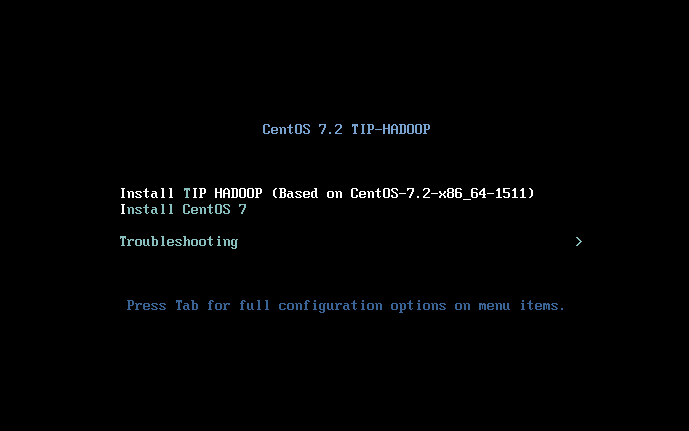
# 环境准备

1. TIP服务器3台以上；
2. 每台服务器128GB内存；
3. 每台服务器至少2张网卡；
4. 每台服务器至少2块磁盘配置RAID1，用于安装操作系统；
5. 每台服务器至少6块磁盘，用于存储（不做Raid）。

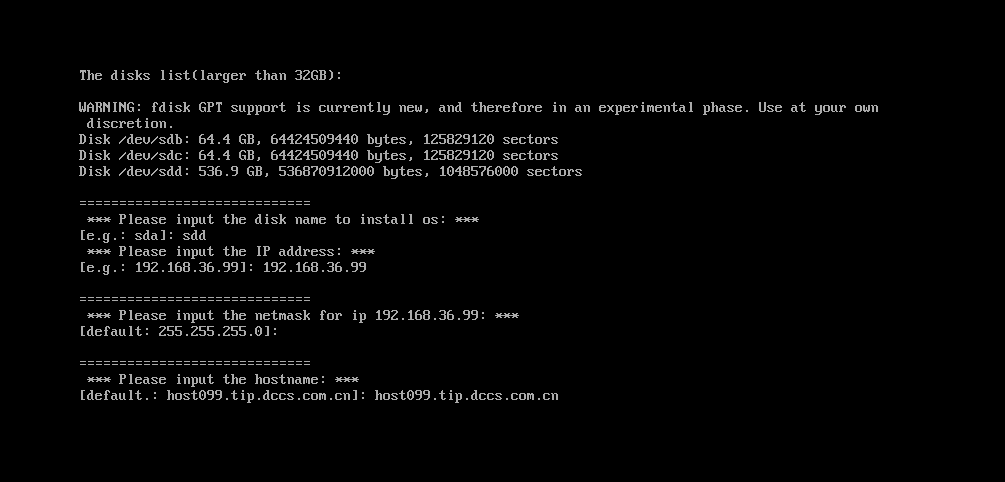
# 安装集群

## 从U盘启动，选择Install TIP Hadoop：

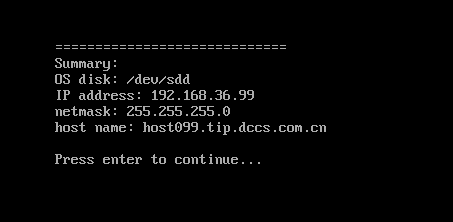


## 根据提示，依次执行以下步骤

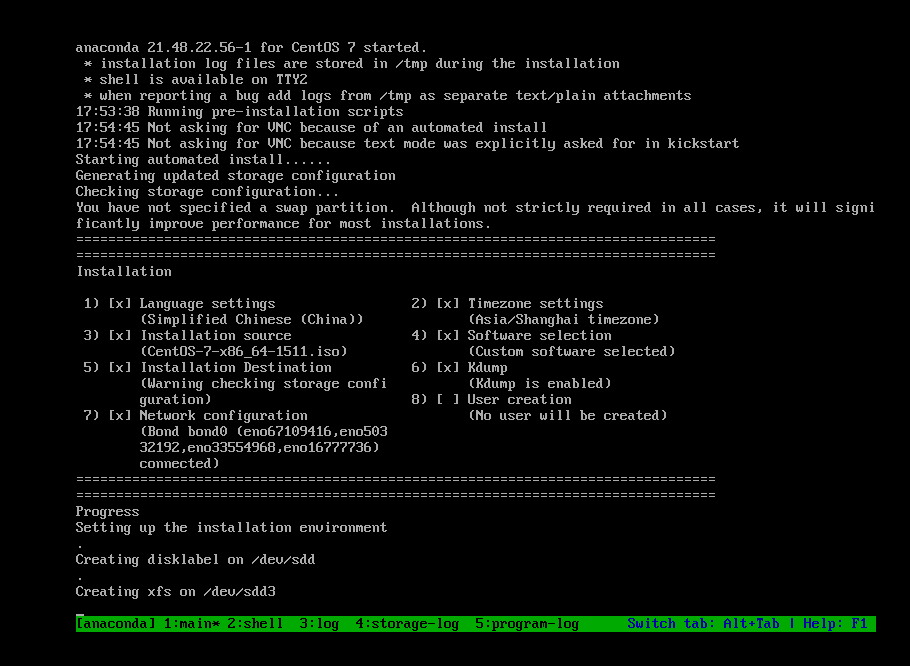
1. 在列表中选择用来安装OS的磁盘，如下图所示：



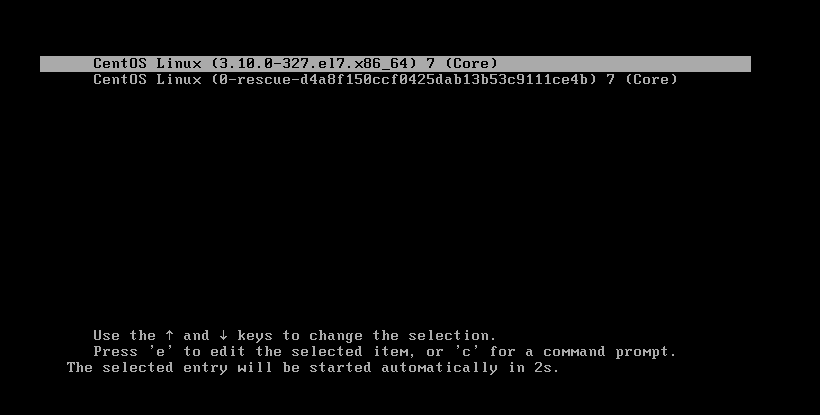
1. 输入完成后，将出现如下图所示的Summary页面。如果输入正确，按回车键继续；如果输入有误，则重启服务器，重新从U盘启动安装程序。



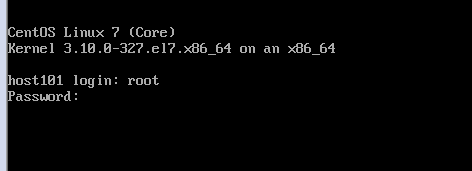
1. 按回车键后继续安装程序，直到系统自动重启：



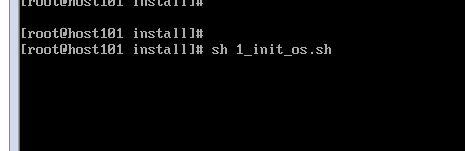
1. 系统自动重启后，拔出U盘，从硬盘引导：



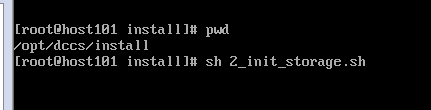
