## Hadoop部署经验总结

Hadoop版本：2.7.2

系统：ubuntu16.04

1. 非安全模式的hadoop集群搭建
2. 软件准备

~~在集群所有机器上安装hadoop2.7.2和java，建议每台机器都安装在相同路径，方便后续设置。~~

在集群master（namenode所在的）机器上安装hadoop2.7.2和java。

1. 修改配置

添加环境变量$HADOOP\_PREFIX=你的hadoop文件夹所在路径。

Hadoop的设置都是通过修改各种配置文件，这些文件可以大致分为三类：

* 只读的默认配置：core-default.xml, hdfs-default.xml, yarn-default.xml and mapred-default.xml
* 根据需要修改的配置文件，都在$HADOOP\_PREFIX/etc/hadoop文件夹中：etc/hadoop/core-site.xml，etc/hadoop/hdfs-site.xml,

etc/hadoop/yarn-site.xml，etc/hadoop/mapred-site.xml

* 环境变量配置文件,也在$HADOOP\_PREFIX/etc/hadoop中：hadoop-env.sh， yarn-env.sh

在hadoop-env.sh中修改$JAVA\_HOME为你的jdk所在路径;

* core-site.xml:

<configuration>  
 <property>  
 <name>fs.defaultFS</name>  
 <value>hdfs://singlenode.ustc.edu:9000</value>  
 </property>  
  
 <property>  
 <name>hadoop.tmp.dir</name>  
 <value>/opt/hadoop-2.7.2/tmp</value>  
 </property>

</configuration>

上面是示例，singlenode.ustc.edu是我的主机全域名，设置应该换为master的全域名（通过hostname -f查看，如果想修改可以修改hosts文件），/opt/hadoop-2.7.2/tmp可以自己设置，默认情况下这是集群数据存放的位置。

* hdfs-site.xml：

<configuration>  
 <property>  
 <name>dfs.replication</name>  
 <value>1</value>  
 </property>  
  
 <property>  
 <name>dfs.webhdfs.enabled</name>  
 <value>true</value>  
 </property>  
  
</configuration>

Dfs.replication是文件的副本数量，因为我搭建的是试验性质，只设置了1,即不备份，一般设置为3,重要数据的话可以更高，但是会增加消耗的存储空间和数据传输时间，dfs.webhdfs.enabled设为true，允许使用webhdfs方式操作hdfs。

* yarn-site.xml：

<configuration>  
  
 <!-- Site specific YARN configuration properties -->  
 <property>  
 <name>yarn.nodemanager.aux-services</name>  
 <value>mapreduce\_shuffle</value>  
 </property>  
  
 </configuration>

* mapred-site.xml:

<configuration>  
 <property>  
 <name>mapreduce.framework.name</name>  
 <value>yarn</value>  
 </property>  
</configuration>

1. 配置ssh无密码登录，hosts和slaves文件

/etc/hosts文件中要包含集群所有机器的host和ip的映射；

$HADOOP\_PREFIX/etc/hadoop/slaves文件中要包含datanode的host或者ip，每个datanode一行，示例：

localhost

datanode1

datanode2

datanode3

ssh的无密码登录指令如下：

ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa

cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

chmod 0600 ~/.ssh/authorized\_keys

这三条指令生成了master主机的公钥id\_dsa.pub和认证文件authorized\_keys，

并将公钥内容加入认证文件，使得master可以ssh无密码登录本机，要让master能无密码登录其他机器，需要将公钥文件内容添加到其他机器的认证文件中。

1. 启动集群
2. 格式化hdfs

$HADOOP\_PREFIX/bin/hdfs namenode -format <cluster\_name>

cluster\_name自己设定

1. 开启hdfs  
    $HADOOP\_PREFIX/sbin/start-dfs.sh
2. 开启yarn  
    $HADOOP\_PREFIX/sbin/start-yarn.sh
3. 开启MapReduce Jobhistory服务  
    $HADOOP\_PREFIX/sbin/mr-jobhistory-daemon.sh \

--config $HADOOP\_CONF\_DIR start historyserver

其中$HADOOP\_CONF\_DIR是hadopp配置文件路径，一般是

$HADOOP\_PREFIX/etc/hadoop

1. 安全模式的hadoop集群搭建
2. 安装配置kerberos

假设我们要搭建的kdc服务器realm为SINGLENODE.COM,服务器所在主机的全域名为host.kdc.com

1. 安装kerberos服务端软件

apt-get install krb5-kdc krb5-admin-server

1. 修改配置文件

对于ubuntu系统，在第一步安装过程中会要求输入realm和主机域名，因此这一步已经完成。对于其他系统，需要修改/etc/krb5.conf:

