

Hive 配置高可用 hiveserver2

作者: [k8s](#)

原文链接: <https://ld246.com/article/1590594400660>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

分布式环境

继 [CentOS7 安装 Hive 2.3.4](#) 准备分布式环境。

host ervice	cpu	memory	ip
node01 iveserver2	1c	2G	10.4.96.4
node02 iveserver2	1c	2G	10.4.96.5
node03	1c	1G	10.4.96.6
node04	1c	1G	10.4.96.7
node05	1c	1G	10.4.96.8
node06 ookeeper	1c	1G	10.4.96.9
node07 ookeeper	1c	1G	10.4.96.10
node08 ookeeper	1c	1G	10.4.96.11
node09 ysql	1c	1G	10.4.96.12

1. 配置 HDFS

配置 hdfs，达到从 hiveserver2 以任何用户登录均在 hdfs 中代理成 **god** 用户（分布式文件系统中作用户）

向 **core-site.xml** 中添加 **hadoop.proxyuser.god.groups**、**hadoop.proxyuser.god.hosts** 配置

```
[god@node01 ~]$ vim /opt/bigdata/hadoop-2.6.5/etc/hadoop/core-site.xml
```

```
...
<configuration>
  <!--指定namenode所属集群-->
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://mycluster</value>
  </property>
  <!--指定zookeeper服务的集群地址-->
  <property>
    <name>ha.zookeeper.quorum</name>
    <value>node06:2181,node07:2181,node08:2181</value>
  </property>
  <property>
    <name>hadoop.proxyuser.god.groups</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.god.hosts</name>
    <value>*</value>
  </property>
</configuration>
```

```
</property>
</configuration>
```

分发配置文件

```
[god@node01 ~]$ for i in `seq 2 8`;do scp /opt/bigdata/hadoop-2.6.5/etc/hadoop/core-site.xml god@node0$i:/opt/bigdata/hadoop-2.6.5/etc/hadoop/core-site.xml;done
```

在两个 NameNode 节点刷新配置

```
[god@node01 ~]$ hdfs dfsadmin -fs hdfs://node01:8020 -refreshSuperUserGroupsConfiguration
[god@node01 ~]$ hdfs dfsadmin -fs hdfs://node02:8020 -refreshSuperUserGroupsConfiguration
```

对 hdfs 中 /tmp 目录赋予 777 权限

```
[god@node01 ~]$ hdfs dfs -chmod 777 /tmp
```

2. 配置 hive

在 **node01** 和 **node02** 上配置 **hive-site.xml** 文件，其中只有 **hive.server2.thrift.bind.host** 配置同，为各自主机名

```
$ vim /opt/bigdata/hive-2.3.4/conf/hive-site.xml
```

```
...
<configuration>
  <!--指定hive仓库目录-->
  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive_remote/warehouse</value>
  </property>
  <!--指定元数据库MySQL信息，注意XML文件中 & 符号的转义问题-->
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://node09:3306/hive_remote?createDatabaseIfNotExist=true&verifyServerCertificate=false&useSSL=false</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>Az123456_</value>
  </property>
  <!--启用hiveserver2权限管理-->
  <property>
    <name>hive.security.authorization.enabled</name>
    <value>true</value>
```

```

</property>
<property>
  <name>hive.server2.enable.doAs</name>
  <value>>false</value>
</property>
<property>
  <name>hive.users.in.admin.role</name>
  <value>god</value>
</property>
<property>
  <name>hive.security.authorization.manager</name>
  <value>org.apache.hadoop.hive.ql.security.authorization.plugin.sqlstd.SQLStdHiveAuthor
zerFactory</value>
</property>
<property>
  <name>hive.security.authenticator.manager</name>
  <value>org.apache.hadoop.hive.ql.security.SessionStateUserAuthenticator</value>
</property>
<!--配置hiveserver2高可用-->
<property>
  <name>hive.server2.support.dynamic.service.discovery</name>
  <value>>true</value>
</property>
<property>
  <name>hive.server2.zookeeper.namespace</name>
  <value>hiveserver2_zk</value>
</property>
<property>
  <name>hive.zookeeper.quorum</name>
  <value>node06:2181,node07:2181,node08:2181</value>
</property>
<property>
  <name>hive.zookeeper.client.port</name>
  <value>2181</value>
</property>
<property>
  <name>hive.server2.thrift.bind.host</name>
  <value>node01/node02</value>
</property>
<property>
  <name>hive.server2.thrift.port</name>
  <value>10001</value>
</property>
</configuration>

```

3. 启动 hiveserver2

```

[god@node01 ~]$ screen -S hiveserver2
[god@node01 ~]$ hive --service hiveserver2

```

```

[god@node02 ~]$ screen -S hiveserver2
[god@node02 ~]$ hive --service hiveserver2

```

两者皆为阻塞界面，按 **Ctrl + A + D** 非中断退出

zkCli 界面:

```
[zk: localhost:2181(CONNECTED) 82] ls /hiveserver2_zk  
[serverUri=node02:10001;version=2.3.4;sequence=0000000005]
```

连接测试

```
beeline> !connect jdbc:hive2://node06,node07,node08/test;serviceDiscoveryMode=zooKeeper;  
zooKeeperNamespace=hiveserver2_zk god 123456  
Connecting to jdbc:hive2://node06,node07,node08/test;serviceDiscoveryMode=zooKeeper;  
zooKeeperNamespace=hiveserver2_zk  
19/11/27 22:18:18 [main]: INFO jdbc.HiveConnection: Connected to node02:10001  
Connected to: Apache Hive (version 2.3.4)  
Driver: Hive JDBC (version 2.3.4)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
0: jdbc:hive2://node06,node07,node08/test> show tables;  
+-----+  
| tab_name |  
+-----+  
| logtbl   |  
| psn2     |  
| psn21    |  
| psn22    |  
| psn5     |  
| psn6     |  
| psn7     |  
+-----+  
7 rows selected (2.965 seconds)  
0: jdbc:hive2://node06,node07,node08/test>
```

```
public class HiveJdbcClient2 {  
  
    private static String driverName = "org.apache.hive.jdbc.HiveDriver";  
  
    public static void main(String[] args) throws SQLException {  
        try {  
            Class.forName(driverName);  
        } catch (ClassNotFoundException e) {  
            e.printStackTrace();  
        }  
  
        Connection conn = DriverManager.getConnection("jdbc:hive2://node01,node02,node03  
default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2_zk", "root", "");  
        Statement stmt = conn.createStatement();  
        String sql = "select * from tbl";  
        ResultSet res = stmt.executeQuery(sql);  
        while (res.next()) {  
            System.out.println(res.getString(1));  
        }  
    }  
}
```