目录

[1. 实验一：Hadoop全分布部署 2](#_Toc10883)

[1.1. 实验目的 2](#_Toc7640)

[1.2. 实验要求 2](#_Toc15198)

[1.3. 实验环境 2](#_Toc11881)

[1.4. 实验过程 2](#_Toc7141)

[1.4.1. 实验任务一：Hadoop集群验证 2](#_Toc17224)

[2. 实验二：Sqoop组件部署 4](#_Toc29320)

[2.1. 实验目的 4](#_Toc21956)

[2.2. 实验要求 5](#_Toc30090)

[2.3. 实验环境 5](#_Toc9336)

[2.4. 实验过程 5](#_Toc27872)

[2.4.1. 实验任务一：Sqoop数据传输验证 5](#_Toc21570)

[3. 实验二：Hive组件部署 7](#_Toc23489)

[3.1. 实验目的 7](#_Toc19974)

[3.2. 实验要求 7](#_Toc22496)

[3.3. 实验环境 7](#_Toc26498)

[3.4. 试验过程 7](#_Toc886)

[3.4.1. 实验任务一：Hive组件验证 7](#_Toc6076)

# 实验一：Hadoop全分布部署

## 实验目的

完成本实验，您应该能够：

* 掌握Hadoop全分布部署
* 掌握Hive组件部署

## 实验要求

* 熟悉常用Linux操作系统命令
* 熟悉Hadoop操作命令
* 了解Hadoop、Hive的作用

## 实验环境

本实验所需之主要资源环境如表1-1所示。

表1-1 资源环境

|  |  |
| --- | --- |
| 服务器集群 | 3个以上节点，节点间网络互通，各节点最低配置：双核CPU、8GB内存、100G硬盘 |
| 运行环境 | CentOS 7.4 |
| 大数据平台 | H3C教学与实践管理平台 |
| 服务和组件 | 完成前面章节的实验，其他服务及组件根据实验需求安装 |

## 实验过程

### 实验任务一：Hadoop集群验证

分布式集群搭建完成后，根据Hadoop两大核心组成，可以通过监测这HDFS分布式文件系统和MapReduce来完成监测工作，通过以下步骤完成Hadoop集群测试：

