云计算项目——基于Hadoop+Hive搭建的存储服务系统

项目成员：2054079 黄绍华

项目介绍

本次课程项目，设计和实现一个由Hadoop2.8.5搭建的多个节点组成的云服务系统，并提供云系统中的存储服务功能。 基于搭建好的本次Hadoop搭建的云服务系统安装 Hive，为了进行Hive的各种操作模拟，在其上模拟创建了一个2021-2022某超市的各种数据，并进行数据库的设计和表的搭建，进行相应数据处理、筛选,最后呈现出一个相关数据的可视化界面。

Hadoop配置

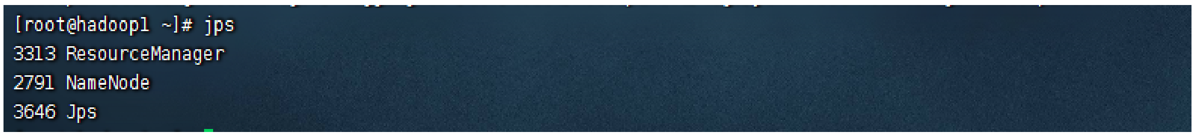
Hadoop结构：

Replication：3块

Block Size: 128MB

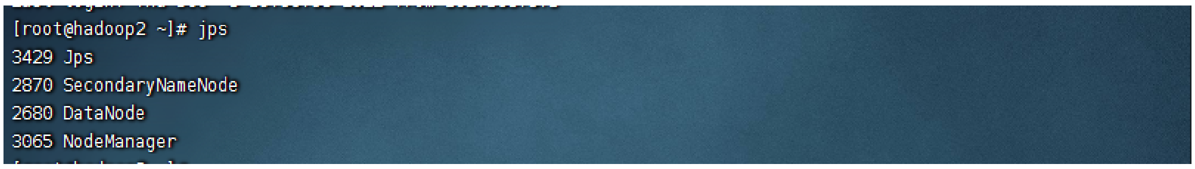
	NameNode	SecondaryNameNode	DataNode	ResourceManager	NodeManager
Node01	✓			✓	
Node02		✓	✓		✓
Node03			✓		✓
Node04			✓		✓

Node01：



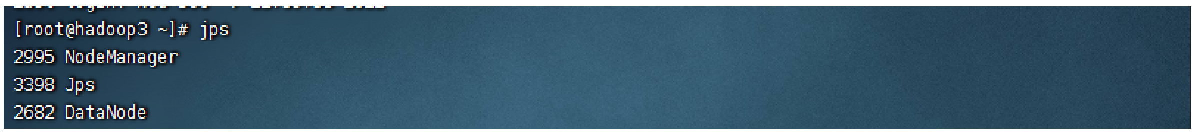
```
[root@hadoop1 ~]# jps
3313 ResourceManager
2791 NameNode
3646 Jps
```

Node02：



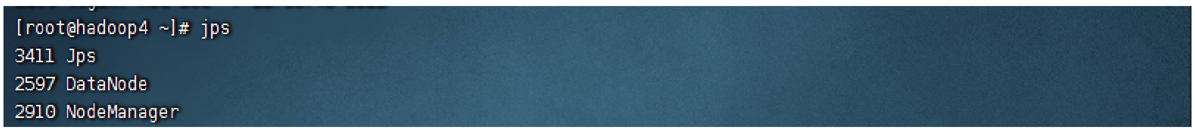
```
[root@hadoop2 ~]# jps
3429 Jps
2870 SecondaryNameNode
2680 DataNode
3065 NodeManager
```

Node03：



```
[root@hadoop3 ~]# jps
2995 NodeManager
3398 Jps
2682 DataNode
```

Node04：

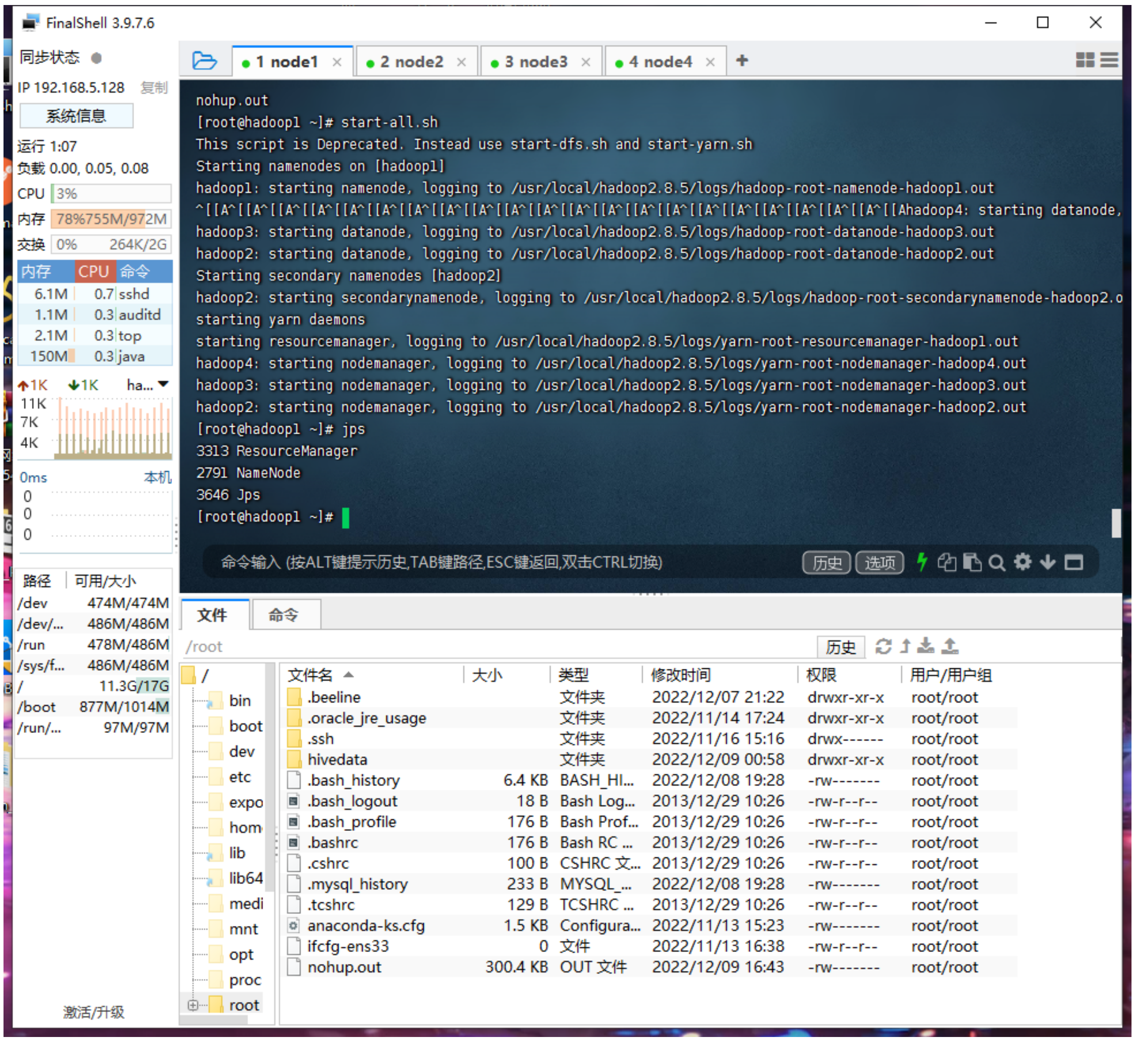


```
[root@hadoop4 ~]# jps
3411 Jps
2597 DataNode
2910 NodeManager
```

搭建工具:

1. FinalShell：ssh连接

   FinalShell是一体化的的服务器,网络管理软件,不仅是ssh客户端,还是功能强大的开发,运维工具,充分满足开发,运维需求。



内存	CPU	命令
6.1M	0.7	sshd
1.1M	0.3	auditd
2.1M	0.3	top
150M	0.3	java

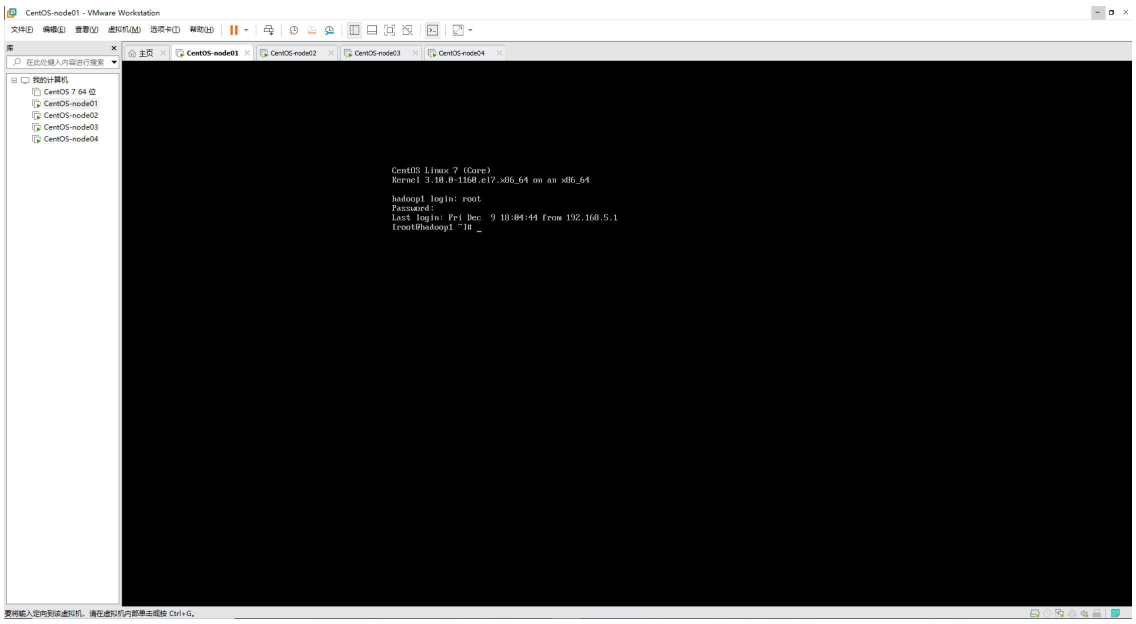
路径	可用/大小
/dev	474M/474M
/dev/...	486M/486M
/run	478M/486M
/sys/f...	486M/486M
/	11.3G/17G
/boot	877M/1014M
/run/...	97M/97M

```
nohup.out
[root@hadoop1 ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [hadoop1]
hadoop1: starting namenode, logging to /usr/local/hadoop2.8.5/logs/hadoop-root-namenode-hadoop1.out
^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[Ahadoop4: starting datanode,
hadoop3: starting datanode, logging to /usr/local/hadoop2.8.5/logs/hadoop-root-datanode-hadoop3.out
hadoop2: starting datanode, logging to /usr/local/hadoop2.8.5/logs/hadoop-root-datanode-hadoop2.out
Starting secondary namenodes [hadoop2]
hadoop2: starting secondarynamenode, logging to /usr/local/hadoop2.8.5/logs/hadoop-root-secondarynamenode-hadoop2.o
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop2.8.5/logs/yarn-root-resourcemanager-hadoop1.out
hadoop4: starting nodemanager, logging to /usr/local/hadoop2.8.5/logs/yarn-root-nodemanager-hadoop4.out
hadoop3: starting nodemanager, logging to /usr/local/hadoop2.8.5/logs/yarn-root-nodemanager-hadoop3.out
hadoop2: starting nodemanager, logging to /usr/local/hadoop2.8.5/logs/yarn-root-nodemanager-hadoop2.out
[root@hadoop1 ~]# jps
3313 ResourceManager
2791 NameNode
3646 Jps
[root@hadoop1 ~]#
```