1. 看到系统登录提示符之后，输入用户名 root ，密码 Dccs12345. 登录：

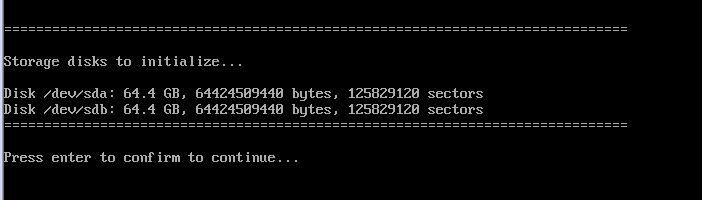


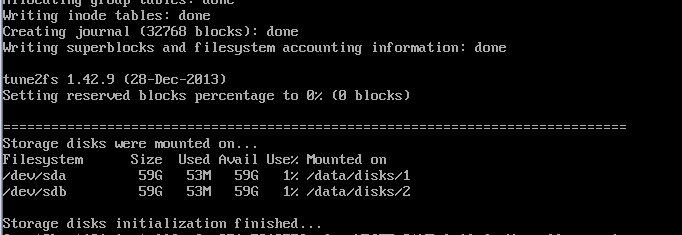
1. 进入 /opt/dccs/install 目录，执行 sh 1\_init\_os.sh 来初始化操作系统，OS初始化完成后会自动重启：



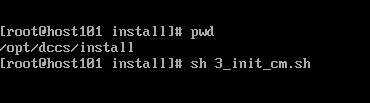
1. 重启后，再次进入/opt/dccs/install 目录，执行2\_init\_storage.sh来初始化存储盘：



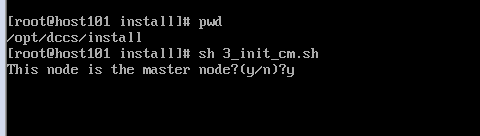




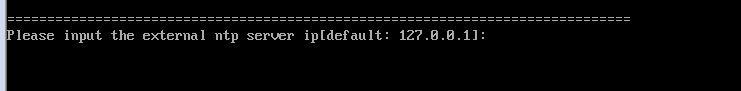
1. 存储盘初始化完成后，执行3\_init\_cm.sh来初始集群代理服务：



8.1 首先选择当前正在初始化的节点是否为master节点：



8.2 如果当前节点是master节点，则输入外部NTP服务器的IP地址，如果当前节点不是master节点，则输入master节点的IP地址：



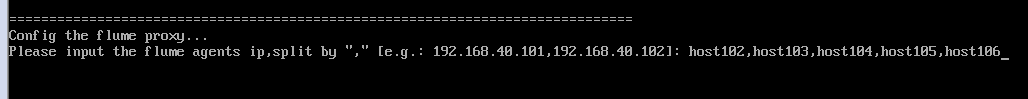
1. 集群代理服务始化完成后，执行4\_init\_tip.sh来初始化TIP服务：

