**Instalación de Hadoop**

**Instalación de Hadoop en su versión SingleNode**

**Creado**: 16 Junio

**Última modificación**: 16 Junio

Para la instalación de Hadoop se ha basado en una instalación realizada en el portal Edureka [1], contrastando la información con la página oficial de Hadoop [2] al igual que con el tutorial disponible en DigitalOcean [3].

**Paso 1.** Se ha descargado la última versión estable de Hadoop, desde la página de lanzamientos de Hadoop [4]

**¿Cómo?**

**1.1:** Acceder a la página de los lanzamientos de Hadoop.

**1.2:** Elegir el enlace llamado “binary” en la fila “2.8.0” y la columna “Tarball”.

**1.3:** Seleccionar el link recomendado de descarga.

**Paso 2.** Se ha descomprimido el fichero descargado

**¿Cómo?**

**2.1:** tar -xvf hadoop-2.8.0.tar.gz

Una vez que Hadoop ha sido descomprimido, hay 3 importantes pasos que se deben seguir para montar el cluster SingleNode:

* El primero de ellos es completar los ficheros de configuración de Hadoop. En nuestro caso, se deben modificar:
  + **core-site.xml** – Informa al demonio de Hadoop donde se ejecuta el NameNode en el cluster. Contiene los ajustes de configuración para el Core de Hadoop como por ejemplo ajustes de entrada/salida que son comunes a HDFS y MapReduce.
  + **hadoop-env.sh –** Ofrece parámetros customizables para cada uno de los servidores. Este fichero almacena las variables de entorno relacionadas con Hadoop como el CLASSPATH o el JAVA\_HOME.
  + **hdfs-site.xml –** Contiene los ajustes de configuración para los demonios correspondientes a HDFS: el NameNode, el NameNode Secundario y los DataNodes. Aquí se puede configurar HDFS para especificarle la replicación de bloques por defecto y las comprobaciones de permisos. Una propiedad muy importante en este fichero es *dfs.data.dir*, que especifica los directorios donde el DataNode almacena los bloques. Cada bloque se almacena en unicamente uno de estos directorios. Otra importante propiedad es *fs.checkpoint.di,* que especifica los directorios donde el NameNode secundario almacena checkpoints (copias de seguridad). Almacena una copia del checkpoint en cada directorio de la lista.
  + **mapred-site.xml.template -** Contiene los ajustes de configuración para los demonios del MapReduce, para el *job tracker* y los *task-trackers*. Una de las propiedades más interesantes de este fichero es *mapred.job.tracker*, donde se configuran el hostname y el puerto del Job Tracker. Si se deja el valor por defecto, el job tracker se ejecuta bajo demanda cuando el cliente ejecute algún trabajo de MapReduce. Otra propiedad interesante es *mapred.local.dir* que indica el/los directorios donde MapReduce almacena datos intermedios para los jobs. Los datos son borrados tras la ejecución de la tarea.
    - El **job tracker** es el servicio de Hadoop que recibe las tareas tareas de MapReduce del cliente y se comunica con el NameNode para determinar la localización de los datos dentro del cluster. Normalmente se ejecuta en un nodo separado y se encarga de encontrar los mejores nodos para ejecutar las tareas basándose en la localización de los datos (proximidad) y los slots disponibles para ejecutar una tarea en determinado nodo. Cuando el Job Tracker esté caído, HDFS seguirá funcionando pero la funcionalidad relacionada con MapReduce no podrá ser realizada.
  + Los **task-trackers** son procesos que se ejecutan en los DataNodes. Se encargan de recibir las tareas de Mapper y Reducer desde el Job Tracker y ejecutarlas. Además, están en comunicación contínua con el Job Tracker para mantenerle informado del progreso de la ejecución de dichas tareas. Si un task-tracker falla, no es considerado un fallo grave, puesto que el Job Tracker se encargará de asignar las tareas a otro Task Tracker de otro nodo.
  + **yarn-site.xml –** Contiene los ajustes de configuración para Yarn. La información contenida sobrescribe los valores por defecto para los parámetros de Yarn.
* El segundo es fijar las variables de entorno tales como el Home de Hadoop o la ruta a Java.
* El ultimo paso es formatear el Namenode. Después seremos capaces de arrancar los deamons de Hadoop.

