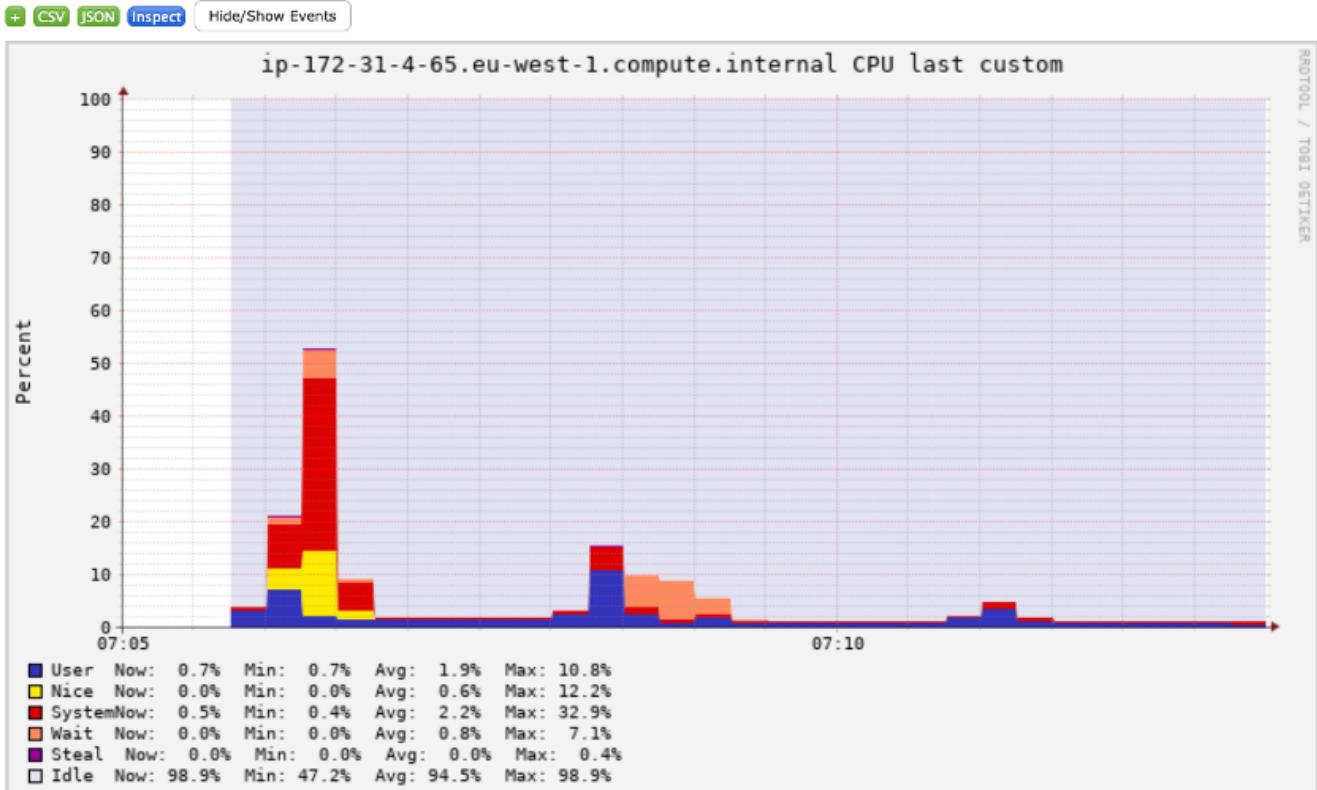


Este mapa, representado en forma de cuadrado, suele contener un cuadrado más pequeño por cada nodo. En nuestro ejemplo hay sólo 3 cuadrados al haber sólo 3 nodos, pero en un sistema de producción con múltiples nodos, se divide en cuadrados más pequeños, y permitiendo en un vistazo rápido conocer el estado de los nodos del clúster.