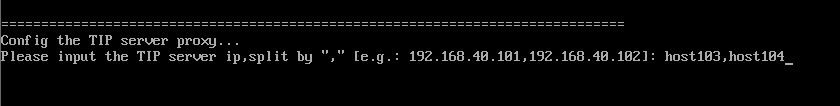
9.1 输入tip战法服务所依赖的MySQL服务器的IP地址，输入Master节点的IP地址即可：

9.2 输入tip战法服务所依赖的Impala Deamon服务器的IP地址和端口，一般情况下直接回车即可：

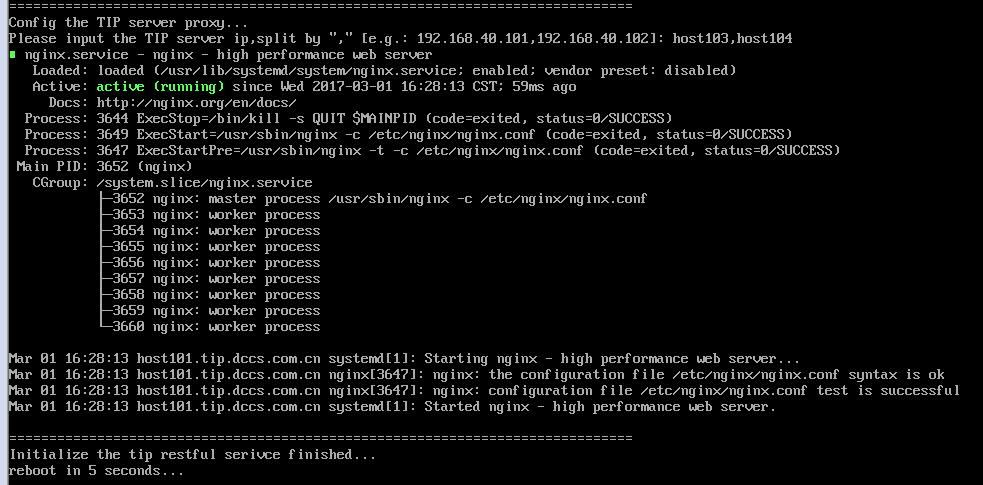
9.3 输入tip战法服务所依赖的Zookeeper服务器的IP地址和端口，一般情况下输入host102所在机器的IP地址即可：



9.4 输入flume代理服务所在服务器的IP地址，一般情况下输入host102、host103、host104、host105、host106所在服务器的IP即可：

9.4 输入flume代理服务所在服务器的IP地址，一般情况下输入host103、host104所在服务器的IP即可：

9.5 出现如下提示，则表示初始化成功：

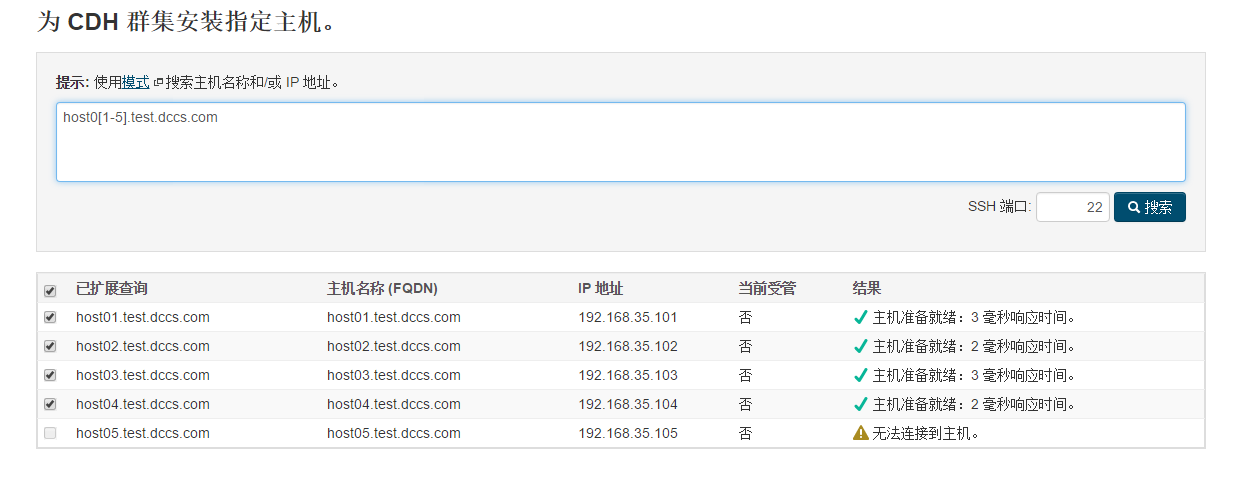


# 安装集群

1. 在Chrome浏览器中，输入http://<master节点IP>:1780；
2. 在Cloudera Manager的登录页面中输入用户名和密码（都是：admin）登录，并接受“最终用户许可条款和条件”；
3. 部署Cloudera Express版：