**Paso 3.** Se han completado los ficheros de configuración de Hadoop

**¿Cómo?**

**3.1:** cd a la carpeta descomprimida de hadoop-2.8.0 (a partir de ahora nos referiremos a esta carpeta como **HADOOP\_HOME**)

**3.2:** *cd etc/hadoop –* Aquí se almacenan los ficheros de configuración que vienen por defecto con Hadoop.

**3.3:** Se ha editado el fichero *core-site.xml* para insertar la propiedad que indica a Hadoop el nombre por defecto del sistema de ficheros:

*<property>*

*<name>fs.default.name</name>*

*<value>hdfs://localhost:9001</value>*

*</property>*

Observamos pues que HDFS está configurado para ejecutarse en local, en el puerto 9001.

**3.4:** Se ha editado el fichero *hdfs-site.xml* y se han añadido las siguientes propiedades:

* ***dfs.replication –*** Número de bloques a replicar por defecto. Si esta propiedad no existe, se establecen por defecto 3 bloques de replicación. Nosotros lo hemos puesto a 1.
* ***dfs.permission –*** Habilita (true) / Deshabilita (false) la comprobación de permisos en HDFS. Nosotros lo hemos puesto a false.
* ***dfs.name.dir –*** Determina el directorio donde se va a almacenar la tabla del NameNode en el sistema de ficheros local. Si se le da una lista de directorios, replica la tabla en todos los directorios, para conseguir redundancia.
* ***dfs.data.dir -*** Determina el directorio donde se van a almacenar los bloques del DataNode en el sistema de ficheros local. Si se le da una lista de directorios, almacena los bloques en todos los directorios.

*<property>*

*<name>dfs.replication</name>*

*<value>1</value>*

*</property>*

*<property>*

*<name>dfs.permission</name>*

*<value>false</value>*

*</property>*

*<property>*

*<name>dfs.name.dir</name>*

*<value>****HADOOP\_HOME****/hadoop2\_data/hdfs/name/data</value>*

*<final>true</final>*

*</property>*

*<property>*

*<name>dfs.data.dir</name>*

*<value>****HADOOP\_HOME****/hadoop2\_data/hdfs/name</value>*

*<final>true</final>*

*</property>*

**3.5:** Se ha editado el fichero yarn*-site.xml* y se han introducido algunas propiedades para arrancar los servicios auxiliares del nodemanager de yarn.

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class </name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

**3.6:** Se ha hecho una copia del mapred-site.xml.template en otro fichero en la misma carpeta llamado mapred-site.xml. Allí será donde añadiremos el framework de MapReduce, en nuestro caso YARN:

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

**3.7:** Se ha editado el fichero hadoop-env.sh para introducir correctamente la ruta a JAVA. Así pues, se ha reemplazado la línea de *export JAVA\_HOME = {{JAVA\_HOME}}* con la ruta real de JAVA.

**Paso 3.** Se han completado los ficheros de configuración de Hadoop

**¿Cómo?**

**3.1:** cd a la carpeta descomprimida de hadoop-2.8.0 (a partir de ahora nos referiremos a esta carpeta como **HADOOP\_HOME**)

**3.2: *cd etc/hadoop –* Aquí se almacenan los ficheros de configuración que vienen por defecto con Hadoop.**

[1] Portal Edureka: <https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster?utm_source=youtube&utm_campaign=hadoop-installation-single-node-051216-wr&utm_medium=description>

[2] Página oficial de Hadoop: <http://hadoop.apache.org/>

[3] Portal DigitalOcean: <https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-16-04>

[4] Hadoop Releases: <http://hadoop.apache.org/releases.html>