

尚硅谷大数据项目之尚品汇（用户认证）

(作者：尚硅谷研究院)

版本：V4.1

第 1 章 Kerberos 部署

1.1 Kerberos 概述

1.1.1 什么是 Kerberos

Kerberos 是一种计算机网络认证协议，用来在非安全网络中，对个人通信以安全的手段进行**身份认证**。这个词又指麻省理工学院为这个协议开发的一套计算机软件。软件设计上采用客户端/服务器结构，并且能够进行相互认证，即客户端和服务端均可对对方进行身份认证。可以用于防止窃听、防止重放攻击、保护数据完整性等场合，是一种应用对称密钥体制进行密钥管理的系统。

1.1.2 Kerberos 术语

Kerberos 中有以下一些概念需要了解：

- 1) KDC (Key Distribute Center)：密钥分发中心，负责存储用户信息，管理发放票据。
- 2) Realm：Kerberos 所管理的一个领域或范围，称之为一个 Realm。
- 3) Rrincipal：Kerberos 所管理的一个用户或者一个服务，可以理解为 Kerberos 中保存的一个账号，其格式通常如下：`primary/instance@realm`
- 4) keytab：Kerberos 中的用户认证，可通过密码或者密钥文件证明身份，keytab 指密钥文件。

 Kerberos认证原理

1.2.1 安装 Kerberos 相关服务

服务端主机执行以下安装命令

客户端主机执行以下安装命令

1.2.2 修改配置文件

1.服务端主机 (hadoop102)

修改/var/kerberos/krb5kdc/kdc.conf 文件，内容如下

修改如下内容

```
[kdcdefaults]
kdc_ports = 88
kdc_tcp_ports = 88

[realms]
EXAMPLE.COM = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
```

```
admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
supported_encypes = aes256-cts:normal aes128-cts:normal
des3-hmac-sha1:normal arcfour-hmac:normal
camellia256-cts:normal camellia128-cts:normal
des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
}
```

2.客户端主机（所有主机）

修改/etc/krb5.conf 文件

```
[root@hadoop102 ~]# vim /etc/krb5.conf
[root@hadoop103 ~]# vim /etc/krb5.conf
[root@hadoop104 ~]# vim /etc/krb5.conf
```

内容如下

```
# Configuration snippets may be placed in this directory as well
includedir /etc/krb5.conf.d/

[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = FILE:/etc/pki/tls/certs/ca-bundle.crt
default_realm = EXAMPLE.COM
#default_ccache_name = KEYRING:persistent:%{uid}

[realms]
EXAMPLE.COM = {
    kdc = hadoop102
    admin_server = hadoop102
}

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
```

1.2.3 初始化 KDC 数据库

在服务端主机（hadoop102）执行以下命令，并根据提示输入密码。

```
[root@hadoop102 ~]# kdb5_util create -s
```

1.2.4 修改管理员权限配置文件

在服务端主机（hadoop102）修改/var/kerberos/krb5kdc/kadm5.acl 文件，内容如下

```
*/admin@EXAMPLE.COM *
```

1.2.5 启动 Kerberos 相关服务

在主节点（hadoop102）启动 KDC，并配置开机自启

```
[root@hadoop102 ~]# systemctl start krb5kdc
[root@hadoop102 ~]# systemctl enable krb5kdc
```

在主节点（hadoop102）启动 Kadmin，该服务为 KDC 数据库访问入口

```
[root@hadoop102 ~]# systemctl start kadmin
[root@hadoop102 ~]# systemctl enable kadmin
```

1.2.6 创建 Kerberos 管理员用户

在 KDC 所在主机（hadoop102），执行以下命令，并按照提示输入密码

```
[root@hadoop102 ~]# kadmin.local -q "addprinc admin/admin"
```

1.3 Kerberos 使用概述

1.3.1 Kerberos 数据库操作

1. 登录数据库

1) 本地登录（**无需认证**）

```
[root@hadoop102 ~]# kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local:
```

2) 远程登录（**需进行主体认证，认证操作见下文**）

```
[root@hadoop102 ~]# kadmin
Authenticating as principal admin/admin@EXAMPLE.COM with
password.
Password for admin/admin@EXAMPLE.COM:
kadmin:
```