## 添加服务器

1. 输入 host[101-110].tip.dccs.com.cn 后点击搜索，如下图所示：



找到符合条件的服务器后，点击“继续”；

1. 选择CDH版本：5.10.0版的CDH和KUDU



（1）、在“定义存储库”中填写： http:// <master节点IP>:18880/cm/5；

（2）、在“自定义 GPG 主要 URL:”中填写：http://192.168.36.200:18880/cm/RPM-GPG-KEY-cloudera

## 群集安装





## 安装配置Hadoop组件

### 选择组件

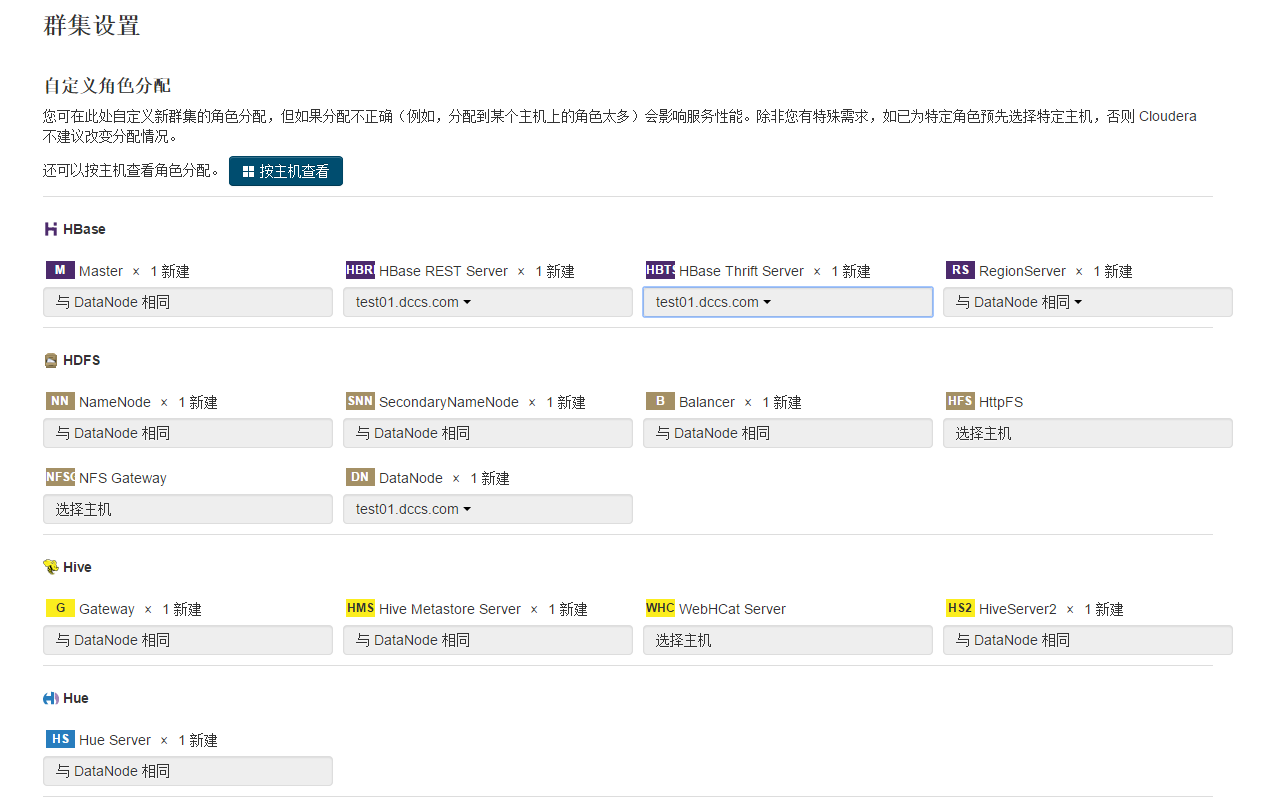
1. 选择“自定义服务”；
2. 选择Hbase、HDFS、Hive、Hue、Impala、Oozie、Spark、Sqoop2、YARN和Zookeeper。



