

hdf-hadoop-00-nodos

parámetros del entorno >

mv-name	hdfs	ip	-
hadoop-template	-	192.168.0.100	-
hadoop-nodo01	namenode	192.168.0.101	-
hadoop-nodo02	datanode	192.168.0.102	-
hadoop-nodo03	datanode	192.168.0.103	-
hadoop-nodo04	datanode	192.168.0.104	-
hardware	value		
ram	2048MB		
procesadores	2		
internal-disk	100G		
data-disk	500G		
user	password		
hdfs-admin	abc123.		

install guest additions >

```
# install guest additions
sudo apt-get install gcc make perl
```

connection host-nodos >

```
ssh hdfs-admin@192.168.0.100
ssh hdfs-admin@192.168.0.101
ssh hdfs-admin@192.168.0.102
ssh hdfs-admin@192.168.0.103
ssh hdfs-admin@192.168.0.104
```

hdf-hadoop Entorno >

1. configurado la dirección IP fija de cada uno de los nodos

2. cambiado el nombre del nodo

3. hemos incluido en su archivo Host las ips los nombres del resto de los nodos

4. instalado Java en su versión 8

- además hemos creado una variable de sistema que se llama Java home a la que le hemos dado la ruta de este Java versión 8

- lo hemos hecho en el archivo `/etc/environment`

- porque esto nos permite que esta variable esté disponible Incluso si no hemos iniciado sesión Dentro de este equipo solamente lo hemos encendido

5. instalado el servidor ssh

- pienso que no sería necesario pero hemos des-comentar las líneas del puerto y de las ips aceptadas hemos reiniciado el servicio hemos generado las claves públicas y privadas y las hemos compartido entre todos los equipos incluido el local Host después hemos

6. descargado hadoop lo hemos descomprimido y lo hemos dejado en una carpeta con un nombre amigable porque lo vamos a usar mucho después de esto hemos vuelto al archivo `/etc/environment` para Añadir un par de cosas en primer lugar

- hemos añadido al path las rutas de la carpeta Bin y sbin de hadoop para hacer accesibles todos sus comandos desde cualquier punto del sistema

- hemos añadido una nueva variable global llamada hado home en la que hemos puesto la ruta base donde están los archivos de hadoop

7. por último he configurado ese segundo disco duro que teníamos

- para que siempre en cada arranque se monte en una carpeta que se llama disc grande cuando configure hado le indicaré que deje sus archivos dentro de esta carpeta en principio con esto acabamos la parte de

Config

hdf-hadoop Conf >

1.- hadoop - identificar el nodo que actúa como principal `hdfs-hadoop/archivos/core-site.xml` `core-site.xml`

2.- hadoop conf nodos `hdfs-hadoop/archivos/hdfs-site.xml` `hdfs-site.xml`

- [2.1 hdfs-site.xml-namenode](#)
- [2.2 hdfs-site.xml-datanode](#)

3.-establecer la tabla de particiones

4.- levantar todos los nodos

hadoop trabaja en una estructura distribuida pero centralizada
hay un nod principal

```
ls ~/hadoop/etc/hadoop/core-site.xml  
tree ~/hadoop/etc/hadoop/
```

📁 **archivos** >

📝 **tree** >

/home/hdfs-admin/hadoop/etc/hadoop/

├── **capacity-scheduler.xml** // identifica al nodo principal

├── **decomisados** // archivo que crearemos nosotros

├── **hdfs-site.xml** // configuramos las propiedades de nodo-datanode y nodo-namenode

├── **workers** //indicaremos los nombres de los nodo-datanode

├── configuration.xsl

├── container-executor.cfg

├── core-site.xml

├── hadoop-env.cmd

├── hadoop-env.sh

├── hadoop-metrics2.properties

├── hadoop-policy.xml

├── hadoop-user-functions.sh.example

├── hdfs-rbf-site.xml

├── httpfs-env.sh

├── httpfs-log4j.properties

├── httpfs-site.xml

├── kms-acls.xml

├── kms-env.sh

├── kms-log4j.properties

├── kms-site.xml

├── log4j.properties

├── mapred-env.cmd

├── mapred-env.sh

├── mapred-queues.xml.template

- |— `mapred-site.xml`
- |— `shellprofile.d`
- | |— `example.sh`
- |— `ssl-client.xml.example`
- |— `ssl-server.xml.example`
- |— `user_ec_policies.xml.template`
- |— `yarn-env.cmd`
- |— `yarn-env.sh`
- |— `yarnservice-log4j.properties`
- |— `yarn-site.xml`

