

# Instalación Hive

## 1. Prerrequisitos

Antes de comenzar la instalación de Hive, asegúrate de cumplir con los siguientes requisitos:

1. **Hadoop instalado y configurado:** Hive depende de Hadoop para la ejecución de trabajos.
2. **MySQL instalado:** MySQL será usado como el metastore para almacenar metadatos de Hive.
3. **Java instalado:** Hive requiere Java para su funcionamiento. Puedes verificar la instalación de Java ejecutando:

```
java -version
```

## 2. Descargar e Instalar Apache Hive

Descargar la última versión estable de Apache Hive:

Ve al [sitio oficial de descargas de Hive](#) y descarga la versión más reciente.

```
wget https://dlcdn.apache.org/hive/hive-4.0.1/apache-hive-4.0.1-bin.tar.gz
tar xzvf apache-hive-4.0.1-bin.tar.gz
mv apache-hive-4.0.1-bin hive
```

Configurar las variables de entorno de Hive:

Añade las siguientes líneas al archivo `.bashrc` o `.profile` para configurar las variables de entorno de Hive:

```
export HIVE_HOME=/home/hadoop/hive
export PATH=$PATH:$HIVE_HOME/bin
```

Luego, recarga el archivo:

```
source ~/.bashrc
```

### 3. Descargar e Instalar el Conector JDBC de MySQL

#### Descargar MySQL Connector/J

Ve a <https://dev.mysql.com/downloads/connector/j/> y descarga el archivo correspondiente para tu sistema. Después de descargarlo, extrae el archivo:

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j_9.1.0-1ubuntu24.04_all.deb
sudo dpkg -i mysql-connector-j_9.1.0-1ubuntu24.04_all.deb
sudo apt-get install -f
dpkg -L mysql-connector-j
```

#### Copiar el archivo JAR a Hive:

Copia el archivo JAR del conector de MySQL al directorio de librerías de Hive para que pueda utilizarlo:

```
cp /usr/share/java/mysql-connector-j-9.1.0.jar $HIVE_HOME/lib/
cp /usr/share/java/mysql-connector-java-9.1.0.jar $HIVE_HOME/lib/
```

### 4. Configurar MySQL como Metastore para Hive

#### Crear una Base de Datos y Usuario para Hive en MySQL

Iniciar sesión en MySQL:

```
sudo mysql -u root -p
```

Crear la base de datos metastore para Hive:

```
CREATE DATABASE metastore;
```

Crear un usuario para Hive y asignar permisos:

```
CREATE USER 'hiveuser'@'localhost' IDENTIFIED BY 'hivepassword';
GRANT ALL PRIVILEGES ON metastore.* TO 'hiveuser'@'localhost';
FLUSH PRIVILEGES;
```

## Configurar hive-site.xml para MySQL

### Editar el archivo hive-site.xml:

Abre el archivo hive-site.xml, que se encuentra en \$HIVE\_HOME/conf/. Si no existe, crea uno copiando el archivo de ejemplo:

```
cp $HIVE_HOME/conf/hive-default.xml.template $HIVE_HOME/conf/hive-site.xml
```

### Añadir la configuración del metastore y las variables del sistema:

```
<!-- Configuración del metastore en MySQL -->
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</value>
  <description>JDBC connect string for a JDBC metastore</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.cj.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hiveuser</value>
  <description>Username to use against metastore database</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>hivepassword</value>
  <description>Password to use against metastore database</description>
</property>

<!-- Solución para las variables del sistema -->
<property>
  <name>system:java.io.tmpdir</name>
  <value>/tmp/hive/java</value>
</property>

<property>
  <name>system:user.name</name>
  <value>${user.name}</value>
</property>

<!-- Propiedades de DataNucleus -->
<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>>false</value>
  <description>Should the datanucleus automatically create the
schema?</description>
</property>

<property>
  <name>datanucleus.fixedDatastore</name>
  <value>>true</value>
```

```
<description>Flag for datanucleus to use a fixed datastore</description>
</property>

<property>
  <name>datanucleus.autoCreateTables</name>
  <value>false</value>
  <description>Flag for datanucleus to automatically create tables</description>
</property>
```

## 5. Inicializar el Esquema de Hive en MySQL

Ejecutar el comando schematool para inicializar el esquema:

Ahora que el metastore está configurado, debes inicializar el esquema en MySQL:

```
schematool -initSchema -dbType mysql --verbose
```

## 6. Configurar Impersonación en Hive

Configurar Impersonación en Hive

Editar el archivo hive-site.xml:

Asegúrate de que la siguiente propiedad esté presente en hive-site.xml para habilitar la impersonación en HiveServer:

```
<property>
  <name>hive.server2.enable.doAs</name>
  <value>true</value>
  <description>Permitir que HiveServer2 actúe en nombre del usuario que
envía consultas</description>
</property>
```

## 7. Iniciar HiveServer y Conectarse con Beeline

Iniciar HiveServer:

```
hiveserver2 &
```

Conectarse a Hive usando Beeline:

```
beeline -u jdbc:hive2://localhost:10000 -n hadoop
```

## 8. Servicio de Monitoreo en el Puerto 10002

Hive también inicia un **servicio de monitoreo web** en el puerto 10002. Este servicio proporciona una interfaz para supervisar el estado y las estadísticas de HiveServer2, incluyendo detalles sobre las sesiones, las consultas activas, y el rendimiento del servidor.

### Acceder al servicio de monitoreo:

Abre tu navegador y ve a:

<http://nodomaster:10002>

Aquí puedes ver información detallada sobre las conexiones activas y el estado del servidor. Es útil para supervisar el rendimiento y el uso de HiveServer2.

## 9. Crea tabla de prueba

Conéctate mediante:

```
beeline -u jdbc:hive2://localhost:10000 -n hadoop
```

Aquí podrás escribir comandos HSQL

```
CREATE TABLE employees ( id INT, name STRING, salary FLOAT );
INSERT INTO employees VALUES (1, 'Alice', 50000.0);
INSERT INTO employees VALUES (2, 'Bob', 60000.0);
INSERT INTO employees VALUES (3, 'Charlie', 55000.0);
INSERT INTO employees VALUES (4, 'Diana', 70000.0);
INSERT INTO employees VALUES (5, 'Eve', 72000.0);
INSERT INTO employees VALUES (6, 'Frank', 48000.0);
INSERT INTO employees VALUES (7, 'Grace', 64000.0);
INSERT INTO employees VALUES (8, 'Hank', 58000.0);
INSERT INTO employees VALUES (9, 'Ivy', 53000.0);
INSERT INTO employees VALUES (10, 'John', 53000.0);
```