图 3‑9

### 设置主机角色

根据集群大小和项目实际情况分配组件。



### 群集设置

由于假设本集群服务器中有8块磁盘用于数据存储，所以在“集群设置”中需要参考[配置存储](#_配置存储_1)一节中的设置情况来配置新的存储路径。

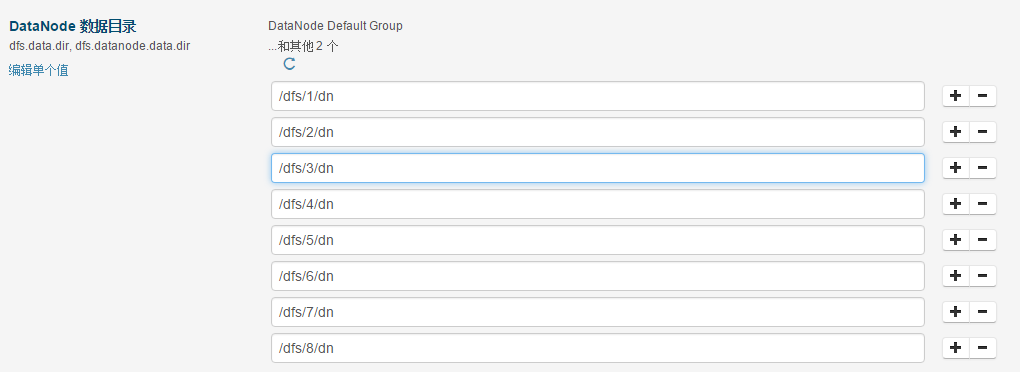
1. 如图3-13，将“DataNode 数据目录”设置为：/data/disks/1/dn, /data/disks/2/dn, /data/disks/3/dn, /data/disks/4/dn, /data/disks/5/dn, /data/disks/6/dn, /data/disks/7/dn, /data/disks/8/dn;

图 3‑13

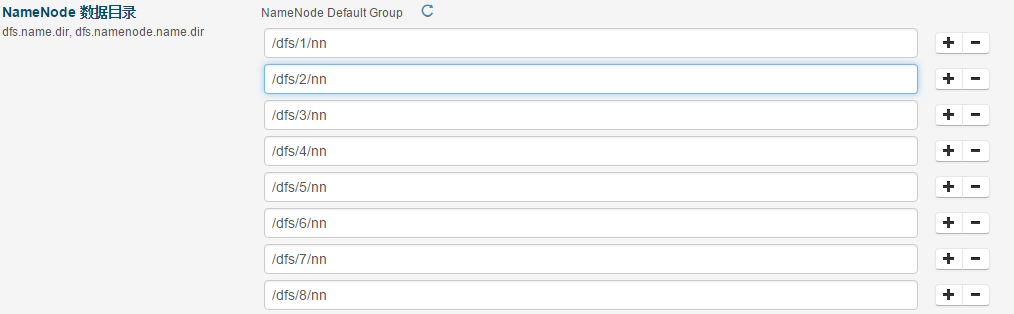
1. 如图3-14，根据服务器中数据磁盘的数量，将“NameNode 数据目录”设置为：/data/disks/1/nn, /data/disks/2/nn, /data/disks/3/nn, /data/disks/4/nn, /data/disks/5/nn, /data/disks/6/nn, /data/disks/7/nn, /data/disks/8/nn;

图 3‑14

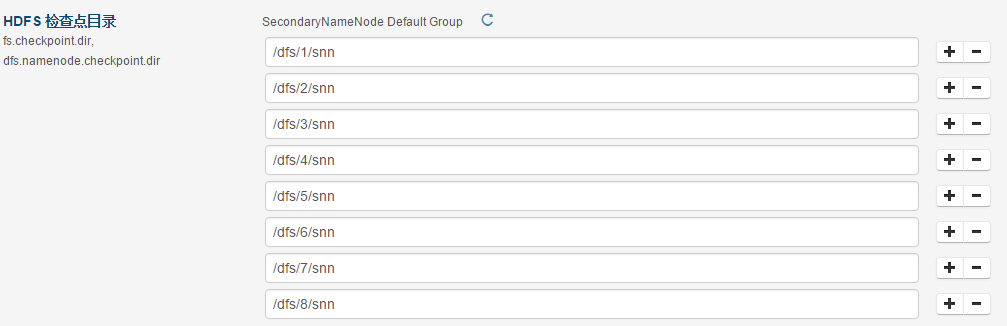
1. 如图3-15，将“HDFS 检查点目录”设置为：/data/disks/1/snn, /data/disks/2/snn, /data/disks/3/snn, /data/disks/4/snn, /data/disks/5/snn, /data/disks/6/snn, /data/disks/7/snn, /data/disks/8/snn

图 3‑15

1. 其他选项根据需要设置，然后点击“继续”。
2. 点击“测试连接”，所有项目测试通过后，点击“继续”。
3. 如图3-16，所有项目“首次运行”成功之后，点击“继续”，集群即可使用。

图 3‑16

### 添加组件Flume

1. 如图3-22示，在“添加服务向导”页面，选择Flume：

图 3‑22

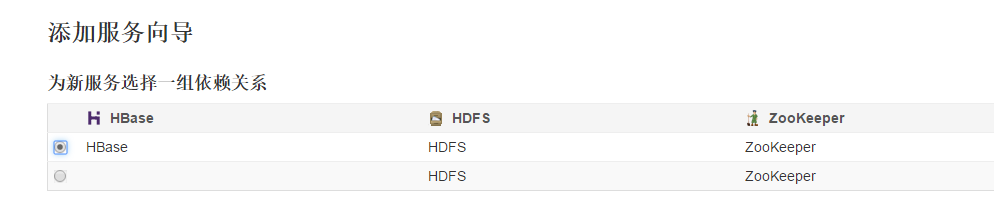
1. 如图3-23所示， 在“依赖关系”中选择HBase、HDFS和Zookeeper：