（1）初始化集群，使用Hadoop命令启动集群。

（2）使用Hadoop命令，创建HDFS文件夹。

（3）使用HDFS命令查看文件系统“/”路径下是否存在文件。

（4）调用Hadoop自带的WordCount程序去测试MapReduce，查看控制台是否能正确统计单词数量。

#### 步骤一：具体测试内容

Hadoop自身提供了许多功能用于监测，通过这些监控功能，可以判断出平台是否已经完成全分布式集群搭建，以下步骤监测集群是否搭建成功：

（1）使用JSP查看各个节点启动的进程情况，都启动成功说明系统启动正常。

master（主节点）启动情况，命令如下：

[hadoop@master ~]# jps

**5825 NameNode**

**6021 SecondaryNameNode**

**6453 Jps**

**6169 ResourceManager**

slave1（从节点）节点启动情况，命令如下：

[hadoop@slave1 ~]# jps

**2928 Jps**

**2640 NodeManager**

**2542 DataNode**

slave2（从节点）节点启动情况，命令如下：

[hadoop@slave2 ~]# jps

**2673 DataNode**

**2771 NodeManager**

**3036 Jps**

（2）查看Hadoop的Web监控页面。

使用浏览器浏览主节点机http://master:50070,能查看NameNode节点状态说明系统启动正常，结果如图16-2所示：

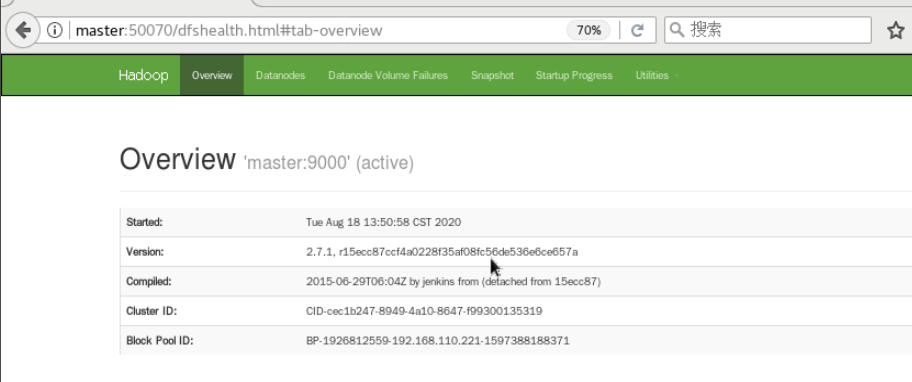


图16-2 NameNode节点状态

使用浏览器浏览master节点http://master:8088，查看所有应用说明系统启动正常，结果如图16-3所示：

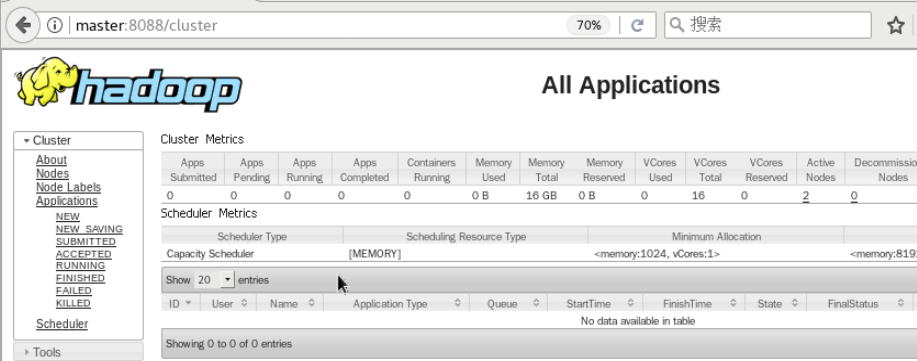


图16-3 Hadoop集群Cluster界面

浏览Nodes说明系统启动正常，结果如图16-4所示：

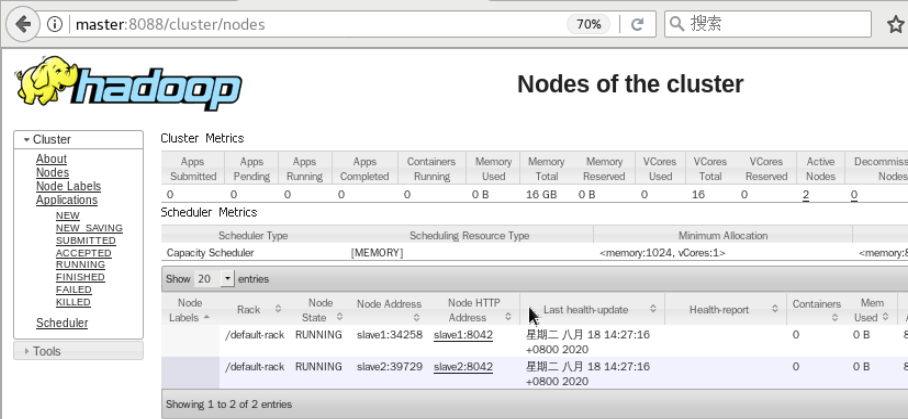


图16-4 Hadoop集群Nodes界面

（3）使用Hadoop命令关闭集群。

使用命令关闭Hadoop集群，返回信息如下，说明系统关闭正常：

[hadoop@master ~]# stop -all.sh

**This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh**

**Stopping namenodes on [master]**

**master: no namenode to stop**

**202.106.155.58: datanode to stop**

**202.106.155.59: datanode to stop**

**Stopping secondary namenodes [0.0.0.0]**

**0.0.0.0: secondarynamenode to stop**

**stopping yarn daemons**

**resourcemanager to stop**

**202.106.155.58: nodemanager to stop**

**202.106.155.59: nodemanager to stop**

# 实验二：Sqoop组件部署

## 实验目的

完成本实验，您应该能够：

* 掌握Sqoop组件部署

## 实验要求

* 熟悉常用Linux操作系统命令
* 熟悉Hadoop操作命令
* 了解Sqoop的作用

## 实验环境

本实验所需之主要资源环境如表1-1所示。

表1-1 资源环境

|  |  |
| --- | --- |
| 服务器集群 | 3个以上节点，节点间网络互通，各节点最低配置：双核CPU、8GB内存、100G硬盘 |
| 运行环境 | CentOS 7.4 |
| 大数据平台 | H3C教学与实践管理平台 |
| 服务和组件 | 完成前面章节的实验，其他服务及组件根据实验需求安装 |