[libdefaults]

default\_realm = SINGLENODE.COM

[realms]

ATHENA.MIT.EDU = {

kdc = host.kdc.com

admin\_server = host.kdc.com

}

1. 创建数据库并启动kerberos服务器

创建数据库：

kdb5\_util create -s

启动kdc服务器和数据库管理进程：

service krb5-kdc restart

service krb5-admin-server restart

或者

krb5kdc

kadmind

1. 创建管理员principal并添加权限

首先创建acl文件,其文件名和路径可以在/etc/krb5kdc/kdc.conf中查看，一般默认为/etc/krb5kdc/kadmin5.acl,创建后在文件中加上内容：

\*/admin@ATHENA.MIT.EDU \*

然后运行命令创建管理员principal:

kadmin.local

kadmin.local: addprinc admin/admin@SINGLENODE.COM

创建完成后需要重启kerberos服务器，然后可以运行一下命令验证是否正确创建：

krb5kdc

kadmind

kinit admin/admin@SINGLENODE.COM

输入创建时的密码，没有报错就是成功了，可以运行klist命令查看kerberos认证状态。

1. 为hadoop创建principal和keytab

需要为hadoop的每个服务组件创建principal：

| **Service** | **Component** | **Mandatory Principal Name** |
| --- | --- | --- |
| HDFS | NameNode | nn/$FQDN |
| HDFS | NameNode HTTP | HTTP/$FQDN |
| HDFS | SecondaryNameNode | sn/$FQDN |
| HDFS | SecondaryNameNode HTTP | HTTP/$FQDN |
| HDFS | DataNode | dn/$FQDN |
| MR2 | History Server | jhs/$FQDN |
| MR2 | History Server HTTP | HTTP/$FQDN |
| YARN | ResourceManager | rm/$FQDN |
| YARN | NodeManager | nm/$FQDN |
| Oozie | Oozie Server | oozie/$FQDN |
| Oozie | Oozie HTTP | HTTP/$FQDN |
| Hive | Hive Metastore  HiveServer2 | hive/$FQDN |
| Hive | WebHCat | HTTP/$FQDN |
| HBase | MasterServer | hbase/$FQDN |
| HBase | RegionServer | hbase/$FQDN |
| Storm | Nimbus server  DRPC daemon | nimbus/$FQDN \*\* |
| Storm | Storm UI daemon  Storm Logviewer daemon  Nodes running process controller (such as Supervisor) | storm/$FQDN \*\* |
| Kafka | KafkaServer | kafka/$FQDN |
| Hue | Hue Interface | hue/$FQDN |
| ZooKeeper | ZooKeeper | zookeeper/$FQDN |
| JournalNode Server\* | JournalNode | jn/$FQDN |
| Gateway | Knox | knox/$FQDN |

第三列中的$FQDN要改为该服务组件所在机器的全域名。

然后创建keytab文件保证无密码认证：

| **Component** | **Principal Name** | **Mandatory Keytab File Name** |
| --- | --- | --- |
| NameNode | nn/$FQDN | nn.service.keytab |
| NameNode HTTP | HTTP/$FQDN | spnego.service.keytab |
| SecondaryNameNode | sn/$FQDN | sn.service.keytab |
| SecondaryNameNode HTTP | HTTP/$FQDN | spnego.service.keytab |
| DataNode | dn/$FQDN | dn.service.keytab |
| MR2 History Server | jhs/$FQDN | nm.service.keytab |
| MR2 History Server HTTP | HTTP/$FQDN | spnego.service.keytab |
| YARN | rm/$FQDN | rm.service.keytab |
| YARN | nm/$FQDN | nm.service.keytab |
| Oozie Server | oozie/$FQDN | oozie.service.keytab |
| Oozie HTTP | HTTP/$FQDN | spnego.service.keytab |
| Hive Metastore  HiveServer2 | hive/$FQDN | hive.service.keytab |
| WebHCat | HTTP/$FQDN | spnego.service.keytab |
| HBase Master Server | hbase/$FQDN | hbase.service.keytab |
| HBase RegionServer | hbase/$FQDN | hbase.service.keytab |
| Storm | storm/$FQDN | storm.service.keytab |
| Kafka | kafka/$FQDN | kafka.service.keytab |
| Hue | hue/$FQDN | hue.service.keytab |
| ZooKeeper | zookeeper/$FQDN | zk.service.keytab |
| Journal Server\* | jn/$FQDN | jn.service.keytab |
| Knox Gateway\*\* | knox/$FQDN | knox.service.keytab |