图 3‑23

1. 如图3-24所示，根据实际情况，分配角色：
2. 完成后，启动Flume服务，如图3-25所示：

图 3‑24

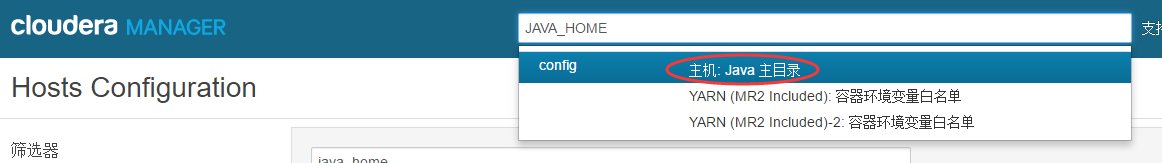


图 3‑25

## 配置Hadoop集群

### 修改JDK路径

1. 在Cloudera Manager主页右上角的搜索框中输入“JAVA\_HOME”，选择“主机：Java主目录”配置项；



1. 将“Java主目录”修改为/usr/java/jdk1.8.0\_60



1. 重启集群



### 为长时间运行的应用程序配置 YARN

长时间运行的应用程序（如 Spark 流式处理作业）需要其他配置，因为默认设置仅允许 hdfs 用户的委派令牌保留最大生存期 7 天，这是不够的。

您可以通过以下方法解决此问题：将 ResourceManager 配置为对应 HDFS NameNode 的代理用户，以便在现有令牌超过其最大生存期时，ResourceManager 可以请求新的令牌。YARN 随后能够代表hdfs 用户继续执行本地化和日志聚合。

在 Cloudera Manager 中配置代理用户，如下所示：

1. 转到 Cloudera Manager Admin Console；
2. 使用**群集**选项卡导航到 **YARN** 服务；
3. 单击**配置**；
4. 在**筛选器**的搜索框中输入yarn.resourcemanager.proxy-user-privileges.enabled；
5. 在右侧“启用 ResourceManger 代理用户权限”的选项框中打钩；
6. 保存设置；
7. 使用**群集**选项卡导航到 **HDFS** 服务；
8. 单击**配置**；
9. 在**范围**中选择**HDFS (服务范围)**；
10. 在**类别**中选择**高级**；
11. 将如下代码加入到**core-site.xml 的群集范围高级配置代码段（安全阀）**属性中：

*<property>*

*<name>hadoop.proxyuser.yarn.hosts</name>*

*<value>\*</value>*

*</property>*

*<property>*

*<name>hadoop.proxyuser.yarn.groups</name>*

*<value>\*</value>*

*</property>*

1. 保存设置；
2. 重启 YARN 和 HDFS 服务。

### 配置YARN应用程序的运行资源

根据以下步骤，进入YARN服务的配置页面。

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**YARN** ；
3. 点击**配置**标签；
4. 在**筛选器**的左下角选择**资源管理**，结果如图3-26所示：

图 3‑26

#### 更改虚拟CPU个数配置

在左侧**筛选器**的搜索框中输入**vcores**，可根据集群规模和应用程序需要，配置以下项目，保存配置之后，重启YARN服务。

|  |  |
| --- | --- |
| 名称 | 配置项 |
| ApplicationMaster 虚拟 CPU 内核 | yarn.app.mapreduce.am.resource.cpu-vcores |
| Map 任务 CPU 虚拟内核 | mapreduce.map.cpu.vcores |
| Reduce 任务 CPU 虚拟内核 | mapreduce.reduce.cpu.vcores |
| 容器虚拟 CPU 内核 | yarn.nodemanager.resource.cpu-vcores |
| 最小容器虚拟 CPU 内核数量 | yarn.scheduler.minimum-allocation-vcores |
| 容器虚拟 CPU 内核增量 | yarn.scheduler.increment-allocation-vcores |
| 最大容器虚拟 CPU 内核数量 | yarn.scheduler.maximum-allocation-vcores |

#### 更改YARN应用程序内存配置

在左侧**筛选器**的搜索框中输入**memory**，可根据集群规模和应用程序需要，配置以下项目，保存配置之后，重启YARN服务。

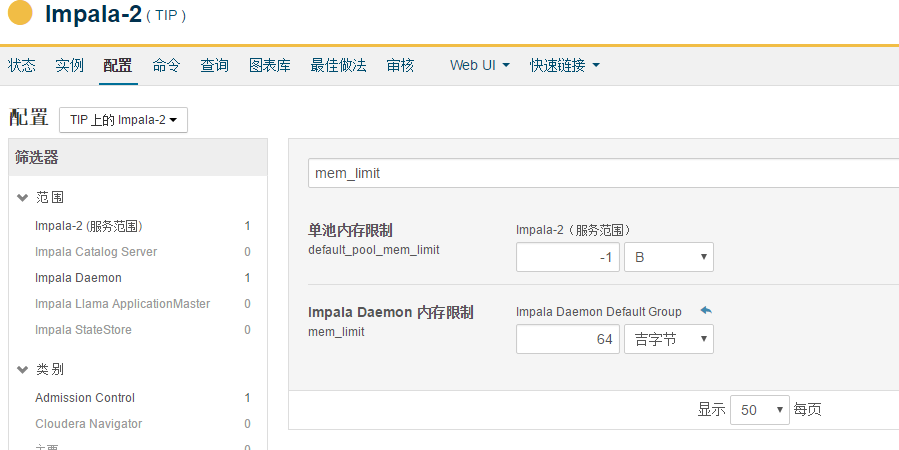
|  |  |
| --- | --- |
| 名称 | 配置项 |
| Map 任务内存 | mapreduce.map.memory.mb |
| Reduce 任务内存 | mapreduce.reduce.memory.mb |
| Cgroup 内存软限制 | memory.soft\_limit\_in\_bytes |
| Cgroup 内存硬限制 | memory.limit\_in\_bytes |
| 容器内存 | yarn.nodemanager.resource.memory-mb |