文件名	大小	类型	修改时间	权限	用户/用户组
.beeline		文件夹	2022/12/07 21:22	drwxr-xr-x	root/root
.oracle_jre_usage		文件夹	2022/11/14 17:24	drwxr-xr-x	root/root
.ssh		文件夹	2022/11/16 15:16	drwx------	root/root
hivedata		文件夹	2022/12/09 00:58	drwxr-xr-x	root/root
.bash_history	6.4 KB	BASH_HI...	2022/12/08 19:28	-rw-------	root/root
.bash_logout	18 B	Bash Log...	2013/12/29 10:26	-rw-r--r--	root/root
.bash_profile	176 B	Bash Prof...	2013/12/29 10:26	-rw-r--r--	root/root
.bashrc	176 B	Bash RC ...	2013/12/29 10:26	-rw-r--r--	root/root
.cshrc	100 B	CSHRC 文...	2013/12/29 10:26	-rw-r--r--	root/root
.mysql_history	233 B	MYSQL_...	2022/12/08 19:28	-rw-------	root/root
.tcshrc	129 B	TCSHRC ...	2013/12/29 10:26	-rw-r--r--	root/root
anaconda-ks.cfg	1.5 KB	Configura...	2022/11/13 15:23	-rw-------	root/root
ifcfg-ens33	0	文件	2022/11/13 16:38	-rw-r--r--	root/root
nohup.out	300.4 KB	OUT 文件	2022/12/09 16:43	-rw-------	root/root

2. VMware Workstation：虚拟机搭建

   VMware Workstation（中文名“威睿工作站”）是一款功能强大的桌面虚拟计算机软件，提供用户可在单一的桌面上同时运行不同的操作系统，和进行开发、测试 、部署新的应用程序的最佳解决方案。VMware Workstation可在一部实体机器上模拟完整的网络环境，以及可便于携带的虚拟机器，其更好的灵活性与先进的技术胜过了市面上其他的虚拟计算机软件。对于企业的 IT开发人员和系统管理员而言， VMware在虚拟网路，实时快照，拖曳共享文件夹，支持 PXE 等方面的特点使它成为必不可少的工具。



```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

hadoop1 login: root
Password:
Last login: Fri Dec  9 18:04:44 from 192.168.5.1
[root@hadoop1 ~]# _
```

3. CentOS：操作系统

   CentOS是免费的、开源的、可以重新分发的开源操作系统，CentOS（Community Enterprise Operating System，中文意思是社区企业操作系统）是Linux发行版之一。

Hadoop平台搭建过程：

1. 对四台虚拟机进行网络和静态IP设置

   IP:

   node01: 192.168.5.128

   node02: 192.168.5.129

   node03: 192.168.5.130

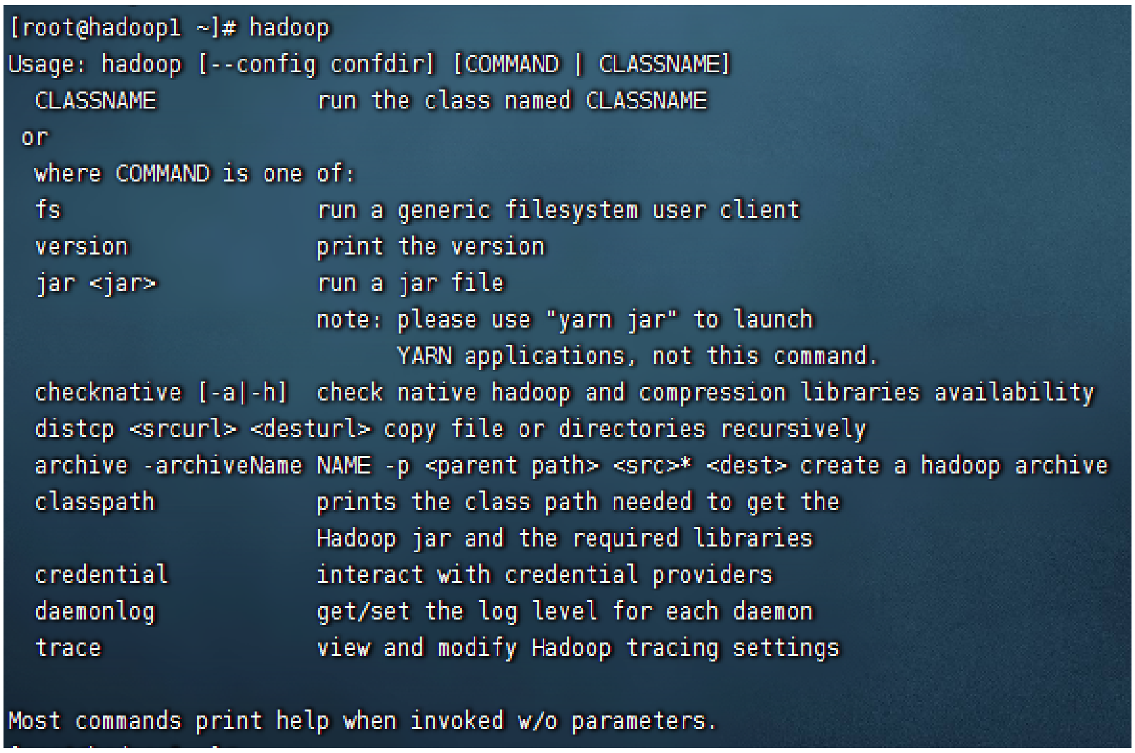
   node04: 192.168.5.131

2. Hadoop、jdk的安装

3. 创建hadoop文件目录

4. 导入jdk和hadoop的环境变量

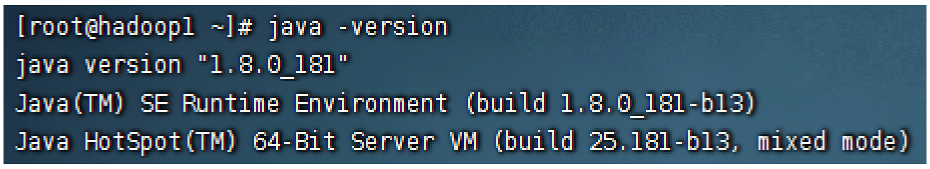
   Hadoop版本：2.8.5



```
[root@hadoop1 ~]# hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
  CLASSNAME            run the class named CLASSNAME
 or
  where COMMAND is one of:
  fs                   run a generic filesystem user client
  version              print the version
  jar <jar>            run a jar file
                       note: please use "yarn jar" to launch
                             YARN applications, not this command.
  checknative [-a|-h]  check native hadoop and compression libraries availability
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath            prints the class path needed to get the
                       Hadoop jar and the required libraries
  credential           interact with credential providers
  daemonlog            get/set the log level for each daemon
  trace                view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
```

jdk版本：1.8



```
[root@hadoop1 ~]# java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

5. 修改hadoop配置文件

配置文件的名称	**作用**
core-site.xml	核心配置文件，主要定义了我们文件访问的格式 hdfs://
hadoop-env.sh	主要配置我们的java路径
hdfs-site.xml	主要定义配置我们的hdfs的相关配置
mapred-site.xml	主要定义我们的mapreduce相关的一些配置
slaves	控制我们的从节点在哪里 datanode nodemanager在哪些机器上
yarm-site.xml	配置我们的resourcemanager资源调度

相关配置文件:

```
[root@localhost ~]# cd /usr/local/hadoop2.8.5/etc/hadoop/          -----进入配置文件目录
[root@localhost hadoop]# vi hadoop-env.sh
添加如下内容：export JAVA_HOME=/usr/local/jdk1.8

[root@localhost hadoop]# vi yarn-env.sh
添加如下内容：export JAVA_HOME=/usr/local/jdk1.8

[root@localhost hadoop]# vi mapred-env.sh
添加如下内容：export JAVA_HOME=/usr/local/jdk1.8
```

```
[root@localhost hadoop]# vi slaves
添加如下内容：
hadoop2
hadoop3
hadoop4

[root@localhost hadoop]# vi core-site.xml
添加如下代码：
<configuration>
         <property>
                 <name>fs.defaultFS</name>
                 <value>hdfs://hadoop1:9000/</value>
                 <description>设定namenode的主机名及端口</description>
          </property>
          <property>
                  <name>hadoop.tmp.dir</name>
                  <value>/usr/local/hadoop2.8.5/tmp/hadoop-${user.name}
</value>
                  <description>存储临时文件的目录</description>
           </property>
           <property>
              <name>hadoop.proxyuser.hadoop.hosts</name>
              <value>*</value>
           </property>
           <property>
                  <name>hadoop.proxyuser.hadoop.groups</name>
                  <value>*</value>
             </property>
 </configuration>

[root@localhost hadoop]# vi hdfs-site.xml
添加如下内容：
<configuration>
       <property>
                       <name>dfs.namenode.http-address</name>