table >

archivo	dirección	uso	descripción
<code>capacity-scheduler.xml</code>	<code>sudo nano ~/hadoop/etc/hadoop/capacity-scheduler.xml</code>	indica que nodo actúa como nodo principal	
<code>ssh_config</code>	<code>sudo nano /etc/ssh/ssh_config</code>	archivo de configuración de servidor del servicio ssh	lo usaremos para indicar el puerto y permitir todas las conexiones

comandos >

comando	descripción
<code>hadoop conftest</code>	indica si los archivos de hadoop estan sintacticamente bien contruidos en cuanto a sus etiquetas, no identifica los errores del contenido de las mismas
<code>scp ~/hadoop/etc/hadoop/core-site.xml nodo02:~/hadoop/etc/hadoop/</code>	copiar archivo de origen al nodo02
<code>sudo systemctl restart ssh</code>	reiniciar servicio ssh
<code>sudo systemctl status ssh</code>	ver estado de servicio ssh

comando	descripción
<code>sudo systemctl restart sshd</code>	reiniciar servicio sshd
<code>ls -la .ssh/</code>	lista clave publica (.pub) y privada

capacity-scheduler.xml >

identificar el `nodo01` como `nodo principal`
replicarlo en todos los nodos

Todos los nodos deben saber cual es el nodo principal

```
sudo nano ~/hadoop/etc/hadoop/capacity-scheduler.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://nodo01:9000</value>
  </property>
</configuration>
```

[!my-folder]- hdfs-site.xml

configurar el `nodo01` como `nameNode`

```
sudo nano ~/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://nodo01:9000</value>
  </property>
</configuration>
```

Transcription >

Configurar entorno >

- [00:04](#)

Hola arrancamos Este vídeo dedicado a instalar y configurar la capa hdfs de hado 3.4 Aprovechando que se acaba de liberar la versión 3.4 de hadoop vamos a recordar cómo es el procedimiento de instalación y configuración de la capa hdfs para ello os presento este escenario con cuatro nodos el primero de ellos tendrá la función exclusivamente de name node y los otros tres tendrán solamente la función de datanode vamos a configurarlo desde el principio y para ello lo primero que vamos hacer va a ser asignar direcciones IP fijas eso lo

- [00:33](#)

haremos con el comando sudo Nano etc net plan y este nombre tan largo Aquí eliminamos esta línea no vamos a necesitar ningún gestor de Red porque lo vamos a configurar a mano añadimos la línea ethernet y ahora tenemos que poner el nombre de la interfaz si no lo sabemos abrimos una nueva terminal hacemos ipa y aquí nos pone el nombre de la interfaz np0 s1 np0 s1 le damos la dirección con adrees y lo que queramos poner con esto sería suficiente pero yo quiero que tenga salida internet así que le vamos a dar una ruta por defecto

- [01:04](#)

también quiero que resuelva dns con lo cual le vamos a dar un par de direcciones bien con esto ya lo tendremos configurado guardamos y ahora vamos a probarlo no da problemas Eso es bueno abrimos una nueva ventana hacemos un ping a Google parece que funciona así que lo damos por bueno comprobamos y ya lo tenemos listo el segundo paso consiste en darle un nombre a este equipo esto lo hacemos mediante el comando sudo Nano etc hostname y aquí le vamos a poner equipo 01 guardamos y salimos también debemos modificar el

- [01:30](#)

archivo Host aquí se guardan las relaciones entre IP y nombre que el ordenador conoce esta línea la 127 011 la vamos a borrar y vamos a Añadir la de todos los nodos del clúster muy bien lo tenemos guardamos salimos y vamos a probarlo vamos a hacer un pin a equipo 03 por ejemplo que ya los tengo configurados y aquí funcionaría siguiente paso Vamos a instalar Java en la versión 8 en teoría hdfs debería funcionar con Java 11 también pero no todas las herramientas de hado están aún actualizadas así que vamos a instalar la