**注意**：如果内存配置过小，YARN应用程序可能无法运行，以Spark（On Yarn）的example为例，它要求“容器内存”至少1408MB。

### 更改HDFS缓存大小

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**HDFS** ；
3. 点击**配置**标签, 搜索 dfs\_datanode\_max\_locked\_memory；
4. 如下图所示，将**用于缓存的最大内存**改为 1 GB：
5. 保存设置，并重启HDFS服务。

### 更改Impala Daemon内存限制

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**Impala** ；
3. 点击**配置**标签, 搜索 mem\_limit；
4. 如下图所示，将**Impala Daemon 内存限制**改为 64 GB：
5. 保存设置，并重启Impala服务。

## 验证安装

### 查看组件状态

要开始测试，请[启动 Cloudera Manager Admin Console](http://www.cloudera.com/content/cloudera/zh-CN/documentation/core/v5-3-x/topics/cm_intro_start_admin_console.html#cmug_topic_3_1)。登录后，主页应如图3-27所示：



图 3‑27

屏幕左侧是当前正在运行的服务及其状态信息的列表。所有服务运行时应处于**良好运行状况** http://www.cloudera.com/content/cloudera/zh-CN/documentation/core/v5-3-x/images/status_health_good.png。您可以单击每个服务以查看有关每个服务的更多详细信息。您也可以通过检查每个主机的检测信号、运行 MapReduce 作业或与现有 Hue 应用程序集群交互来测试您的安装。

### 检查主机检测信号

检查所有 Agent 是否正在运行的一种方法是查看最后一个检测信号的产生时间。您可以通过单击**主机**选项卡来完成此任务，可在此选项卡中查看所有主机的列表及它们的**最后一个检测信号**的值。默认情况下，每个 Agent 必须每 15 秒成功产生一个检测信号。**最后一个检测信号**的最新值意味着 Server 与 Agent 成功通信。

### 验证HDFS

在集群服务器（NameNode）的Linux Shell中输入以下命令：

*sudo -u hdfs hadoop fs -mkdir /test*

*sudo -u hdfs hadoop fs -ls /*

如果出现如下输出，表示HDFS部署成功：

[root@host01 cdh]# sudo -u hdfs hadoop fs -ls /

Found 4 items

drwxr-xr-x - hbase hbase 0 2015-10-27 11:17 /hbase

drwxr-xr-x - hdfs supergroup 0 2015-10-28 14:28 /test

drwxrwxrwt - hdfs supergroup 0 2015-10-28 14:27 /tmp

drwxr-xr-x - hdfs supergroup 0 2015-10-26 15:06 /user

### 验证YARN

在集群服务器（NameNode）的Linux Shell中输入以下命令：

*sudo -u hdfs hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 10*

如果出现如下输出，表示Yarn部署成功：

*Job Finished in 33.797 seconds*

*Estimated value of Pi is 3.20000000000000000000*

### 验证HBase

1. 在集群服务器（HBase Master）的Linux Shell中输入以下命令，进入HBase Shell：

hbase shell

1. 在HBase Shell中依次输入以下命令：

*create 'people','cf'*

*put 'people','1','cf:name','test1'*

*put 'people','1','cf:age','20'*

*put 'people','2','cf:name','test2'*

*put 'people','2','cf:age','30'*

*get 'people','1'*

1. 如果HBase Shell中输出如下结果，表示HBase安装成功：

*hbase(main):001:0> get 'people','1'*

*COLUMN CELL*

*cf:age timestamp=1445916236613, value=20*

*cf:name timestamp=1445916236542, value=test1*

*2 row(s) in 0.4360 seconds*

### 验证Hive

1. 在集群服务器（Hive Metastore Server）的Linux Shell中输入以下命令，进入Hive Shell：

*hive*

1. 在Hive Shell中输入以下命令，创建表：

*DROP TABLE IF EXISTS people;*

*CREATE EXTERNAL TABLE people (id string,name string,age int) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" =":key,cf:name,cf:age") TBLPROPERTIES("hbase.table.name" = "people");*