En el desplegable que aparece en la parte superior izquierda, se puede elegir un nodo concreto del clúster para visualizar únicamente sus métricas:



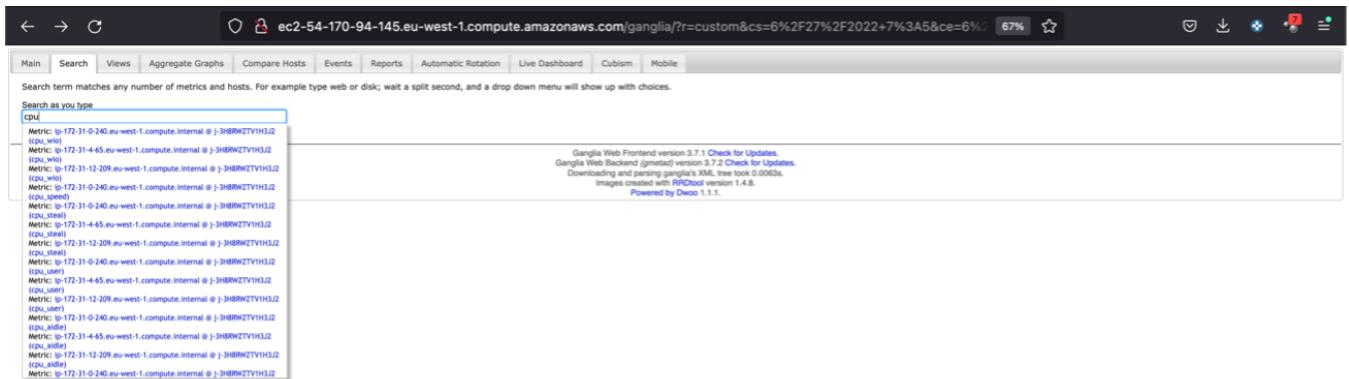
Todas las gráficas ofrecen la posibilidad de exportar los datos en CSV o JSON, como se puede ver en los botones verdes que aparecen en la parte superior del gráfico:



Íñigo Sanz (Dominio público)

O pulsando en el botón “+” se puede definir los umbrales de aviso y añadir el gráfico a un dashboard a medida que se puede construir en la pestaña “Views”.

En la pestaña “Search” se puede buscar por cualquier nodo o métrica, por ejemplo, buscando las métricas que contengan “cpu”



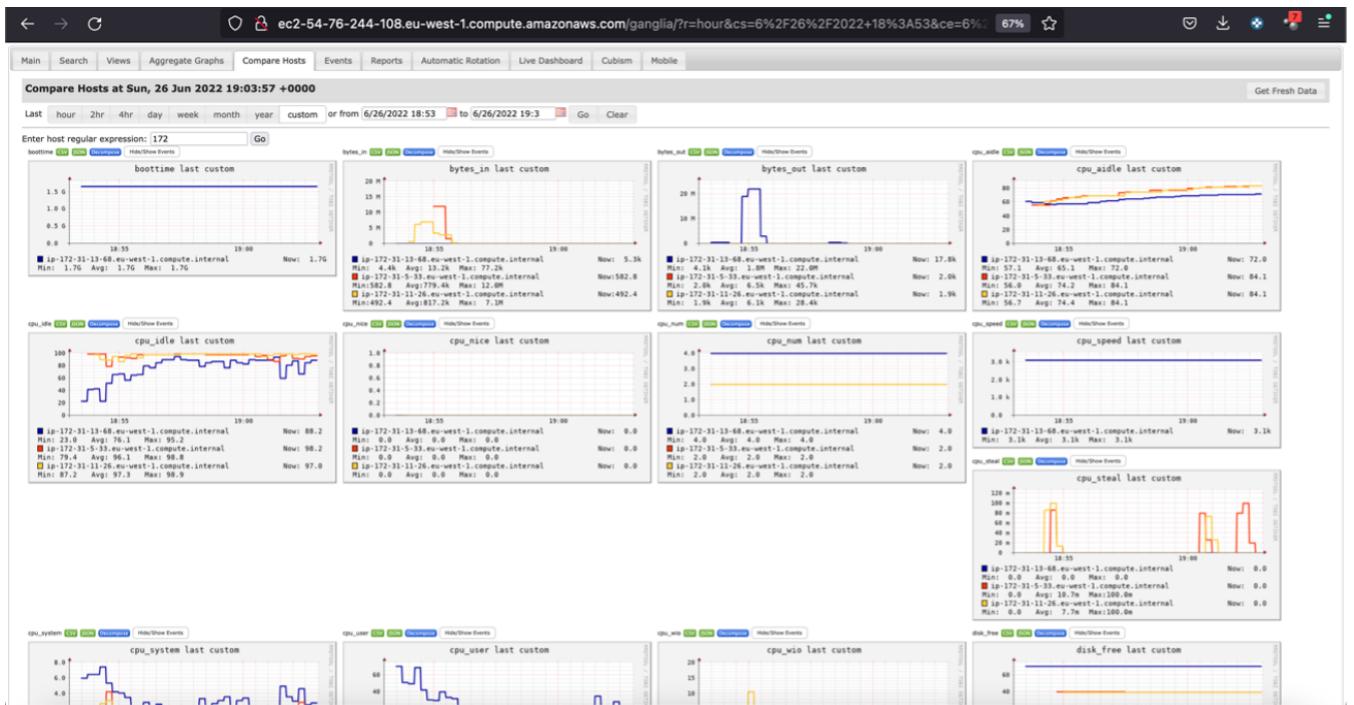
Íñigo Sanz (Dominio público)