## 实验过程

### 实验任务一：Sqoop数据传输验证

#### 步骤一：查看Sqoop版本

通过Sqoop相关命令能够查询到Sqoop版本号为1.4.7，则表示Sqoop部署成功，部署成功信息如下：

**Please set $HCAT\_HOME to the root of your HCatalog installation.**

**Please set $ACCUMULO\_HOME to the root of your Accumulo installation.**

**20/04/30 14:30:16 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7**

**Sqoop 1.4.7**

**git commit id 2328971411f57f0cb683dfb79d19d4d19d185dd8**

**Compiled by maugli on Thu Dec 21 15:59:58 STD 2017**

#### 步骤二：Sqoop连接MySQL数据库

Sqoop需要启动Hadoop集群，通过Sqoop连接MySQL查看数据库列表判断是否安装成功，输入命令查看控制台最后输出是否为MySQL中的数据库。

[hadoop@master ~]# sqoop list-databases --connect jdbc:mysql://127.0.0.1:3306/ --username root -P

**Warning: /opt/sofeware/sqoop/../hcatalog does not exist! HCatalog jobs will fail.**

**Please set $HCAT\_HOME to the root of your HCatalog installation.**

**Warning: /opt/sofeware/sqoop/../accumulo does not exist! Accumulo imports will fail.**

**Please set $ACCUMULO\_HOME to the root of your Accumulo installation.**

**Warning: /opt/sofeware/sqoop/../zookeeper does not exist! Accumulo imports will fail.**

**Please set $ZOOKEEPER\_HOME to the root of your Zookeeper installation.**

**19/04/22 18:54:10 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7**

**Enter password:**

**19/04/22 18:54:14 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.**

**information\_schema**

**hive**

**mysql**

**performance\_schema**

**sys**

#### 步骤三：Sqoop将HDFS数据导入到MySQL

（1）使用Sqoop命令，将HDFS中“/user/test”数据导入到MySQL中（其中HDFS数据源已经上传到大数据平台，MySQL相关数据库、数据表在第九章已经创建）。

删除原数据库student表的数据

[hadoop@master hadoop]$ mysql -u root -p

**Enter password:**

mysql> use sample ;

**Reading table information for completion of table and column names**

**You can turn off this feature to get a quicker startup with -A**

**Database changed**

mysql> select \* from student;

**+--------+----------+**

**| number | name |**

**+--------+----------+**

**| 1 | zhangsan |**

**| 2 | lisi |**

**| 3 | wangwu |**

**| 4 | sss |**

**| 5 | ss2 |**

**| 6 | ss3 |**

**+--------+----------+**

**6 rows in set (0.00 sec)**

mysql> delete from student;

**Query OK, 6 rows affected (0.00 sec)**

mysql> select \* from student;

**Empty set (0.00 sec)**

mysql> exit

**Bye**

将HDFS数据导入到MySQL

[hadoop@master ~]$ sqoop export --connect "jdbc:mysql://master:3306/sample?useUnicode=true&characterEncoding=utf-8" --username root --password Password123$ --table student --input-fields-terminated-by ',' --export-dir /user/test