*SELECT \* FROM people;*

1. 在Hive Shell中输入以下命令，查看people表中的内容：

*SELECT id,name,age FROM people;*

1. 如果Hive Shell中输出如下结果，表示Hive安装成功：

*hive> SELECT id,name,age FROM people;*

*OK*

*1 test1 20*

*2 test2 30*

*Time taken: 0.423 seconds, Fetched: 2 row(s)*

### 验证Impala

1. 在集群服务器（Impala Daemon）的Linux Shell中输入以下命令，进入Impala Shell：

*impala-shell*

1. 在Impala Shell中输入以下命令，查看people表中的内容：

*invalidate metadata;*

*SELECT id,name,age FROM people;*

1. 如果Impala Shell中输出如下结果，表示Impala安装成功：

*[localhost:21000] > SELECT id,name,age FROM people;*

*Query: select id,name,age FROM people*

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

*| id | name | age |*

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

*| 1 | test1 | 20 |*

*| 2 | test2 | 30 |*

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

*Fetched 2 row(s) in 0.16s*

# 部署TIP相关组件

## 部署数据接收组件

### Flume

1. 停止Flume服务；
2. 在Flume Agent所在的所有机器的Shell中输入以下命令：

*rm -fr /data/flume*

*mkdir -p /data/flume*

*chown -R impala:impala /data/flume*

*rm -fr /var/log/flume-ng*

*mkdir -p /var/log/flume-ng*

*chown -R impala:impala /var/log/flume-ng*

1. 在集群其中一台机器的Shell中输入以下命令：

*sudo -u hdfs hadoop fs mkdir -p /flume/tip*

*sudo -u hdfs hadoop fs -chown -R impala:impala /flume/tip*

1. 修改flume的用户配置，在Flume的配置界面中，将**系统用户**和**系统组**修改为impala



1. 在Flume的配置界面中搜索“Java配置选项”，并增加**--no-reload-conf=true**



1. 修改flume的内存配置：

在Flume的配置界面中，搜索“xmx”，并找到“**Agent 的 Java 堆栈大小（字节）**”配置项将其根据实际情况修改为4GB。

1. 修改flume插件的配置：
   1. 在Cloudera Manager的Flume配置中将“代理名称”修改为：tip\_agent
   2. 在Cloudera Manager的Flume配置中修改“配置文件”配置项（实际是修改flume.conf)，加入以下内容，实际配置根据需要修改，保存过后重启Flume服务。

*#agent config*

*tip\_agent.sources=tip\_http\_source*

*tip\_agent.channels=text\_csv\_channel image\_channel*

*tip\_agent.sinks=text\_csv\_sink image\_sink*

*#sources*

*tip\_agent.sources.tip\_http\_source.type=com.dccs.platform.flume.source.TipHTTPSource*

*tip\_agent.sources.tip\_http\_source.handler=com.dccs.platform.flume.source.TipSourceHandler*

*tip\_agent.sources.tip\_http\_source.bind=0.0.0.0*

*tip\_agent.sources.tip\_http\_source.port=11111*

*tip\_agent.sources.tip\_http\_source.channels=text\_csv\_channel image\_channel*

*# channel selector configuration*

*tip\_agent.sources.tip\_http\_source.selector.type = multiplexing*

*tip\_agent.sources.tip\_http\_source.selector.header = TIP-DATA-TYPE*

*tip\_agent.sources.tip\_http\_source.selector.mapping.CSV = text\_csv\_channel*

*tip\_agent.sources.tip\_http\_source.selector.mapping.IMAGE = image\_channel*

*# TIP CSV channel*

*tip\_agent.channels.text\_csv\_channel.type=com.dccs.platform.flume.channel.TipChannel*

*tip\_agent.channels.text\_csv\_channel.checkpointDir = /data/flume/tip/csv/checkpoint*

*tip\_agent.channels.text\_csv\_channel.dataDirs = /data/flume/tip/csv/data*

*tip\_agent.channels.image\_channel.type=com.dccs.platform.flume.channel.TipChannel*

*tip\_agent.channels.image\_channel.checkpointDir = /data/flume/tip/image/checkpoint*

*tip\_agent.channels.image\_channel.dataDirs = /data/flume/tip/image/data*

*# TipHdfsSink*

*tip\_agent.sinks.text\_csv\_sink.type = com.dccs.platform.flume.sink.TipHdfsSink*

*tip\_agent.sinks.text\_csv\_sink.channel = text\_csv\_channel*

*tip\_agent.sinks.text\_csv\_sink.hdfs.path =/flume/tip*

*tip\_agent.sinks.text\_csv\_sink.hdfs.batchSize = 10*

*tip\_agent.sinks.text\_csv\_sink.zookeeperQuorum=host102,host103,host104*

*#Use dccs Hbase sink*

*tip\_agent.sinks.image\_sink.channel=image\_channel*

*tip\_agent.sinks.image\_sink.type=com.dccs.platform.flume.sink.TipImageSink*

*tip\_agent.sinks.image\_sink.serializer=com.dccs.platform.flume.serializer.AsyncTipImageHBaseSerializer*

*tip\_agent.sinks.image\_sink.zookeeperQuorum=host102,host103,host104*

*tip\_agent.sinks.image\_sink.serializer.zookeeperQuorum=host102,host103,host104*