- [01:58](#)

versión 8 porque aunque es antiguo es totalmente compatible con todas las herramientas de Had para ello lo primero actualizamos lo hacemos con sudo apt update mal sudo muy bien y ahora sudo apt install openjdk -8 gjdk error al escribir

bien ahora que lo tenemos instalado vamos a dar uno de los pasos más importantes en la instalación de Job y este consiste en crear una variable global que se llame Java home con la ruta a la versión de Java que hado va a utilizar esta ruta la podemos obtener de muchas maneras pero la más

- [02:29](#)

sencilla aprovechando que tenemos la instalación aquí es subir un poquito y aquí la tenemos normalmente aquí nos aparecerán unos ejecutables dentro de la carpeta bin pues Tendremos que subir un nivel más con lo cual la ruta es esta la copio y ahora tengo que pegarla en uno de los archivos que se leen al arrancar la sesión podría hacerlo en varios sitios pero el lugar que garantiza que va a funcionar 100% seguro es en el archivo bar etc bar environment Así que lo editamos es un archivo de configuración del sistema con lo cual

- [02:55](#)

necesita permisos administrativos y aquí lo tenemos aquí en una nueva línea ponemos el nombre de la variable igual y entre comillas vamos a pegar la ruta que acabamos de copiar hace un momento y guardamos voy a pegarle un reinicio para que se actualice todo veo que el nombre del equipo ya está correctamente cambiado y con el comando env compruebo las variables de sistema que están establecidas aquí tengo que buscar la mía y ahí la tenemos jav home con la ruta actualizada muy bien el siguiente paso consiste en

- [03:23](#)

instalar el servidor ssh esto lo hacemos con `sudo apt install Open ssh server` bien ya está instalado analizando la salida veo que me he creado un archivo de configuración para `>>>` el servidor Lo noto porque tiene un nombre aquí con una `d` `sshd` Esto hace referencia al servidor y también veo que se ha creado un servicio probablemente para iniciar el demonio en cada arranque esto pienso que debería funcionar sin ningún problema pero por si acaso voy a editar el archivo de configuración etc `ssh sshd gu config` y voy a descomentar este par de

- [03:52](#)

líneas Puerto 22 y que Escucha todas las direcciones en principio es el funcionamiento por defecto pero así ya lo dejo explícito `control x guardar y salir` como he hecho cambios en un archivo de configuración lo apropiado Es reiniciar el servicio esto lo hago con `sudo Systemctl restart sshd` ssh en principio está listo vamos a comprobarlo `sudo Systemctl` hacemos ahora un estatus ssh y

Aparentemente veo que está bien está escuchando todas las direcciones a través del pto de 2 está activo y está enable lo que indica que se iniciará con

- 04:22

cada arranque voy a aprovechar que estoy con ssh ahora para generar mi pareja de llaves esto lo hago con ssh-keygen sin más argumentos no hacen falta me pregunta por el nombre lo dejo en blanco alguna contraseña lo dejo en blanco y me genera Mi par de llaves la pública y la privada vamos a comprobarlo hacemos un ls -la dentro de la carpeta oculta ssh y aquí tenemos las dos llaves ID rsa la privada ID rsa.

- 04:49

public de pública limpiamos la pantalla y ahora vamos a descargar hup para averiguar la ruta desde donde queremos Descargar abrimos un navegador y buscamos las descargas de hado nos vamos a ellas y buscamos lo que nos interese acaba de salir la versión 3.4 es la nueva acaba de salir Hace unos días Pues vamos a aprovechar para instalarla Recuerda que si tu equipo tiene un procesador Intel o amd debes descargarte esta versión Yo tengo un procesador Arm con lo cual tengo que Descargar esta versión pincho en ella y aquí tengo la ruta del archivo que me

- 05:15

tengo que descargar pues la copio y la descargo con el comando wget pego la dirección intro muy bien una vez descargado comprobamos ls -la ahí tengo el archivo son 900 meg pues vamos a descomprimirlo esto lo hacemos con el comando tar -xzf y el nombre del archivo volvemos a comprobar con un ls -la y ahí tengo la carpeta tiene una d aquí esto es una carpeta con el nombre hop-1 es un nombre un poco más largo de lo que me gustaría porque voy a tener que utilizarlo al configurar variables y poner rutas con lo cual lo voy a cambiar