                       <value>hadoop1:50070</value>
                       <description>NameNode地址和端口号</description>
       </property>
       <property>
                       <name>dfs.namenode.secondary.http-address</name>

                       <value>hadoop2:50090</value>
                       <description>SecondNameNode地址和端口号</description>

        </property>
        <property>
                        <name>dfs.replication</name>
                        <value>3</value>
                        <description>设定HDFS存储文件的副本个数，默认为
3</description>
        </property>
        <property>
                       <name>dfs.namenode.name.dir</name>
```

```
<value>file:///usr/local/hadoop2.8.5/hdfs/name</value>
                       <description>namenode用来持续存储命令空间和交换日志的本地文件系统路径</description>
        </property>
        <property>
                        <name>dfs.datanode.data.dir</name>

<value>file:///usr/local/hadoop2.8.5/hdfs/data</value>
                        <description>DataNode在本地存储块文件的目录列表
</description>
        </property>
        <property>
                      <name>dfs.namenode.checkpoint.dir</name>

<value>file:///usr/local/hadoop2.8.5/hdfs/namesecondary</value>

                      <description>设置secondarynamenode存储临时镜像的本地文件系统路径</description>
         </property>
         <property>
                         <name>dfs.webhdfs.enabled</name>
                         <value>true</value>
                         <description>是否允许网页浏览HDFS文件</description>
         </property>
         <property>
                        <name>dfs.stream-buffer-size</name>
                        <value>131072</value>
                        <description>默认是4kb，作为Hadoop缓冲区，用于Hadoop读HDFS的文件和写HDFS的文件，还有map的输出都用到了这个缓冲区容量，对于现在的硬件，可以设置为128kb（131072）</description>
         </property>
         <property>
                         <name>mapreduce.framework.name</name>

                         <value>yarn</value>
                          <description>Execution framework set to Hadoop YARN.</description>
          </property>
</configuration>

[root@localhost hadoop]# vi yarn-site.xml
添加如下内容：
<configuration>
<!-- Site specific YARN configuration properties -->
        <property>
                        <name>yarn.resourcemanager.hostname</name>

                        <value>hadoop1</value>
        </property>
        <property>
                        <name>yarn.nodemanager.aux-services</name>
                        <value>mapreduce_shuffle</value>
          </property>
          <property>
```

```
                          <name>yarn.nodemanage.aux-
services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>
            </property>
            <property>
                            <name>yarn.resourcemanager.address</name>

                            <value>hadoop1:8032</value>
            </property>
            <property>

<name>yarn.resourcemanager.schduler.address</name>
                            <value>hadoop1:8030</value>
            </property>
            <property>
                            <name>yarn.resourcemanager.resource-
tracker.address</name>
                            <value>hadoop1:8031</value>
            </property>
            <property>
                            <name>yarn.resourcemanager.admin.address</name>

                            <value>hadoop1:8033</value>
             </property>
             <property>

<name>yarn.resourcemanager.webapp.address</name>
                             <value>hadoop1:8088</value>
             </property>
</configuration>

[root@localhost hadoop]# vi mapred-site.xml.template
添加如下内容:
<configuration>
        <property>
                        <name>mapreduce.framework.name</name>
                        <value>yarn</value>
        </property>
        <property>
                        <name>mapreduce.jobhistory.address</name>

                        <value>hadoop1:10020</value>
        </property>
        <property>
                        <name>mapreduce.jobhistory.webapp.address</name>

                        <value>hadoop1:19888</value>
        </property>
</configuration>
```

6. 修改虚拟机主机名
7. 绑定hostname与ip地址

8. 关闭防火墙
9. 配置节点之间的免密登录
10. 格式化HDFS文件系统

    在namenode结点上:

    ```
    [root@hadoopnode1 .ssh]# hdfs namenode -format
    ```

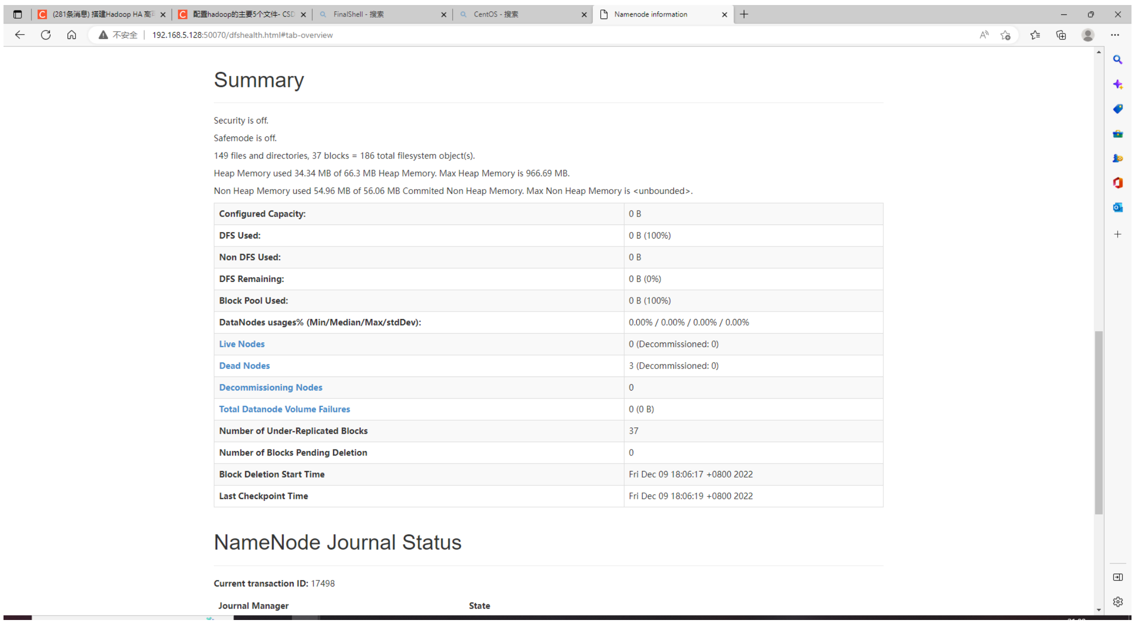
11. 启动HDFS文件系统

    在namenode结点上:

    ```
    [root@hadoopnode1 ~]# start-all.sh
    ```

    webUI网站查看信息:

    NameNode:


(281条消息) 搭建Hadoop HA 高
配置hadoop的主要5个文件- CSD
FinalShell - 搜索
CentOS - 搜索
Namenode information
不安全
192.168.5.128:50070/dfshealth.html#tab-overview

Summary

Security is off.

Safemode is off.

149 files and directories, 37 blocks = 186 total filesystem object(s).

Heap Memory used 34.34 MB of 66.3 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 54.96 MB of 56.06 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

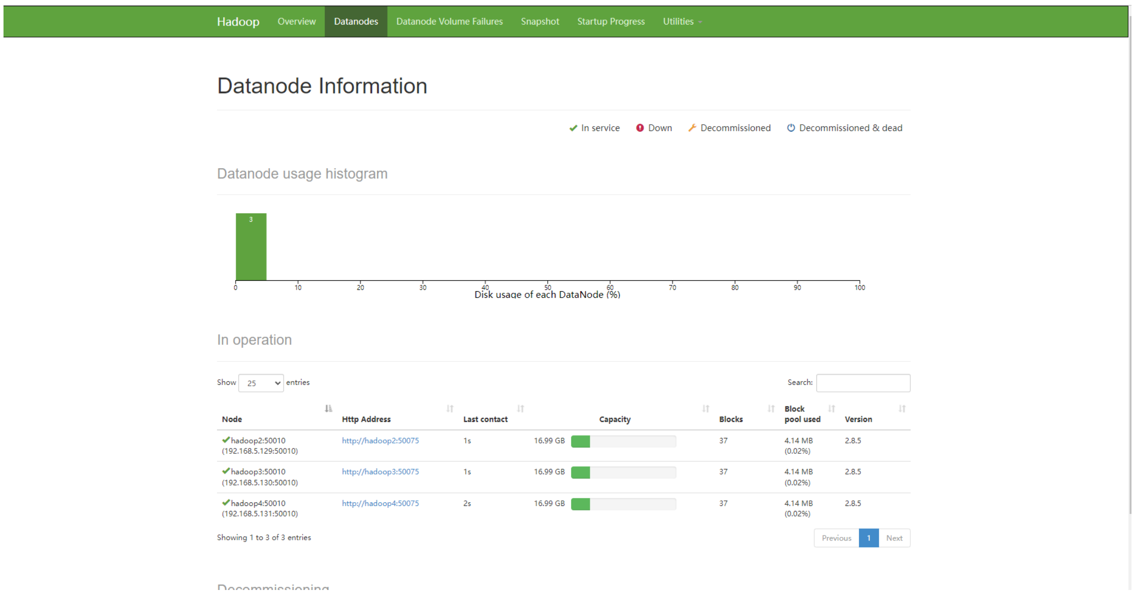
Configured Capacity:	0 B
DFS Used:	0 B (100%)
Non DFS Used:	0 B
DFS Remaining:	0 B (0%)
Block Pool Used:	0 B (100%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	0 (Decommissioned: 0)
Dead Nodes	3 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	37
Number of Blocks Pending Deletion	0
Block Deletion Start Time	Fri Dec 09 18:06:17 +0800 2022
Last Checkpoint Time	Fri Dec 09 18:06:19 +0800 2022

NameNode Journal Status

Current transaction ID: 17498

Journal Manager	State

DataNode:



Datanode Information

✓ In service ● Down Decommissioned Decommissioned & dead

Datanode usage histogram

In operation

Show 25 entries Search:

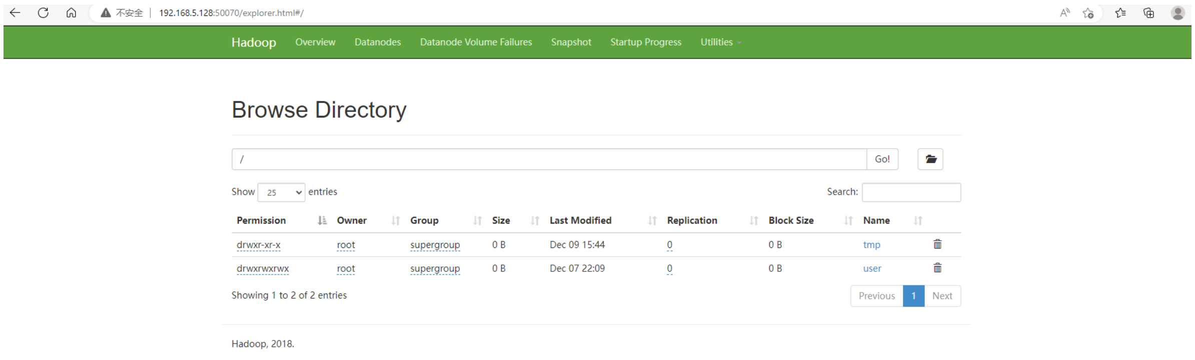
Node	Http Address	Last contact	Capacity	Blocks	Block pool used	Version
✓hadoop2:50010 (192.168.5.129:50010)	http://hadoop2:50075	1s	16.99 GB	37	4.14 MB (0.02%)	2.8.5
✓hadoop3:50010 (192.168.5.130:50010)	http://hadoop3:50075	1s	16.99 GB	37	4.14 MB (0.02%)	2.8.5
✓hadoop4:50010 (192.168.5.131:50010)	http://hadoop4:50075	2s	16.99 GB	37	4.14 MB (0.02%)	2.8.5

Showing 1 to 3 of 3 entries

Previous 1 Next

Hadoop分布式存储系统

Browse Directory：



192.168.5.128:50070/explorer.html#/

Browse Directory

/

Show 25 entries Search:

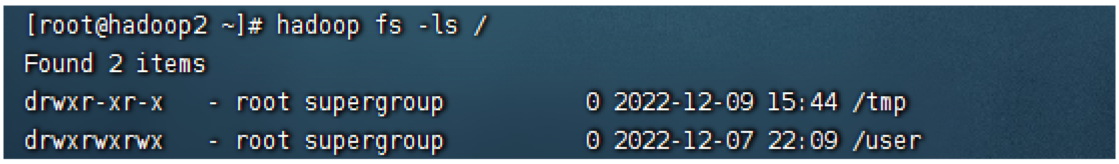
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	Dec 09 15:44	0	0 B	tmp
drwxrwxrwx	root	supergroup	0 B	Dec 07 22:09	0	0 B	user

Showing 1 to 2 of 2 entries

Hadoop, 2018.

查看文件目录：

执行： `hadoop fs -ls /`

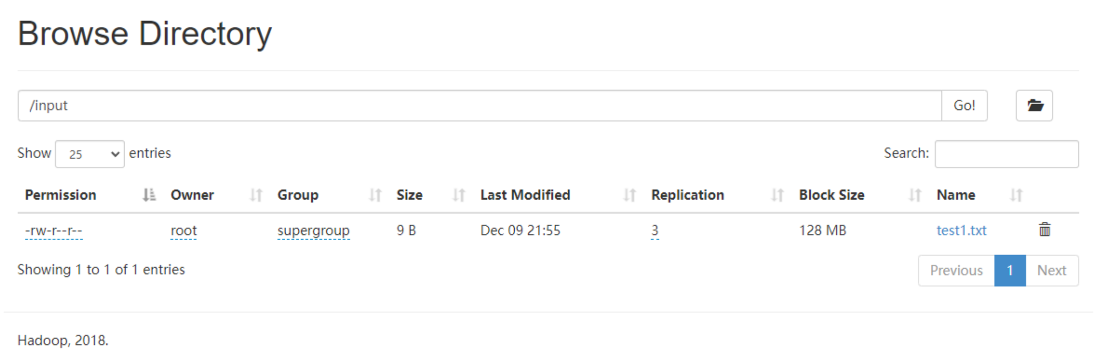


