

Домашнее задание №3

БПИ227 Артемьев Александр

Следующая инструкция представляет из себя пошаговое руководство по развертыванию Apache Hive, причем с возможностью использования более чем одним клиентом. После выполнения второго дз у нас уже есть развернутый на 3 нодах hdfs и узлы со следующими адресами:

jn_glob_ip – узел для входа в jn
jn_local_ip – локальный адрес jn
nn_ip – локальный адрес nn
dn-00_ip – локальный адрес dn-00
dn-01_ip – локальный адрес dn-01
user_name – имя пользователя, который может подключиться к jn

А также развернутый на этих узлах YARN, с веб интерфейсом

Порядок действий:

1) Для начала необходимо проверить, что поднятая структура с hadoop и yarn уже успешно функционирует. Например, проверкой логов или веб-интерфейса.

2) В случае задачи с возможностью использования более чем одним клиентом, для Apache Hive необходимо отдельное хранилище, в данном примере мы будем использовать PostgreSQL. Поэтому установим и настроим Postgres:

Подключаемся к серверу

```
ssh <user_name>@<jn_glob_ip>
```

(либо `ssh -L 9870:nn:9870 -L 8088:nn:8088 -L 19888:nn:19888 <user_name>@<jn_glob_ip>`)

Переходим на NameNode

```
ssh nn
```

Устанавливаем postgres

```
sudo apt install postgresql
```

Перейдем на пользователя postgres

```
sudo -i -u postgres
```

После чего уже настраиваем postgres для работы с hive

```
psql
```

Создаем БД для метастора

```
CREATE DATABASE metastore;
```

Создаем пользователя с паролем

```
CREATE USER hive with password 'hmp';
```

Выдаем все привилегии пользователю:

```
GRANT ALL PRIVILEGES ON DATABASE "metastore" to hive;
```

Передаем владения metastore этому же пользователю

```
ALTER DATABASE metastore OWNER TO hive;
```

Выходим из psql и из акк. postgres

```
\q
```

```
exit
```

3) Теперь изменим конфиг postgres чтобы он был доступен не только локально:

```
sudo vim /etc/postgresql/16/main/postgresql.conf
```

Изменяем имя хоста, которое он будет слушать:

```
listen_addresses = 'nn'
```

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
#listen_addresses = 'localhost'      # what IP address(es) to listen on;  
listen_addresses = 'nn'              # comma-separated list of addresses;  
                                     # defaults to 'localhost'; use '*' for all  
                                     # (change requires restart)  
port = 5432                          # (change requires restart)  
max_connections = 100                # (change requires restart)  
#reserved_connections = 0           # (change requires restart)  
#superuser_reserved_connections = 3 # (change requires restart)  
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories  
                                     # (change requires restart)  
#unix_socket_group = ''             # (change requires restart)  
#unix_socket_permissions = 0777    # begin with 0 to use octal notation  
                                     # (change requires restart)  
  
-- INSERT --
```

66,28-41

4%

4) Изменяем конфиг безопасности:

```
sudo vim /etc/postgresql/16/main/pg_hba.conf
```

Добавляем 2 строки в конфиг для разрешения подключений

```
host    metastore    hive    192.168.1.1/32    password  
host    metastore    hive    <jn_local_ip>/32    password
```

5) Перезапускаем postgres

```
sudo systemctl restart postgresql
```

6) Выходим на jn на пользователе <user_name>

```
exit
```

7) Установим клиента postgres для того, чтобы мы могли обращаться к БД

```
sudo apt install postgresql-client-16
```

8) Заходим на аккаунт hadoop

```
sudo -i -u hadoop
```

9) Устанавливаем дистрибутив hive

```
wget https://archive.apache.org/dist/hive/hive-4.0.0-alpha-2/apache-hive-4.0.0-alpha-2-bin.tar.gz
```

10) Распакуем архив с hive

```
tar -xzvf apache-hive-4.0.0-alpha-2-bin.tar.gz
```

11) Установим драйвер postgre для hive

```
cd apache-hive-4.0.0-alpha-2-bin/lib/  
wget https://jdbc.postgresql.org/download/postgresql-42.7.4.jar
```

12) И теперь изменим наши конфиги

```
vim ../conf/hive-site.xml
```

И запишем в него следующее:

```
<configuration>  
  <property>  
    <name>hive.server2.authentication</name>  
    <value>NONE</value>  
  </property>  
  <property>  
    <name>hive.metastore.warehouse.dir</name>  
    <value>/user/hive/warehouse</value>  
  </property>  
  <property>  
    <name>hive.server2.thrift.port</name>  
    <value>5433</value>  
    <description>TCP port number to listenon, default 10000</description>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionURL</name>  
    <value>jdbc:postgresql://nn:5432/metastore</value>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionDriverName</name>  
    <value>org.postgresql.Driver</value>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionUserName</name>  
    <value>hive</value>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionPassword</name>  
    <value>hmp</value>  
  </property>  
</configuration>
```

13) Добавляем переменное окружение в наш профиль

```
vim ~/.profile
```

Добавляем следующие строчки:

```
export HIVE_HOME=/home/hadoop/apache-hive-4.0.0-alpha-2-bin
```

```
export HIVE_CONF_DIR=$HIVE_HOME/conf
export HIVE_AUX_JARS_PATH=$HIVE_HOME/lib/*
export PATH=$PATH:$HIVE_HOME/bin
и применяем переменные окружения
```

```
source ~/.profile
```

14) Создаем директории и раздаем права

```
hdfs dfs -mkdir -p /user/hive/warehouse
hdfs dfs -chmod g+w /user/hive/warehouse
hdfs dfs -chmod g+w /tmp
```

15) Инициализируем БД

```
cd ..
bin/schematool -dbType postgres -initSchema
```

16) Запускаем hive и подключаемся

```
hive --hiveconf hive.server2.enable.doAs=false --hiveconf
hive.security.authorization.enable=false --service hiveserver2 1>> /tmp/hs2.log 2>> /tmp/hs2.log
&
```

```
beeline -u jdbc:hive2://jn:5433 -n scott -p tiger
```

Мои данные:

Env team-1

User team

пароль для входа x=T35T_sMdm4

узел для входа 176.109.91.3

jn 192.168.1.6

nn 192.168.1.7

dn-00 192.168.1.8

dn-01 192.168.1.9