O buscando los nodos que contengan el nombre “172” (todos los nodos del clúster del ejemplo tienen un nombre que empieza por 172).



Íñigo Sanz (Dominio público)

Las capacidades que ofrece Ganglia para la monitorización de sistemas distribuidos es muy amplia, pudiendo generar métricas agregadas, cuadros de mando a medida e infinitad de posibilidades. Por ejemplo, en la siguiente imagen se muestra un cuadro de mando comparando diferentes métricas de los 3 nodos que componen el clúster. Este tipo de cuadros de mando puede resultar muy útil para detectar problemas en algunos de los nodos:



Íñigo Sanz (Dominio público)

Sin embargo, pese a que Ganglia es un buen sistema de monitorización, la realidad es que en los clústers en los que se dispone de herramientas como Ambari o Cloudera Manager, éstas reemplazan por completo a Ganglia, ya que permiten no sólo monitorizar, sino también administrar el clúster en una única herramienta.

Para saber más

Si quieres conocer más información sobre Ganglia, puedes acceder a su página oficial en [este enlace](#).

Autoevaluación

¿Qué diferencias hay entre Ganglia y Ambari o Cloudera Manager?. Selecciona las opciones correctas:

- Ganglia no permite modificar parámetros de configuración de Hadoop.

- Ganglia no permite realizar acciones en Hadoop como parar un servicio o un nodo.

- Ganglia ofrece métricas de uso de CPU en los nodos, mientras que Ambari o Cloudera Manager no.

[Mostrar retroalimentación](#)

Solución

1. Correcto
2. Correcto
3. Incorrecto

7.- Guía práctica de Spark en Databricks

En esta guía vamos a aprender a usar Spark en [Databricks](#). Azure Databricks es un clúster de Spark que se puede probar de forma gratuita. Para ello necesitas una cuenta "community". Si no dispones de una, es el momento de crearla. El fichero adjunto es un "notebook" de Databricks que debes importar. Para hacerlo, descárgalo en tu disco duro y ve a la consola de Databricks y en el menú lateral pulsa sobre "Workspace"; se abrirá un panel lateral, pulsa botón derecho sobre el panel y luego "import" y selecciona el fichero descargado. Una vez importado verás que en el entorno "workspace" tienes un directorio llamado "apache-spark-programming-with-databricks", y dentro distintos subdirectorios. Esta guía consta de los libros contenidos en los subdirectorios: "ASP 1", "ASP 2", "ASP3".

En el vídeo adjunto se realiza el proceso de importación (usando una URL en vez de descargando el fichero) y se explica paso a paso cada uno de los libros.

Guía Práctica BDA04



Guía práctica BDA04

- [Notebook de Databricks \(Ventana nueva\)](#)