- 05:46

con el comando mv a solamente hado vuelvo a hacer un ls -la y ahí lo tengo voy a abrir un momento un explorador de archivos para recordar las rutas que teníamos dentro de hado Aquí lo tengo hado y aquí vemos la estructura de las carpetas que lo componen todas son importantes pero ahora en la instalación nos van a interesar tres nos va a interesar la carpeta Bin y la carpeta sbin Aquí están los archivos ejecutables que en el siguiente paso Vamos a introducir dentro del path del sistema operativo y también más adelante

- [06:13](#)

nos va a interesar la carpeta etc etc es una carpeta típica de sistemas Linux donde están los archivos de configuración y aquí pasa Exactamente eso Si entramos dentro de etc Hay una carpeta que se llama hado y dentro de hado encontraremos los archivos de configuración que vamos a tener que modificar Así que esta ruta hado etc do la vamos a usar Pues habitualmente en este vídeo bien ahora que lo conocemos vamos con los cambios volvemos al terminal y vamos a Añadir esas dos carpetas que acabamos de comentar Bin y

- [06:38](#)

ese Bin dentro del P del sistema operativo cómo hacemos eso pues nuevamente nos vamos a el archivo etc environment y ahí encontramos el pth con las rutas por defecto que tiene el sistema operativo lo que vamos a hacer es Añadir estas dos nuevas rutas nos vamos al final y antes de las comillas separamos de la ruta anterior con dos puntos ponemos la ruta completa en mi caso es home administrador que es el usuario que estoy utilizando hado y dentro de hadou teníamos la carpeta Pin y la carpeta esbin que tengo que volver

- [07:05](#)

a poner home administrador hado esbin con esto lo que consigo Es que puedo llamar a los comandos de hup desde cualquier carpeta del sistema voy a aprovechar que estoy aquí para crear una nueva variable del sistema que se va a llamar jop home esta variable es tan importante como el Java home aquí le vamos a indicar Dónde está la raíz de los archivos de hado y gracias a ello a partir de ahí él sabrá dónde Buscar todas las cosas la ruta de jup está en Home admin administrador hado guardamos y salimos aprovechamos este momento para

- [07:33](#)

meterle un reinicio lo primero que hacemos Es comprobar a ver si están vemos que las rutas de Bin y sbin de jop están listas jop home también y Java home ya estaba antes ahora Sigue estando bien esta parte la tenemos Recuerda que hado quiere hacer conexiones ssh pero sin necesidad de introducir contraseña cuando realiza alguna comunicación con el resto de nodos para poder hacer eso es necesario intercambiar las llaves públicas entre todos los nodos incluso con el propio local Host para hacer eso utilizaremos El Comando

- [08:02](#)

ssh-copy-id con el parámetro menos y para instalar al que le indicaremos la ruta donde está la llave pública y a continuación el destino donde la queramos instalar. Pues en este caso va a ser con el usuario administrador en local Host como es la primera vez que nos conectamos nos pregunta si estamos seguros Sí y la contraseña de este nodo claro no me deja porque está mal escrito el nombre del usuario administrador ahora sí ya la hemos añadido vamos a hacer lo mismo pero en lugar de local Host vamos a poner equipo 01 que es el mismo pero a

- 08:32

pesar de ser el mismo como tiene otro nombre no lo reconoce directamente le decimos que sí a esta nueva conexión lo que pasa es que nos dice que ya existía esta llave dentro pero lo que hemos conseguido es que podamos hacer ahora un ssh sin contraseña tanto a local Host como a equipo 01 vamos a hacer esto mismo con el resto de los nodos se la mandamos al nodo dos conectamos con él por primera vez nos pide la contraseña y la añadimos vamos al equipo tres Es primera vez que conectamos nos pide la contraseña y la instalamos vamos a al

- 09:00

equipo cuatro primera vez contraseña y la instalamos para que la comunicación se pueda iniciar en ambos sentidos será necesario hacer esto mismo en el resto de los nodos de manera que todos se puedan conectar entre todos sin contraseña una vez que hemos terminado con el intercambio de claves vamos a probarlo entonces hacemos un ssh equipo 4 y conectamos sin necesidad de contraseña vamos a hacer otro al equipo dos conectamos bien y vamos por ejemplo a hacer uno desde aquí al equipo tres y conectamos correctamente vamos ahora a