添加principal和创建keytab的命令示例如下，需要先输入kadmin.local：

addprinc -randkey dn/$FQDN

ktadd -k $keytab\_file\_name dn/$FQDN

$keytab\_file\_name改为keytab的全路径，例如/etc/security/keytab/dn.service.keytab

1. 添加用户和分组

给集群所有机器新建用户组hadoop，然后添加以下用户：

hdfs,yarn,mapred,http

具体命令为：

groupadd hadoop

useradd hdfs -g hadoop

1. 设置hadoop的安全模式

主要就是修改配置文件,下面列出了各个文件需要添加的属性：

* core-site.xml：

<property>  
 <name>hadoop.security.authorization</name>  
 <value>true</value>  
 <description>Is service-level authorization enabled?</description>  
</property>  
  
<property>  
 <name>hadoop.security.authentication</name>  
 <value>kerberos</value>  
 <description>Possible values are simple (no authentication)

, and kerberos  
 </description>  
</property>

<property>

<name>hadoop.security.auth\_to\_local</name>

<value>

RULE:[2:$1@$0]([snjd]n@SINGLENODE.COM)s/.\*/hdfs/

RULE:[2:$1@$0](hdfs@SINGLENODE.COM)s/.\*/hdfs/

RULE:[2:$1@$0](HTTP@SINGLENODE.COM)s/.\*/http/

RULE:[2:$1@$0]([nr]m@SINGLENODE.COM)s/.\*/yarn/

RULE:[2:$1@$0](jhs@SINGLENODE.COM)s/.\*/mapred/

RULE:[1:$1@$0](^hue@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^sentry@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^hive@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^oozie@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^yarn@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^mapred@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^hdfs@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^zookeeper@.\*)s/^.\*$/nobody/

RULE:[1:$1@$0](^httpfs@.\*$)s/^.\*$/nobody/

RULE:[1:$1@$0](^HTTP@.\*$)s/^.\*$/nobody/

RULE:[2:$1/$2@$0](^.\*$)s/^.\*$/nobody/

DEFAULT

</value>

<description>Maps kerberos principals to local user names</description>

</property>

hadoop.security.auth\_to\_local这个属性可以将kerberos的principal映射到系统用户名，具体的编写规则可以参考http://web.mit.edu/Kerberos/krb5-latest/doc/admin/conf\_files/krb5\_conf.html中关于auth\_to\_local的说明。

* hdfs-site.xml

<property>  
 <name>dfs.web.authentication.kerberos.principal</name>  
 <value>HTTP/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <name>dfs.web.authentication.kerberos.keytab</name>  
 <value>/etc/security/keytab/spnego.service.keytab</value>  
</property>  
  
<property>  
 <name>dfs.permissions.enabled</name>  
 <value>true</value>  
 <description>  
 If "true", enable permission checking in HDFS.  
 If "false", permission checking is turned off,  
 but all other behavior is unchanged.  
 Switching from one parameter value to the other does not change the mode,  
 owner or group of files or directories.  
 </description>  
</property>  
  
<property>  
 <name>dfs.permissions.superusergroup</name>  
 <value>hadoop</value>  
 <description>The name of the group of super-users.</description>  
</property>  
  
<property>  
 <name>dfs.block.access.token.enable</name>  
 <value>true</value>  
 <descripion>  
 If "true", access tokens are used as capabilities for accessing datanodes.  
  
 If "false", no access tokens are checked on accessing datanodes.  
 </descripion>  
</property>  
  
<property>  
 <name>dfs.namenode.kerberos.principal</name>  
 <value>nn/\_HOST@SINGLENODE.COM</value>  
</property>  
<property>  
 <name>dfs.secondary.namenode.kerberos.principal</name>  
 <value>sn/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <name>dfs.datanode.kerberos.principal</name>  
 <value>dn/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <name>dfs.namenode.keytab.file</name>  
 <value>/etc/security/keytab/nn.service.keytab</value>  
</property>  
  
<property>  
 <name>dfs.secondary.namenode.keytab.file</name>  
 <value>/etc/security/keytab/sn.service.keytab</value>  
</property>