```
[root@hadoop2 ~]# hadoop fs -ls /
Found 2 items
drwxr-xr-x   - root supergroup          0 2022-12-09 15:44 /tmp
drwxrwxrwx   - root supergroup          0 2022-12-07 22:09 /user
```

文件上传：

执行：

```
hadoop dfs -mkdir /input
vi test1.txt
hadoop fs -put /root/test1.txt   /input
```



Browse Directory

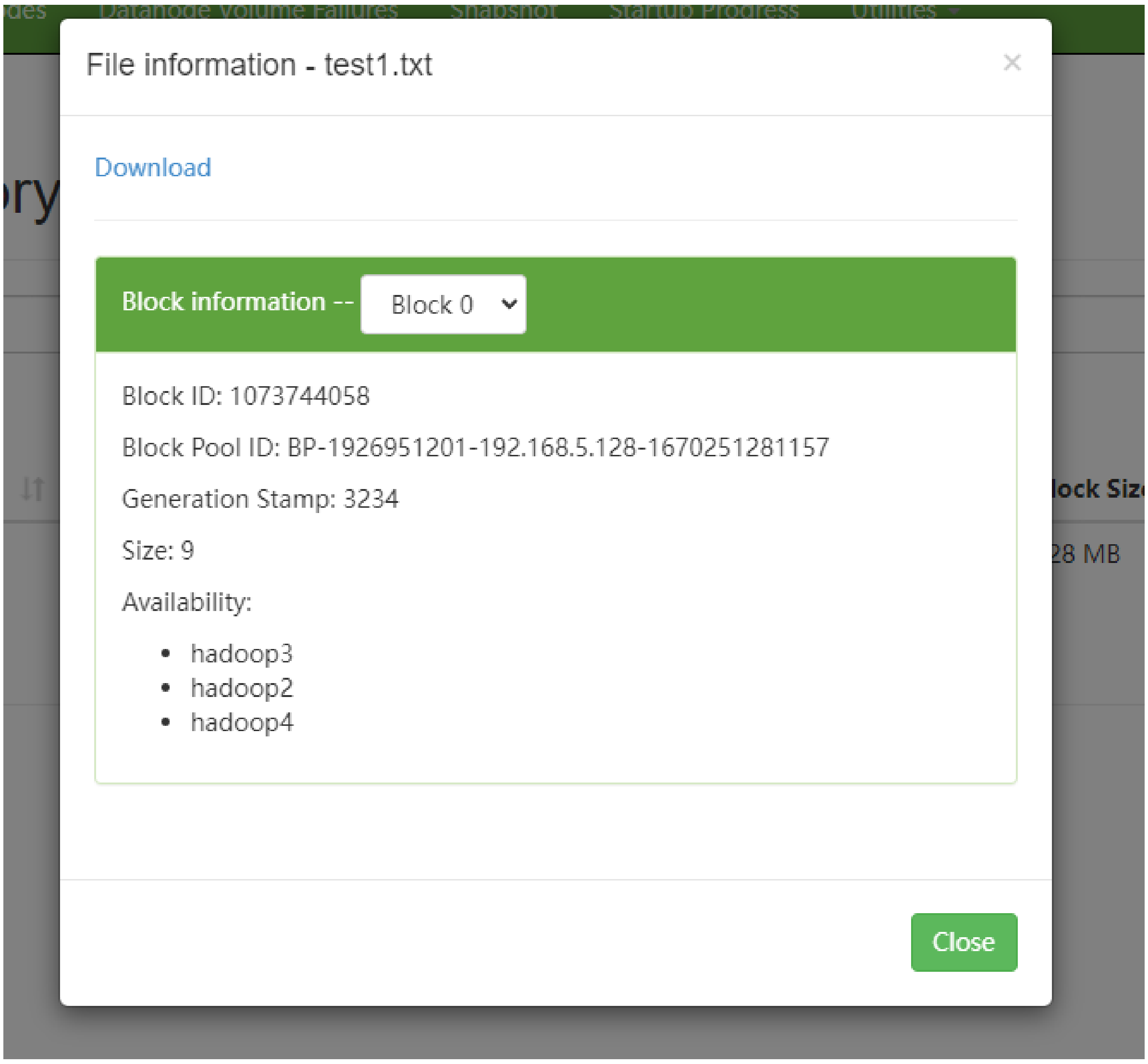
/input

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	9 B	Dec 09 21:55	3	128 MB	test1.txt

Showing 1 to 1 of 1 entries

Hadoop, 2018.


File information - test1.txt

Download

Block information -- Block 0

Block ID: 1073744058

Block Pool ID: BP-1926951201-192.168.5.128-1670251281157

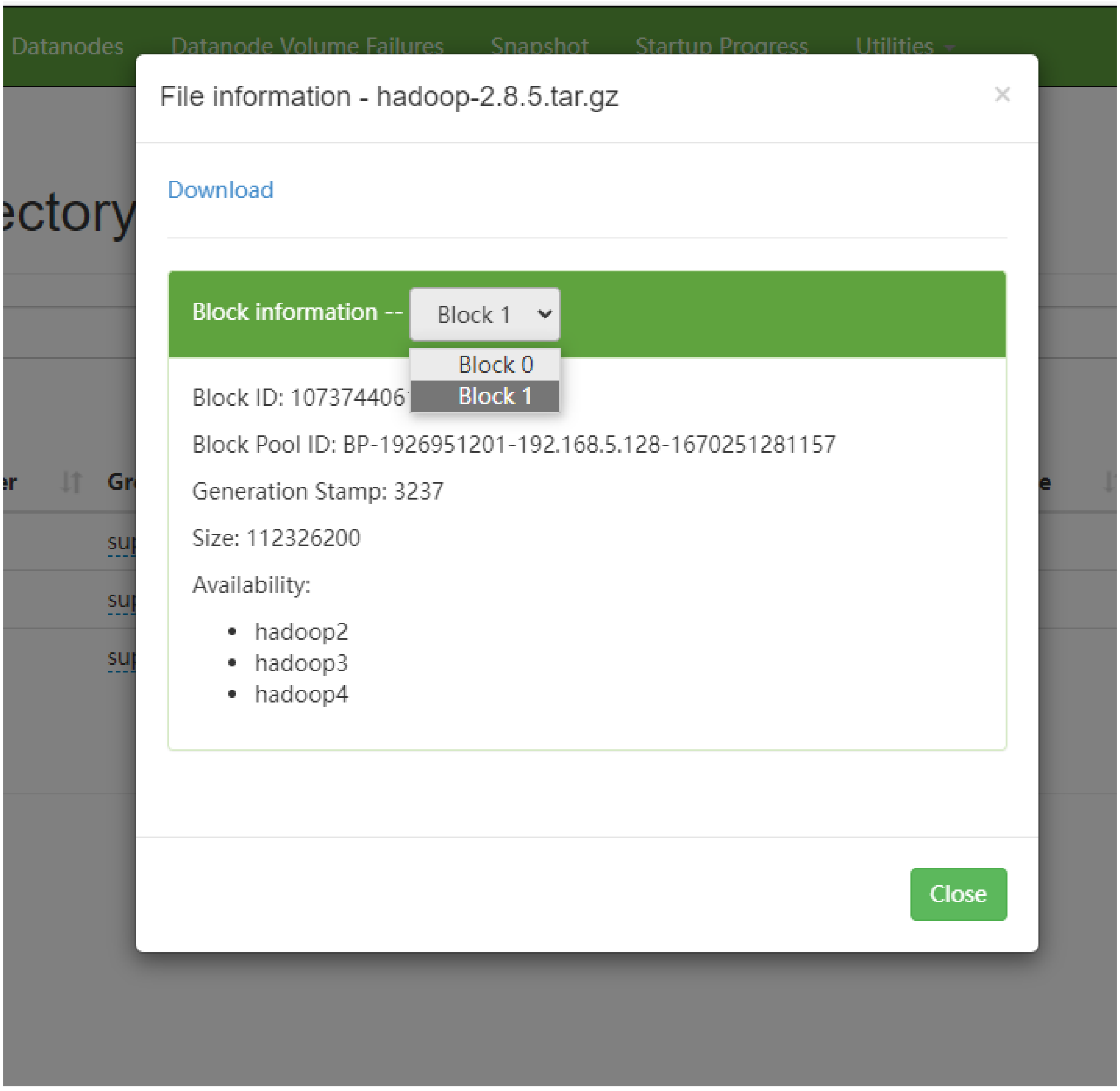
Generation Stamp: 3234

Size: 9

Availability:

- hadoop3
- hadoop2
- hadoop4

若上传文件过大，将被分为多个块：



File information - hadoop-2.8.5.tar.gz

Block ID: 107374406

Block Pool ID: BP-1926951201-192.168.5.128-1670251281157

Generation Stamp: 3237

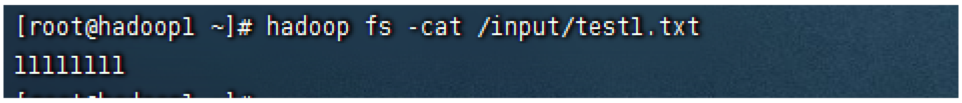
Size: 112326200

Availability:

- hadoop2
- hadoop3
- hadoop4

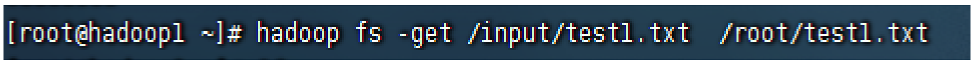
文件获取：

执行：`hadoop fs -cat /input/test1.txt`

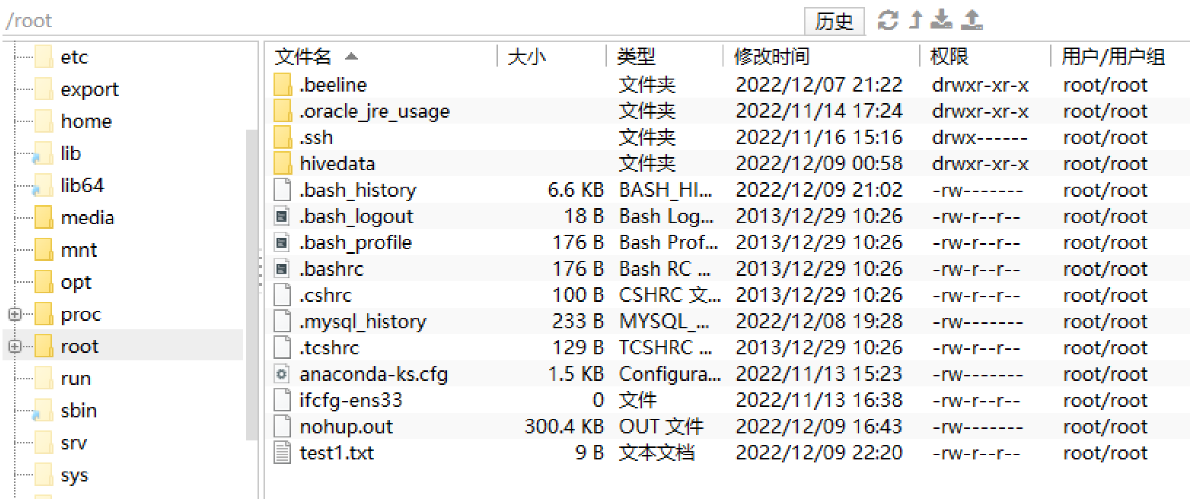