- 09:30

configurar la parte de almacenamiento no es necesario pero en la práctica nosotros instalamos un segundo disco duro dedicado exclusivamente a almacenar los datos de la capa hdfs Aquí vamos a hacer lo mismo en primer lugar vamos a hacer un lsblk para ver los dispositivos de bloques que hay en este sistema como ves aquí hay un disco de 500 Gb que no está configurado Así que en primer lugar vamos a particionar sudo fdisk y la ruta dep y en mi caso vdb pulso intro contraseña y me dice que el dispositivo como no contiene una tabla de

- 10:02

particiones reconocida se ha creado un identificador antiguo si lo vemos con la orden p vemos que la etiqueta nos indica que va a estar formateado según mbr vamos a cambiarlo a gpt y volvemos a comprobar ahora ya sí creamos una nueva

partición con la *n* valores todos por defecto para que nos tome el total del valor volvemos a comprobar y ahí tenemos el disco bien configurado particiones modernas *gpd* y una única partición con el total del disco *W* para guardar y volvemos a comprobar con un *ls blk* ahora ya vemos que el disco con 500 Gb

- [10:33](#)

tiene una partición también de 500 Gb vamos a formatearla *make filesystem x4* y la ruta *dep vdb* partición 1 eamos intro y ya tenemos el disco listo lo tenemos particionado y lo tenemos formateado vamos a asignar la carpeta de montaje voy a crear una carpeta que se llama disco grande y lo que quiero es que en todos los inicios en todos los arranques se Monte ese segundo disco duro en esta carpeta para hacer esto tengo que editar con permisos administrativos el archivo etc file System Table *fs t* Aquí voy a Añadir una línea con seis columnas en

- [11:08](#)

primer lugar la ruta del dispositivo *dep bar vdb 1* separo con un tabulador y ahora le doy la ruta de la carpeta donde se va a montar *home administrador disco grande* vuelvo a separar con un tabulador y le indico el sistema de ficheros *x4* separo con un tabulador y escribo las opciones que me interesen como solo lectura o que no *H* que escriba directamente en disco yo lo voy a dejar con la configuración por defecto y a continuación dos ceros porque no queremos ni que se compruebe ni indicarle el orden en el que habría que

- [11:38](#)

comprobarse muy bien guardamos salimos y vamos directamente a probarlo antes voy a hacer un *lsblk* para confirmar que la partición está creada pero no está montada no tiene ninguna línea Aquí vamos a ver qué pasa después del reinicio a la vuelta vuelvo a hacer un *ls blk* y ahora ya sí confirmo que el disco tiene una partición y que está montada en la carpeta que le hemos dicho Aquí hay un pequeño problema y es que esta carpeta la ha montado con permisos administrativos Por lo cual solo un superusuario puede escribir en ella

- [12:09](#)

Vamos a cambiar esto con el comando *sudo ch men r* mayúscula para que sea recursivo el nombre del usuario que queremos que sea propietario y su grupo Y por último la ruta *home administrador disco grande* si ahora volemos a hacer un *ls men* la veremos que a partir de este momento la carpeta disco grande ya me perten y ya puedo escribir tranquilamente Muy bien pues en este punto en lo que se refiere a la

configuración del sistema operativo ya Lo tendríamos todo listo vamos a hacer un repaso de lo que hemos hecho primer

- [12:39](#)

lugar hemos configurado la dirección IP fija de cada uno de los nodos le hemos cambiado el nombre y hemos incluido en su archivo Host las ips y los nombres del resto de los nodos después hemos instalado Java en su versión 8 y además hemos creado una variable de sistema que se llama Java home a la que le hemos dado la ruta de este Java versión 8 lo hemos hecho en el archivo bar etc bar environment porque esto nos permite que esta variable esté disponible Incluso si no hemos iniciado sesión Dentro de este equipo solamente lo hemos encendido