<property>  
 <name>dfs.datanode.keytab.file</name>  
 <value>/etc/security/keytab/dn.service.keytab</value>  
</property>  
  
  
<property>  
 <name>dfs.namenode.https-address</name>  
 <value>singlenode.ustc.edu:50470</value>  
 <description>The namenode secure http server address and port.</description>  
</property>  
  
<property>  
 <name>dfs.https.port</name>  
 <value>50470</value>  
 <description>The namenode secure http server address and port.</description>  
</property>  
  
<property>  
 <name>dfs.datanode.data.dir.perm</name>  
 <value>700</value>  
 <description>Permissions for the directories on on the local filesystem where  
 the DFS data node store its blocks. The permissions can either be octal or  
 symbolic.  
 </description>  
</property>  
  
<property>  
 <name>dfs.cluster.administrators</name>  
 <value>hdfs</value>  
 <description>This configuration is used to control who can access the  
 default servlets in the namenode, etc.  
 </description>  
</property>  
  
<property>  
 <name>dfs.namenode.kerberos.internal.spnego.principal</name>  
 <value>${dfs.web.authentication.kerberos.principal}</value>  
</property>  
  
<property>  
 <name>dfs.secondary.namenode.kerberos.internal.spnego.principal</name>  
 <value>${dfs.web.authentication.kerberos.principal}</value>  
</property>  
  
<property>  
 <name>dfs.datanode.address</name>  
 <value>0.0.0.0:1004</value>  
 <description>  
 The datanode server address and port for data transfer.  
 </description>  
</property>  
  
<property>  
 <name>dfs.datanode.http.address</name>  
 <value>0.0.0.0:1006</value>  
 <description>  
 The datanode http server address and port.  
 </description>  
</property>  
  
<property>  
 <name>dfs.http.policy</name>  
 <value>HTTP\_ONLY</value>  
 <description>Decide if HTTPS(SSL) is supported on HDFS  
 This configures the HTTP endpoint for HDFS daemons:  
 The following values are supported:  
 - HTTP\_ONLY : Service is provided only on http  
 - HTTPS\_ONLY : Service is provided only on https  
 - HTTP\_AND\_HTTPS : Service is provided both on http and https  
 </description>  
</property>

* yarn-site.xml

<property>  
 <description>The hostname of the RM.</description>  
 <name>yarn.resourcemanager.hostname</name>  
 <value>resourcemanager.ustc.edu</value>  
</property>  
  
<property>  
 <description>The Kerberos principal for the resource manager.</description>  
 <name>yarn.resourcemanager.principal</name>  
 <value>rm/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <description>The keytab for the resource manager.</description>  
 <name>yarn.resourcemanager.keytab</name>  
 <value>/etc/security/keytab/rm.service.keytab</value>  
</property>  
  
<property>  
 <description>The kerberos principal for the node manager.</description>  
 <name>yarn.nodemanager.principal</name>  
 <value>nm/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <description>Keytab for NM.</description>  
 <name>yarn.nodemanager.keytab</name>  
 <value>/etc/security/keytab/nm.service.keytab</value>  
</property>

* mapred-site.xml

<property>  
 <name>mapreduce.jobhistory.keytab</name>  
 <value>/etc/security/keytab/jhs.service.keytab</value>  
</property>  
  
<property>  
 <name>mapreduce.jobhistory.principal</name>  
 <value>jhs/\_HOST@SINGLENODE.COM</value>  
</property>  
  
<property>  
 <name>mapreduce.jobhistory.webapp.spnego-keytab-file</name>  
 <value>/etc/security/keytab/spnego.service.keytab</value>  
</property>  
  
<property>  
 <name>mapreduce.jobhistory.webapp.spnego-principal</name>  
 <value>HTTP/\_HOST@SINGLENODE.COM</value>  
</property>

|  |  |
| --- | --- |
| hadoop.http.authentication.simple.anonymous.allowed | false |
| hadoop.http.authentication.signature.secret.file | /etc/security/http\_secret |
| hadoop.http.authentication.type | kerberos |
| hadoop.http.authentication.kerberos.keytab | /etc/security/keytabs/spnego.service.keytab |
| hadoop.http.authentication.kerberos.principal | HTTP/\_HOST@SINGLENODE.COM |
| hadoop.http.filter.initializers | org.apache.hadoop.security.AuthenticationFilterInitializer |
| hadoop.http.authentication.cookie.domain | hortonworks.local |