```
[root@hadoop1 ~]# hadoop fs -cat /input/test1.txt
11111111
```

执行：`hadoop fs -get /input/test1.txt  /root/test1.txt`



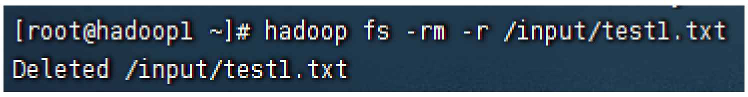
```
[root@hadoop1 ~]# hadoop fs -get /input/test1.txt  /root/test1.txt
```



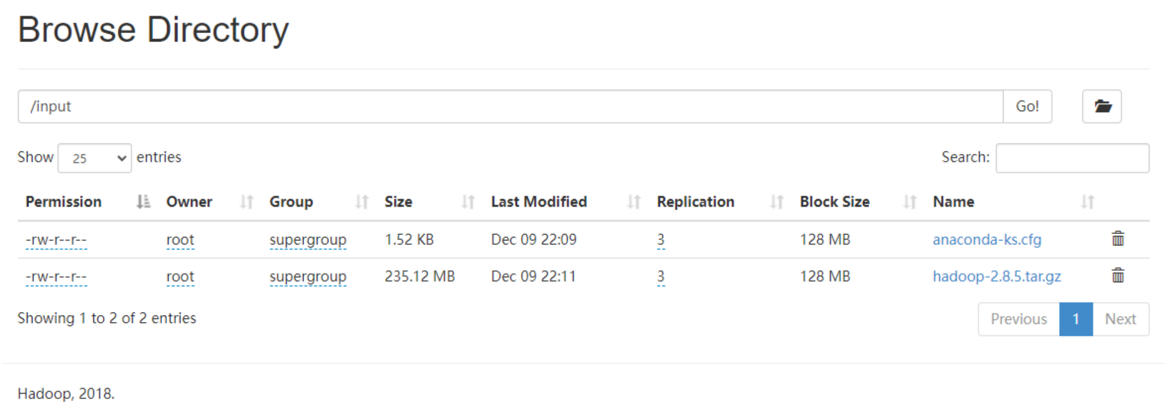
文件名	大小	类型	修改时间	权限	用户/用户组
.beeline		文件夹	2022/12/07 21:22	drwxr-xr-x	root/root
.oracle_jre_usage		文件夹	2022/11/14 17:24	drwxr-xr-x	root/root
.ssh		文件夹	2022/11/16 15:16	drwx------	root/root
hivedata		文件夹	2022/12/09 00:58	drwxr-xr-x	root/root
.bash_history	6.6 KB	BASH_HI...	2022/12/09 21:02	-rw-------	root/root
.bash_logout	18 B	Bash Log...	2013/12/29 10:26	-rw-r--r--	root/root
.bash_profile	176 B	Bash Prof...	2013/12/29 10:26	-rw-r--r--	root/root
.bashrc	176 B	Bash RC ...	2013/12/29 10:26	-rw-r--r--	root/root
.cshrc	100 B	CSHRC 文...	2013/12/29 10:26	-rw-r--r--	root/root
.mysql_history	233 B	MYSQL_...	2022/12/08 19:28	-rw-------	root/root
.tcshrc	129 B	TCSHRC ...	2013/12/29 10:26	-rw-r--r--	root/root
anaconda-ks.cfg	1.5 KB	Configura...	2022/11/13 15:23	-rw-------	root/root
ifcfg-ens33	0	文件	2022/11/13 16:38	-rw-r--r--	root/root
nohup.out	300.4 KB	OUT 文件	2022/12/09 16:43	-rw-------	root/root
test1.txt	9 B	文本文档	2022/12/09 22:20	-rw-r--r--	root/root

文件删除：

执行：`hadoop fs -rm -r /input/test1.txt`



```
[root@hadoop1 ~]# hadoop fs -rm -r /input/test1.txt
Deleted /input/test1.txt
```



Browse Directory

/input

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	1.52 KB	Dec 09 22:09	3	128 MB	anaconda-ks.cfg
-rw-r--r--	root	supergroup	235.12 MB	Dec 09 22:11	3	128 MB	hadoop-2.8.5.tar.gz

Showing 1 to 2 of 2 entries

Hadoop, 2018.

可见,test1.txt已被删除

Hive配置

1. MySQL安装，版本：mysql-5.7.29-1.el7.x86_64

```
mkdir /export/software/mysql

#上传mysql-5.7.29-1.el7.x86_64.rpm-bundle.tar 到上述文件夹下  解压
tar xvf mysql-5.7.29-1.el7.x86_64.rpm-bundle.tar