- [13:07](#)

además también hemos instalado el servidor ssh pienso que no sería necesario pero hemos descom las líneas del puerto y de las ips aceptadas hemos reiniciado el servicio hemos generado las claves públicas y privadas y las hemos compartido entre todos los equipos incluido el local Host después hemos descargado hup lo hemos descomprimido y lo hemos dejado en una carpeta con un nombre amigable porque lo vamos a usar mucho después de esto hemos vuelto al archivo bar etc bar environment para Añadir un par de cosas en primer lugar

- [13:36](#)

hemos añadido al pth las rutas de la carpeta Bin y sbin de jup para hacer accesibles todos sus comandos desde cualquier punto del sistema y hemos añadido una nueva variable global llamada hado home en la que hemos puesto la ruta base donde están los archivos de hadoop por último he configurado ese segundo disco duro que teníamos para que siempre en cada arranque se monte en una carpeta que se llama disc grande cuando configure hado le indicaré que deje sus archivos dentro de esta carpeta en principio con esto acabamos la parte de instalación

Configurar Hadoop >

- [14:06](#)

vamos ahora con la parte de configuración me voy a ir a la carpeta donde están los archivos de configuración dentro de jup etc hado y al hacer un ls veré aquí todos los archivos de configuración de hadoop nosotros vamos a editar este archivo el coresite Recuerda que hadoop trabaja en una estructura distribuida pero

centralizada es decir hay un nodo principal o que actúa con funciones principales este archivo se el encargado de indicar a todos los nodos Quién es ese nodo principal vamos a editarlo con

- 14:34

Nano coresite.xml aquí bajaremos hasta las etiquetas de configuración y dentro de ellas vamos a Añadir property name y el nombre de la propiedad fs.default.file System Cerramos el nombre e indicamos el nombre del nodo y el puerto donde se alojará el name node nos aseguramos que está todo bien guardamos y salimos para comprobar si está todo bien Podemos ejecutar el Comando `jup conf test` que nos indicará si los archivos de configuración están sintácticamente correctamente contruidos aquí tenemos a coresite y parece que sí ojo que esto no garantiza

- 15:10

que esté todo bien me he podido equivocar al escribir el nombre de la propiedad por ejemplo o un valor incorrecto este tipo de comprobaciones semánticas no las hace esta aplicación como dijimos antes este archivo es común a todos los nodos entonces tendría que replicarlo en todos los nodos podría escribirlos en cada uno de ellos o directamente los puedo copiar desde aquí con el comando `scp` origen y destino podemos copiar a través de ssh Así que le digo que me copie este archivo en equipo 02 en la ruta `home administrador`

- 15:39

`jup etc jup` Parece que bien lo copio en el tres y lo copio en el 4 vamos a comprobar si está todo bien voy a hacer aquí un `Cat hup etc hup coresite` y aquí me aparecen los datos que acabo de configurar en el uno todo correcto en este momento de la configuración todos los nodos de mi claser sabrían Quién es el principal simplemente consultando el archivo coresite que ya todos tienen actualizado vamos a configurar ahora el principal con la función de `name node principal` va a ser el uno que es el que estamos configurando y para ello vamos a

- 16:12

tener que editar el archivo `hdfs site.xml` igual que en el anterior dentro de la parte de configuración vamos a Añadir valores a varias propiedades en esta primera propiedad le vamos a indicar al `name node` donde puede dejar sus archivos es importante cambiar esta propiedad Porque por defecto esto nos está apuntando a una carpeta temporal que no es persistente en cada inicio del ordenador se borra con lo cual perderíamos todos nuestros datos si nos fijamos en la ruta vemos Que `home administrador disco grande` ya existe ya

- [16:42](#)

la he creado es el segundo disco duro ese que tengo pero dentro de esta carpeta voy a crear una subcarpeta llamada name node esto es habitual porque muchas veces en los nodos se Ejecutan varias funciones un mismo nodo podría ser name node y Data node para que no se mezclen sus datos vamos a crear una carpeta para cada uno ahora en cuanto acabe de editar este archivo crearé esta carpeta la segunda propiedad que voy a retocar es la del número de réplicas por defecto son tres réplicas las que va a hacer hdfs pero yo lo voy a