**Warning: /opt/sofeware/sqoop/../hcatalog does not exist! HCatalog jobs will fail.**

**Please set $HCAT\_HOME to the root of your HCatalog installation.**

**Warning: /opt/sofeware/sqoop/../accumulo does not exist! Accumulo imports will fail.**

**Please set $ACCUMULO\_HOME to the root of your Accumulo installation.**

**Warning: /opt/sofeware/sqoop/../zookeeper does not exist! Accumulo imports will fail.**

**Please set $ZOOKEEPER\_HOME to the root of your Zookeeper installation.**

**20/05/29 23:54:10 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7**

**20/05/29 23:55:01 INFO mapreduce.Job: Running job: job\_1553997792849\_0005**

**20/05/29 23:54:12 INFO mapreduce.Job: Job job\_1553997792849\_0005 running in uber mode : false**

**20/05/29 23:54:12 INFO mapreduce.Job:  map 0% reduce 0%**

**20/05/29 23:54:25 INFO mapreduce.Job:  map 100% reduce 0%**

**20/05/29 23:54: INFO mapreduce.Job: Task Id : attempt\_1553997792849\_0005\_m\_000000\_0, Status : FAILED**

**AttemptID:attempt\_1553997792849\_0005\_m\_000000\_0 Timed out after 600 secs**

**20/05/29 23:54: INFO mapreduce.Job:  map 0% reduce 0%**

**20/05/29 23:54: INFO mapreduce.Job: map 100% reduce 0%**

（2）通过MySQL命令,查看数据是否成功导入MySQL

[hadoop@master ~]# mysql -u root -pPassword123$

mysql> use sample;

**Reading table information for completion of table and column names**

**You can turn off this feature to get a quicker startup with -A**

**Database changed**

mysql> select \* from student;

**+--------+----------+**

**| number | name |**

**+--------+----------+**

**| 1 | zhangsan |**

**| 2 | lisi |**

**| 3 | wangwu |**

**| 4 | sss |**

**| 5 | ss2 |**

**| 6 | ss3 |**

**+--------+----------+**

**6 rows in set (0.00 sec)**

# 实验二：Hive组件部署

## 实验目的

完成本实验，您应该能够：

* 掌握Hive组件部署

## 实验要求

* 熟悉常用Linux操作系统命令
* 熟悉Hadoop操作命令
* 了解Hive的作用

## 实验环境

本实验所需之主要资源环境如表1-1所示。

表1-1 资源环境

|  |  |
| --- | --- |
| 服务器集群 | 3个以上节点，节点间网络互通，各节点最低配置：双核CPU、8GB内存、100G硬盘 |
| 运行环境 | CentOS 7.4 |
| 大数据平台 | H3C教学与实践管理系统 |
| 服务和组件 | 完成前面章节的实验，其他服务及组件根据实验需求安装 |

## 试验过程

### 实验任务一：Hive组件验证

#### 步骤一：初始化Hive

启动Hadoop集群的环境下，使用Hive初始化命令查看初始化控制台日志信息。

[hadoop@master ~]# schematool -dbType mysql -initSchema

**SLF4J: Class path contains multiple SLF4J bindings.**

**SLF4J: Found bindings in [jar:file:/opt/hive/lib/hive-jdbc-2.0.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]**

**SLF4J: Found bindings in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.4.1jar!/org/slf4j/impl/StaticLoggerBinder.class[**

**SLF4J: Found bindings in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]**

**SLF4J: See http://www.slf4j.org/codes.html#multip\_bindings for an explanation.**

**SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4LoggerFactory]**

**Metastore connection URL: jdbc:mysql://master:3306/hive?createDatabaseIfNotExist=true$useSSL=false**

**Metastore Connection Driver: com.mysql.jdbc.Driver**

**Metastore:connection User: root**

**Staring metastore schema initalization to 2.0.0**

**Initialzation script hive-schema-2.0.0.mysql.sql**

**Initialzation script completed**

**schemaTool completed**

#### 步骤二：启动Hive

由于Hive是Hadoop生态中的一个组件，只需要测试hive时候能正常启动即可，启动Hadoop环境下通过Hive命令，查看Hive启动信息。

[hadoop@master ~]# hive

**Logging initialized using configuration in jar:file:/opt/hive/lib/hive-common-1.1.0.jar!/hive-log4j.properties**

**SLF4J: Class path contains multiple SLF4J bindings.**

**SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]**

**SLF4J: Found binding in [jar:file:/opt/hive/lib/hive-jdbc-1.1.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]**

**SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.**

**SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]**

hive>