#执行安装
yum -y install libaio

[root@node3 mysql]# rpm -ivh mysql-community-common-5.7.29-1.el7.x86_64.rpm
mysql-community-libs-5.7.29-1.el7.x86_64.rpm mysql-community-client-5.7.29-
1.el7.x86_64.rpm mysql-community-server-5.7.29-1.el7.x86_64.rpm

warning: mysql-community-common-5.7.29-1.el7.x86_64.rpm: Header V3 DSA/SHA1
Signature, key ID 5072e1f5: NOKEY
Preparing...                          #################################
[100%]
Updating / installing...
   1:mysql-community-common-5.7.29-1.e################################# [
25%]
   2:mysql-community-libs-5.7.29-1.el7################################# [
50%]
   3:mysql-community-client-5.7.29-1.e################################# [
75%]
   4:mysql-community-server-5.7.29-1.e################                  (
49%)
```

2. 修改配置文件：

   - hive-env.sh

     ```
     cd /export/server/apache-hive-3.1.2-bin/conf
     mv hive-env.sh.template hive-env.sh

     vim hive-env.sh
     export HADOOP_HOME=/export/server/hadoop-3.3.0
     export HIVE_CONF_DIR=/export/server/apache-hive-3.1.2-bin/conf
     export HIVE_AUX_JARS_PATH=/export/server/apache-hive-3.1.2-bin/lib
     ```

   - hive-site.xml

     ```
     <configuration>
     <!-- 存储元数据mysql相关配置 -->
     <property>
         <name>javax.jdo.option.ConnectionURL</name>
         <value>jdbc:mysql://192.168.5.128:3306/hive3?
     createDatabaseIfNotExist=true&amp;useSSL=false&amp;useUnicode=true&amp;c
     haracterEncoding=UTF-8</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>hadoop</value>
     </property>

     <!-- H2S运行绑定host -->
     <property>
         <name>hive.server2.thrift.bind.host</name>
         <value>192.168.5.128</value>
     </property>

     <!-- 远程模式部署metastore metastore地址 -->
     <property>
         <name>hive.metastore.uris</name>
         <value>thrift://192.168.5.128:9083</value>
     </property>

     <!-- 关闭元数据存储授权  -->
     <property>
         <name>hive.metastore.event.db.notification.api.auth</name>
         <value>false</value>
     </property>
     ```

```
</configuration>
```

3. 启动hive

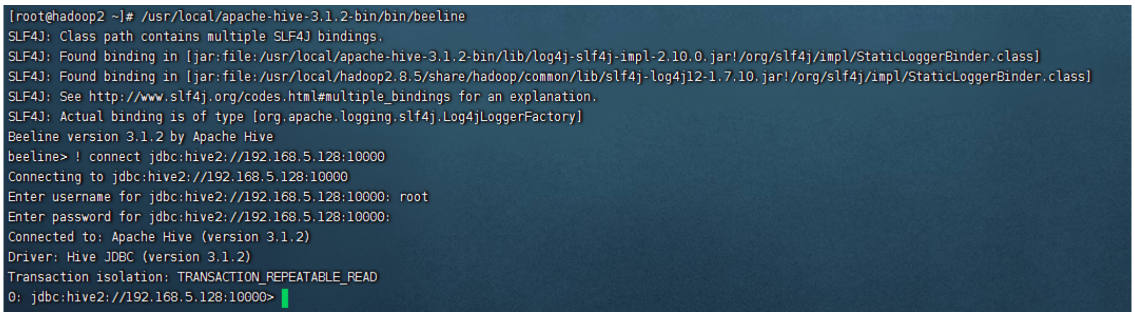
```
nohup /export/server/apache-hive-3.1.2-bin/bin/hive --service metastore &
nohup /export/server/apache-hive-3.1.2-bin/bin/hive --service hiveserver2 &
```

4. beeline客户端连接

   如下图所示：

   执行：

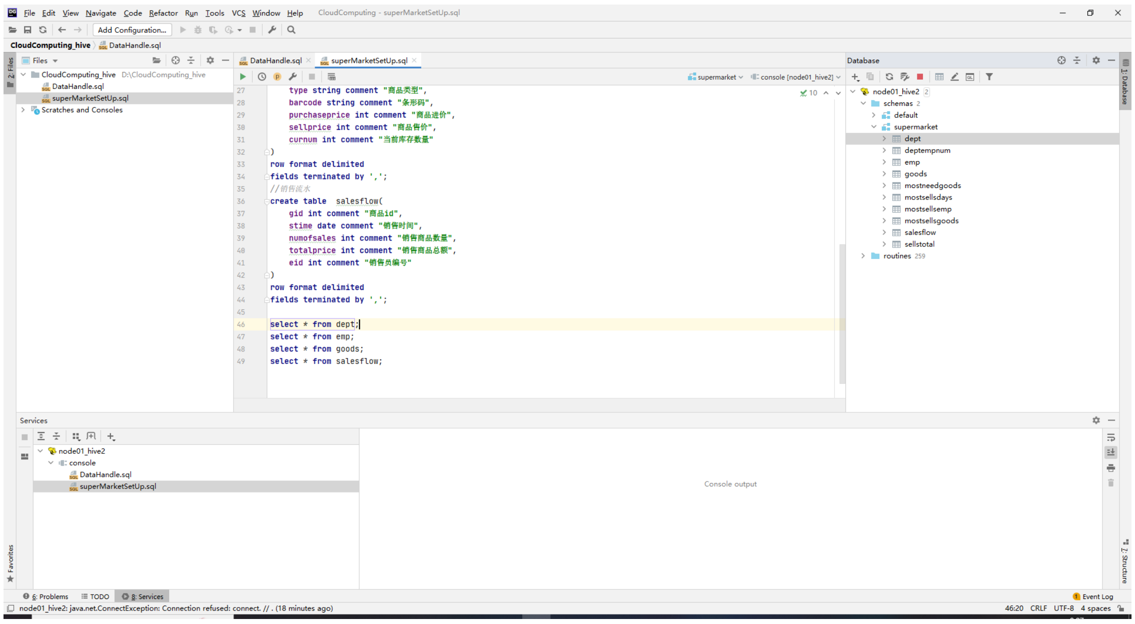
```
 /usr/local/apache-hive-3.1.2-bin/bin/beeline
 ! connect jdbc:hive2://192.168.5.128:10000
```



```
[root@hadoop2 ~]# /usr/local/apache-hive-3.1.2-bin/bin/beeline
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop2.8.5/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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.apache.logging.slf4j.Log4jLoggerFactory]
Beeline version 3.1.2 by Apache Hive
beeline> ! connect jdbc:hive2://192.168.5.128:10000
Connecting to jdbc:hive2://192.168.5.128:10000
Enter username for jdbc:hive2://192.168.5.128:10000: root
Enter password for jdbc:hive2://192.168.5.128:10000:
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://192.168.5.128:10000>
```

5. DataGrip使用

   本次项目采用DataGrip工具进行数据库的搭建以及数据的处理，连接上hive数据库后，即可进行hive sql的编写



```
    type string comment "商品类型",
    barcode string comment "条形码",
    purchaseprice int comment "商品进价",
    sellprice int comment "商品售价",
    curnum int comment "当前库存数量"
)
row format delimited
fields terminated by ',';
//销售流水
create table  salesflow(
    gid int comment "商品id",
    stime date comment "销售时间",
    numofsales int comment "销售商品数量",
    totalprice int comment "销售商品总额",
    eid int comment "销售员编号"
)
row format delimited
fields terminated by ',';

select * from dept;
select * from emp;
select * from goods;
select * from salesflow;
```

```
node01_hive2 2
  schemas 2
    default
    supermarket
      dept
      deptempnum
      emp
      goods
      mostneedgoods
      mostsellsdays
      mostsellsemp
      mostsellsgoods
      salesflow
      sellstotal
  routines 259
```

```
node01_hive2: java.net.ConnectException: Connection refused: connect. // . (18 minutes ago)
```

数据库设计

初始数据源(元数据)

goods：存储商品的信息

    gid:商品id   gname:商品名字   type:商品类型   barcode:商品码   purchaseprice:购入价格   sellprice:出售价格   curnum:当前库存

emp：存储员工信息

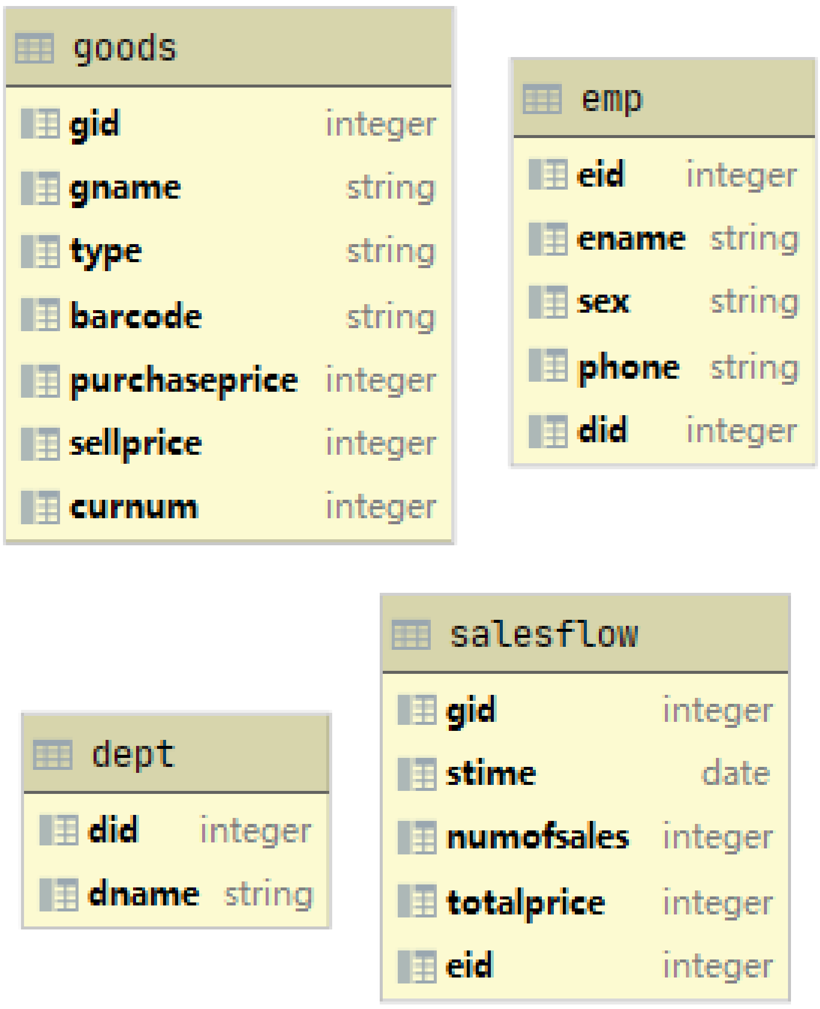
edi:员工id　　ename:员工姓名　　sex:员工性别　　phone:员工电话号码　　did:所属部门id

dept：存储部门信息

did:部门id　　dname:部门名称

salesflow:存储每日商品的出售量

gid:出售商品id　　stime:出售时间　　numofsales:出售数量　　totalprice:出售总价　　eid:售出的员工



goods	
gid	integer
gname	string
type	string
barcode	string
purchaseprice	integer
sellprice	integer
curnum	integer

emp	
eid	integer
ename	string
sex	string
phone	string
did	integer

dept	
did	integer
dname	string

salesflow	
gid	integer
stime	date
numofsales	integer
totalprice	integer
eid	integer

相应sql脚本:

```
create database SuperMarket;

//部门表
create table dept(
    did int comment  "部门id",
    dname string comment "部门名称"
 )
```

```
row format delimited
fields terminated by ',';
//员工表
create table emp(
    eid int comment "员工id",
    ename string comment "员工姓名",
    sex string comment "员工性别",
    phone string comment "电话号码",
    did int comment "所属部门id"
)
row format delimited
fields terminated by ',';
//商品表
create table goods(
    gid int comment "商品id",
    gname string comment "商品名称",
    type string comment "商品类型",
    barcode string comment "条形码",
    purchaseprice int comment "商品进价",
    sellprice int comment "商品售价",
    curnum int comment "当前库存数量"
)
row format delimited
fields terminated by ',';
//销售流水
create table  salesflow(
    gid int comment "商品id",
    stime date comment "销售时间",
    numofsales int comment "销售商品数量",
    totalprice int comment "销售商品总额",
    eid int comment "销售员编号"
)
row format delimited
fields terminated by ',';
```

数据处理

为了体现云计算对某数据库的处理过程，模拟了相应数据，并利用相应脚本对数据库中数据进行筛选、排序等处理，完成一定的功能：

1. 筛选出出售量最多的10天：

```
create table if not exists mostSellsDays
comment "出售最多的10天"
as
select
  stime,
  sum(totalprice) as totalsales
from supermarket.salesflow
group by stime
order by totalsales desc
limit 10;
```

2. 统计各部门员工人数：

```
create table if not exists deptEmpNum
comment "统计员工部门人数"
as
select
  d.did as deptID,
  d.dname as deptName,
  count(*) as memberNum
from supermarket.emp join supermarket.dept d on emp.did = d.did
group by d.did,d.dname;
```

3. 统计该年度销售量最多的员工:

```
create table if not exists mostSellsEmp
comment "统计销售最多的员工"
as
select
  e.eid as empID,
  ename as empName,
  sum(totalprice) as salesNum
from supermarket.salesflow join emp e on salesflow.eid = e.eid
group by e.eid,ename
order by salesNum desc
limit 10;
```

4. 统计最畅销的商品:

```
create table if not exists mostSellsGoods
comment "统计最畅销的商品"
as
select
  g.gid,
  gname,
  sum(numofsales) as salesAmount
from supermarket.salesflow join goods g on salesflow.gid = g.gid
group by g.gid,gname
order by salesAmount desc
limit 10;
```

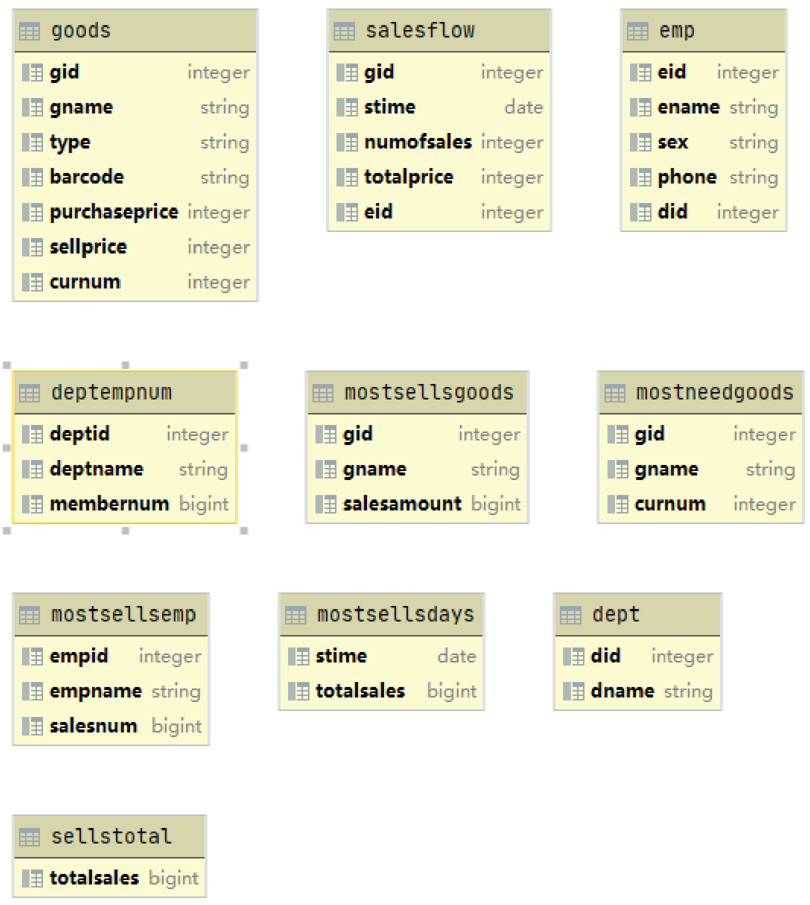
5. 统计最缺货的商品:

```
create table if not exists mostNeedGoods
comment "统计最缺货的商品"
as
select
  gid,
  gname,
  curnum
from supermarket.goods
order by curnum
limit 10;
```

6. 统计出售的总量:

```
create table if not exists sellsTotal
comment "出售总量"
as
select
  sum(totalprice) as totalsales
from supermarket.salesflow;
```

最终生成数据库如下：



goods	
gid	integer
gname	string
type	string
barcode	string
purchaseprice	integer
sellprice	integer
curnum	integer

salesflow	
gid	integer
stime	date
numofsales	integer
totalprice	integer
eid	integer

emp	
eid	integer
ename	string
sex	string
phone	string
did	integer

deptempnum	
deptid	integer
deptname	string
membernum	bigint

mostsellsgoods	
gid	integer
gname	string
salesamount	bigint

mostneedgoods	
gid	integer
gname	string
curnum	integer

mostsellsemp	
empid	integer
empname	string
salesnum	bigint

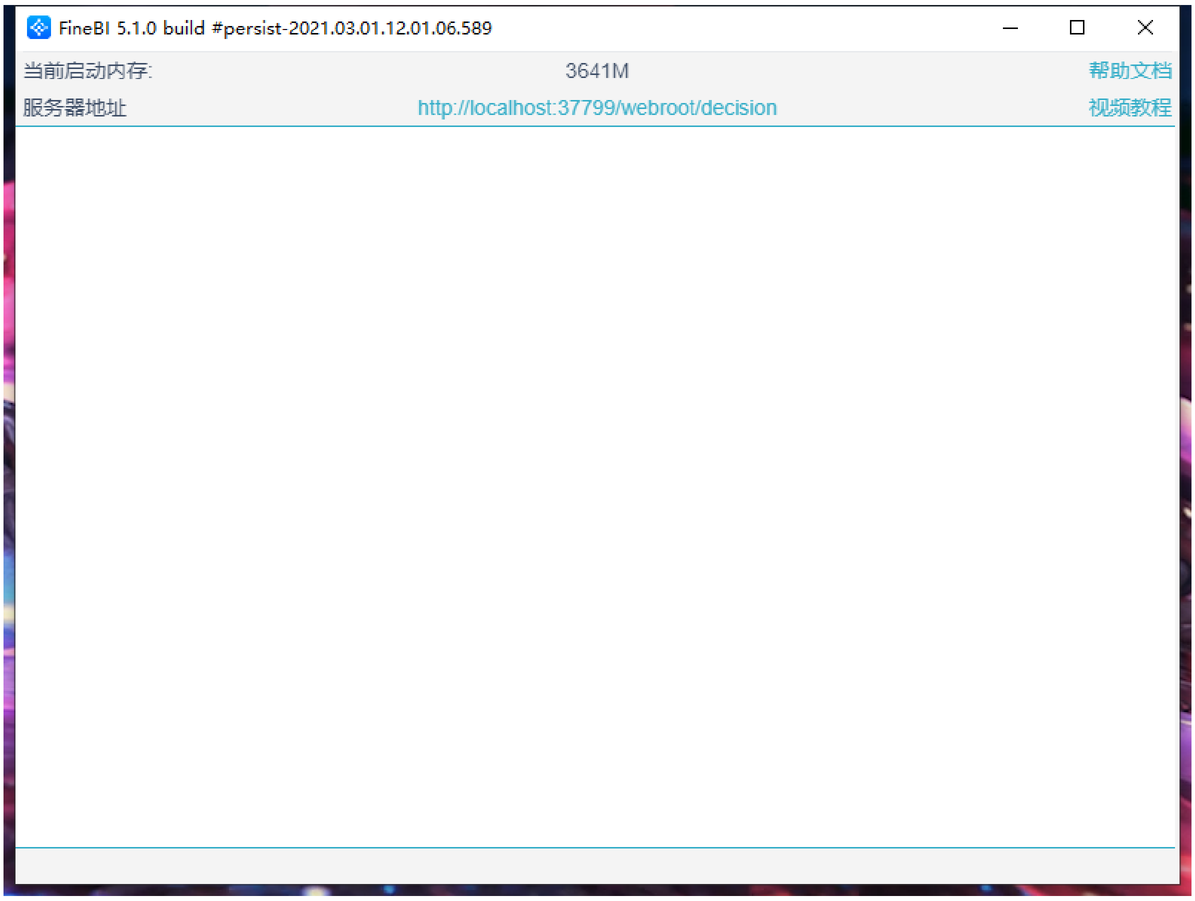
mostsellsdays	
stime	date
totalsales	bigint

dept	
did	integer
dname	string

sellstotal	
totalsales	bigint

可视化处理

本次项目借助FinBI,对最终处理结果进行呈现：

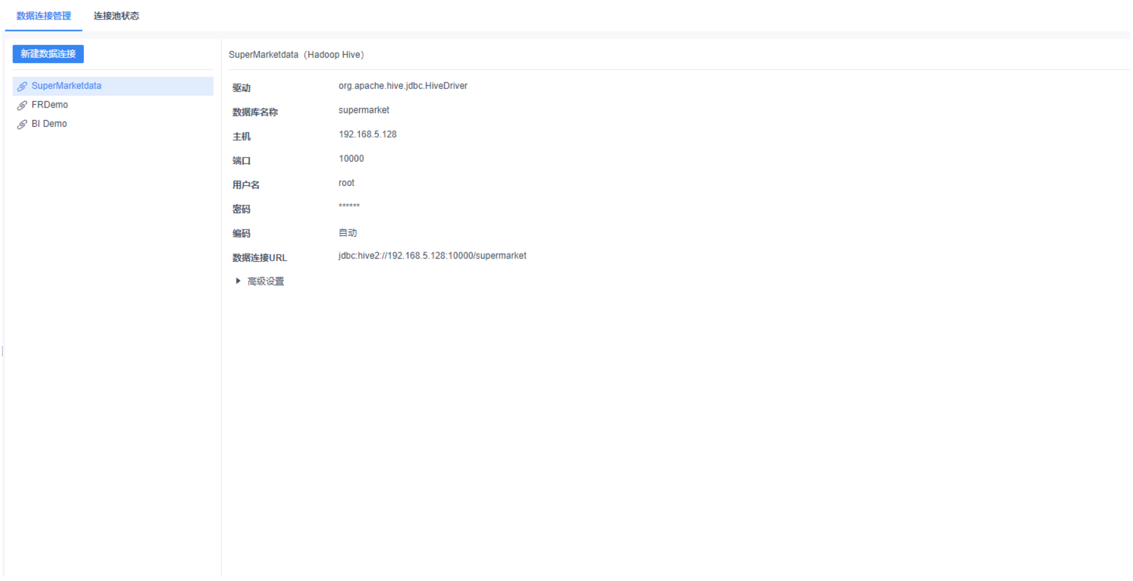

FineBI 5.1.0 build #persist-2021.03.01.12.01.06.589

当前启动内存: 3641M 帮助文档

服务器地址 http://localhost:37799/webroot/decision 视频教程

具体过程:

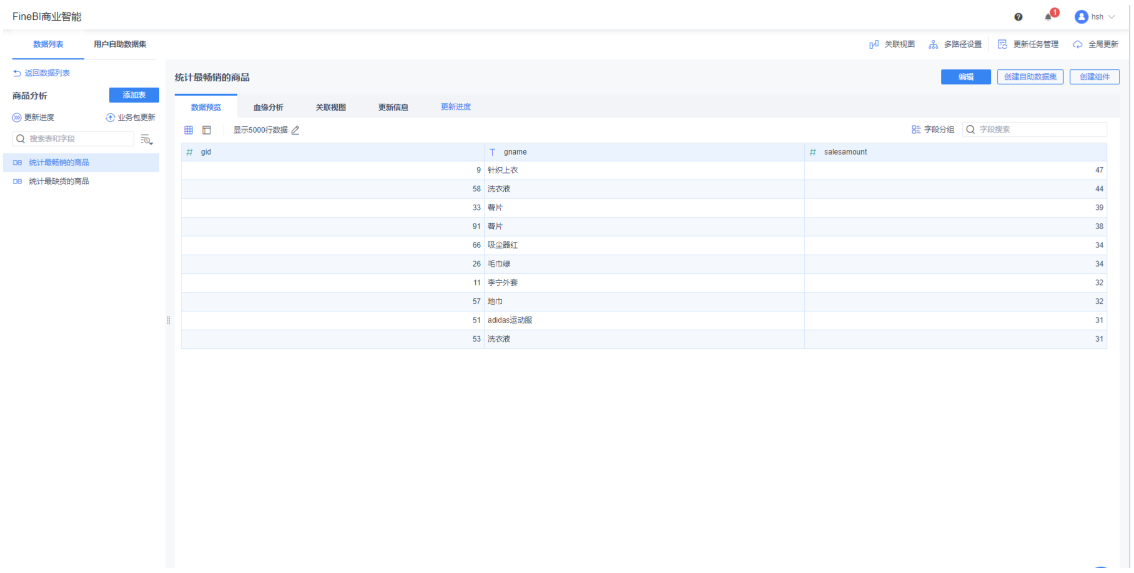
1. 连接hive数据库



SuperMarketdata (Hadoop Hive)

驱动	org.apache.hive.jdbc.HiveDriver
数据库名称	supermarket
主机	192.168.5.128
端口	10000
用户名	root
密码	******
编码	自动
数据连接URL	jdbc:hive2://192.168.5.128:10000/supermarket

2. 导入处理数据后生成的表数据

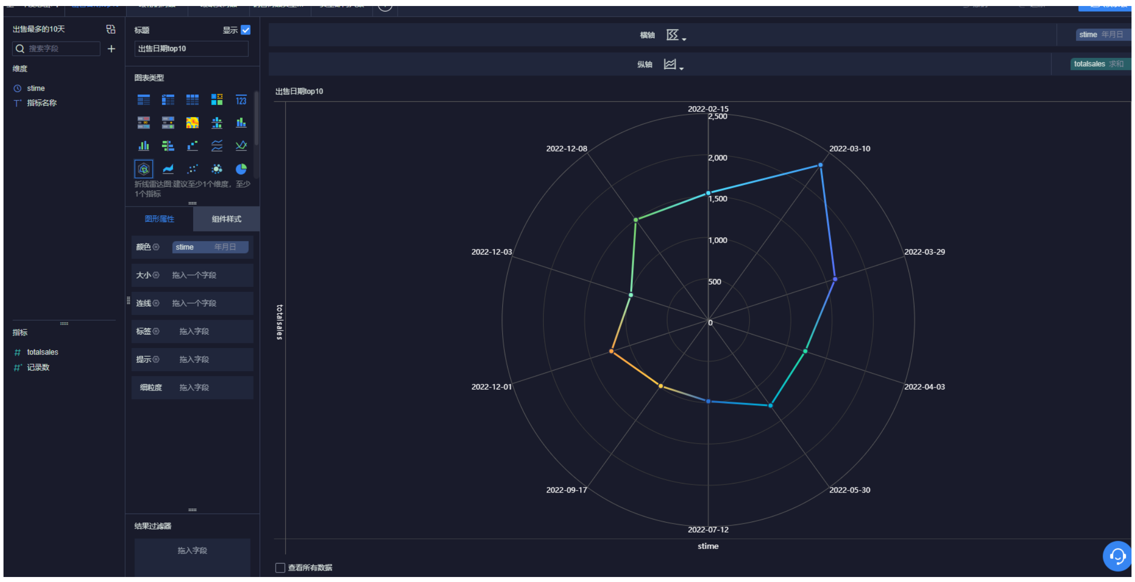


统计最畅销的商品

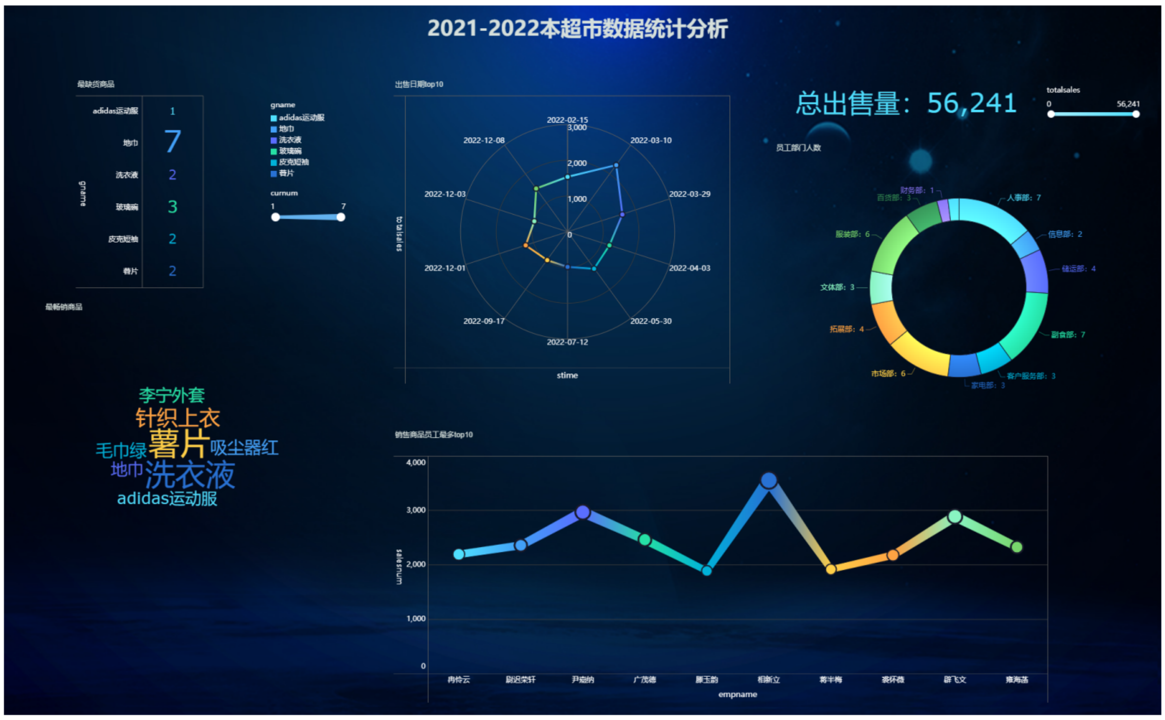
显示5000行数据

# gid	T gname	# salesamount
9	针织上衣	47
58	洗衣液	44
33	薯片	39
91	薯片	38
66	吸尘器红	34
26	毛巾绿	34
11	李宁外套	32
57	地巾	32
51	adidas运动服	31
53	洗衣液	31

3. 对仪表盘进行设计以及数据导入



最终可视化结果：


2021-2022本超市数据统计分析
最缺货商品
gname
adidas运动服 1
地巾 7
洗衣液 2
玻璃碗 3
皮克短袖 2
薯片 2
gname
adidas运动服
地巾
洗衣液
玻璃碗
皮克短袖
薯片
curnum
最畅销商品
李宁外套
针织上衣
薯片
毛巾绿
吸尘器红
地巾
洗衣液
adidas运动服
出售日期top10
totalsales
stime
总出售量：56,241
totalsales
员工部门人数
财务部：1
百货部：3
人事部：7
信息部：2
储运部：4
服装部：6
文体部：3
拓展部：4
副食部：7
市场部：6
客户服务部：3
家电部：3
销售商品员工最多top10
salesnum
冉伶云
尉迟荣轩
尹嘉纳
广茂德
滕玉韵
相新立
蒋半梅
裘怀薇
辟飞文
雍海菡
empname