- [17:11](#)

bajar a dos porque lo tengo todo virtualizado y no quiero desperdiciar recursos con dos copias de cada bloque Tengo suficiente estas dos siguientes propiedades tienen que ver con el tiempo de detección en el que se detecta la falta de un nodo y se declara como nodo muerto Recuerda que este era un parámetro relativamente sensible porque cuando un nodo se declara como muerto se desencadena un proceso de copia de emergencia para mantener el número de réplicas de todos los datos con estos datos 1 y 500 mantenemos el tiempo de

- [17:38](#)

declaración de nodos muertos en unos 10 segundos con la propiedad Host lo que hacemos Es declarar en un archivo todos los nodos que componen el clúster el valor de esta propiedad contiene la ruta que apunta al archivo donde está esta información habitualmente usamos el archivo workers podríamos elegir cualquier otro archivo para guardar esta información pero workers es un un nombre adecuado porque es un archivo que ya viene dentro de los archivos de configuración de hado y además es usado por muchos otros scripts dentro de hado

- [18:06](#)

por último en esta propiedad indicamos la ruta del archivo que contiene los nodos que van a ser decomisados es decir aquellos nodos que por alguna razón van a estar fuera del sistema durante un tiempo gracias a estos informamos a hado de que van a ser decomisados en el momento que se declaren como muertos hado no desencadenará ninguna copia de bloques porque ya la ha hecho antes Recuerda que decomisar un nodo es indicarle a hado que ese nodo va a ser apagado por cuestiones de mantenimiento o de lo que sea y que por lo tanto

- [18:35](#)

aparecerá como muerto en algún momento Gracias a que estamos informando con antelación hodo podrá hacer las copias de los bloques que están en ese nodo y no usarlo más de esta manera cuando el técnico lo apague el número de réplicas dentro del clúster se mantiene fijo cuando el técnico acabe y lo encienda ese nodo volverá a la vida lo eliminaremos del archivo este de decomisados y volverá a formar parte del clúster como un nodo más igual que antes tenía tenemos un archivo preferido el archivo workers para indicar qué

- [19:02](#)

nodos forman parte del clúster para los nodos decomisados no hay ningún archivo en especial con lo cual necesitaremos crear uno Yo le voy a llamar simplemente de comisados con esto tendríamos el archivo de configuración habría que repasar si todas las etiquetas están bien abiertas y bien cerradas Pero eso lo comprobaremos después con la utilidad de hup antes de salir voy a recordar qué es lo que tenía que hacer en primer lugar tengo que crear dentro de disco grande una carpeta que se llame name node porque le estoy diciendo que ahí es

- [19:26](#)

donde va a dejar sus archivos relacionados después también tengo que incluir en el archivo workers a todos los esclavos Y por último tengo que crear un archivo vacío llamado decomisados bien pues guardo salgo y voy a hacer una comprobación de si los archivos están bien escritos como vemos en el archivo hdfs site ha encontrado una etiqueta de cierre de configuración cuando debería ha una etiqueta de cierre de property vamos a editarlo otra vez a ver qué ha pasado y aquí está el problema Esta etiqueta no se cierra la

- [19:56](#)

cerramos guardamos salimos y volvemos a comprobar ahora ya sí nos dice que sintácticamente está correcto bien Vamos a crear entonces la carpeta name node dentro de disco grande y lo comprobamos con un ls en disco grande y ahí tenemos la carpeta vamos a editar el archivo workers y Aquí vamos a indicar los esclavos con lo cual borramos local Host y escribimos los nombres o las ips de los nodos Esclavos todos aquellos que van a ser datanodes fijaros que yo no estoy incluyendo el equipo uno porque quiero que el equipo uno sea

- [20:28](#)

exclusivamente nam node si quisiéramos que el equipo uno además de name node también ejerciera las labores de datanode tendríamos que añadirlo en esta lista no es el caso así que guardo y salgo y aún me queda pendiente crear el archivo de

comisados esto lo hago con el comando touch de comisados que me crea este archivo sin ningún dato si lo hago un ls tengo el archivo de comisados y workers ya lo he editado con esto el name node ya está configurado el siguiente paso sería formatearlo esto lo hacemos con el comando hdfs name node

- [21:01](#)

format pulsamos intro y analizamos la salida Si no pone que el nodo ha sido correctamente formateado Entonces es que algo ha fallado tendríamos que comprobar si la semántica del archivo de configuración está bien Es decir si estamos apuntando a las rutas correctas o hay algún problema de permisos en las carpetas bien en este punto el name node ya está configurado podría arrancarlo y ver los datos del clúster vacíos como lo vimos en la práctica lo haremos luego ahora voy a configurar los datanodes me voy a uno de ellos y edito el archivo de

- [21:32](#)

configuración jup etc jup hdfs site xml veréis que estamos editando el mismo fichero de hecho en muchos tutoriales veréis que en un solo nodo se configura todo el archivo hdfs site y después se copia entre los nodos igual que hicimos antes con el Core site esto se puede hacer sin ningún problema de hecho hasta apece lo más eficiente ya que los nodos buscarán la información en este archivo en función del papel que estén desarrollando si son name node buscarán ciertos nombres de propiedades y si son Data node buscarán otros nombres de

- [22:04](#)

propiedades la única razón por la que se paro los contenidos es para que quede absolutamente claro Qué propiedades pertenecen al name node por ejemplo la replicación y qué propiedades pertenecen a los Data nodes que en nuestro caso exclusivamente va a ser la carpeta donde van a dejar sus datos Bueno vamos a ello dentro de configuración vamos a configurar una nueva propiedad llamada dfs datanode Data dir en esta propiedad lo que le estamos indicando a este Data node Es que la carpeta donde debe dejar sus archivos está en Home administrador

- [22:37](#)

disco grande y dentro de disco grande en una carpeta que se llama datanode guardaré comprobaré que este archivo Está sintácticamente bien escrito y me aseguraré de que la carpeta datanode está creada existe dentro de disco grande guardo al hacer la comprobación veo que tengo un error Así que vuelvo a editar y aquí está el error en value no cierro value bien compruebo todo está bien Ahora me

aseguro de crear la carpeta Ahí está creada y este archivo va a ser Exactamente igual en todos los datanodes con lo cual voy a aprovechar para

- [23:09](#)

copiarlo copio en el tres lo copio en el cuatro y lo que sí Tengo que asegurarme Es que la carpeta existe en todos muy bien pues ahora ya sí debería estar funcionando en todos lo que hemos hecho en esta parte de configuración es Modificar el archivo coresite para indicar en una propiedad Quién va a ser el nodo principal este archivo coresite lo hemos copiado a todo todos los nodos después en el name node hemos modificado el archivo hdfs site solo con las propiedades que va a tener el name node hemos actualizado el archivo workers con

- [23:39](#)

todos los nodos esclavos y hemos creado un archivo de comisados que por ahora no vamos a utilizar pero si en algún momento tenemos que dar alguno de baja lo pasaremos primero por ahí después formateamos el name node hemos creado la tabla de ficheros dentro del name node a continuación lo que hicimos es en cada uno de los datanodes editar el archivo de configuración HD fs site solo con las propiedades de los Data nodes lo hicimos en uno y lo copiamos en el resto de los Data nodes la única propiedad que modificamos fue la ruta donde los

- [24:08](#)

datanodes dejarán sus datos esa ruta apunta a una carpeta que se llama datanode dentro de disco grande que ha sido necesario crearlo en cada uno de los nodos con esto ya lo tenemos todo listo podríamos levantar el servicio name node y todos los datanodes cada uno en su nodo con el comando que ya habíamos visto durante las prácticas pero gracias a la propiedad de Host y tener los nodos esclavos configurados en el archivo workers ya podemos hacerlo desde el name node con uno de los scripts que tiene por ejemplo Start dfs

- [24:37](#)

sh este Script va a consultar al corite Quién es el nam node y lanzará la función nam node en ese nodo además consultará en el archivo workers Quiénes son los esclavos se conectará a través de ssh y lanzará allí la función de Data node este Warning solo me sale la primera vez porque me dice que la ruta de los logs no existe que la va a crear con lo cual ya nunca más me va a salir y en principio si no hay señales de error todo está funcionando para comprobar que funciona bien abrimos un navegador y nos conectamos a local Host Puerto 9870 Esta

- [25:08](#)

es la interfaz web Y si nos vamos a la pestaña de datanodes aquí abajo tenemos nuestros tres datanodes configurados correctamente ya tenemos nuestra capa hdfs funcionando hasta aquí Este vídeo