退出输入：exit

2. 创建 Kerberos 主体

登录数据库，输入以下命令，并按照提示输入密码

```
kadmin.local: addprinc test
```

也可通过以下 shell 命令直接创建主体

```
[root@hadoop102 ~]# kadmin.local -q "addprinc test"
```

3. 修改主体密码

```
kadmin.local :cpw test
```

4. 查看所有主体

```
kadmin.local: list_principals
K/M@EXAMPLE.COM
admin/admin@EXAMPLE.COM
kadmin/admin@EXAMPLE.COM
kadmin/angepw@EXAMPLE.COM
kadmin/hadoop105@EXAMPLE.COM
kiprop/hadoop105@EXAMPLE.COM
krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

1.3.2 Kerberos 认证操作

1. 密码认证

- 1) 使用 kinit 进行主体认证，并按照提示输入密码

```
[root@hadoop102 ~]# kinit test
Password for test@EXAMPLE.COM:
```

- 2) 查看认证凭证

```
[root@hadoop102 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@EXAMPLE.COM

Valid starting      Expires            Service principal
10/27/2019          18:23:57          10/28/2019 18:23:57
krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 11/03/2019 18:23:57
```

2. 密钥文件认证

- 1) 生成主体 test 的 keytab 文件到指定目录/root/test.keytab

```
[root@hadoop102 ~]# kadmin.local -q "xst -norandkey -k /root/test.keytab test@EXAMPLE.COM"
```

注: -norandkey 的作用是声明不随机生成密码，若不加该参数，会导致之前的密码失效。

- 2) 使用 keytab 进行认证

```
[root@hadoop102 ~]# kinit -kt /root/test.keytab test
```

- 3) 查看认证凭证

```
[root@hadoop102 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@EXAMPLE.COM

Valid starting      Expires            Service principal
08/27/19            15:41:28          08/28/19 15:41:28
krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 08/27/19 15:41:28
```

3. 销毁凭证

```
[root@hadoop102 ~]# kdestroy
[root@hadoop102 ~]# klist
klist: No credentials cache found (ticket cache
FILE:/tmp/krb5cc_0)
```

第 2 章 创建 Hadoop 系统用户

为 Hadoop 开启 Kerberos，需为不同服务准备不同的用户，启动服务时需要使用相应的用户。须在**所有节点**创建以下用户和用户组。

User: Group	Daemons
hdfs: hadoop	NameNode, Secondary NameNode, JournalNode, DataNode
yarn: hadoop	ResourceManager, NodeManager
mapred: hadoop	MapReduce JobHistory Server

创建 hadoop 组

```
[root@hadoop102 ~]# groupadd hadoop
[root@hadoop103 ~]# groupadd hadoop
[root@hadoop104 ~]# groupadd hadoop
```

创建各用户并设置密码

```
[root@hadoop102 ~]# useradd hdfs -g hadoop
[root@hadoop102 ~]# echo hdfs | passwd --stdin hdfs

[root@hadoop102 ~]# useradd yarn -g hadoop
[root@hadoop102 ~]# echo yarn | passwd --stdin yarn

[root@hadoop102 ~]# useradd mapred -g hadoop
[root@hadoop102 ~]# echo mapred | passwd --stdin mapred

[root@hadoop103 ~]# useradd hdfs -g hadoop
[root@hadoop103 ~]# echo hdfs | passwd --stdin hdfs

[root@hadoop103 ~]# useradd yarn -g hadoop
[root@hadoop103 ~]# echo yarn | passwd --stdin yarn

[root@hadoop103 ~]# useradd mapred -g hadoop
[root@hadoop103 ~]# echo mapred | passwd --stdin mapred

[root@hadoop104 ~]# useradd hdfs -g hadoop
[root@hadoop104 ~]# echo hdfs | passwd --stdin hdfs

[root@hadoop104 ~]# useradd yarn -g hadoop
[root@hadoop104 ~]# echo yarn | passwd --stdin yarn

[root@hadoop104 ~]# useradd mapred -g hadoop
[root@hadoop104 ~]# echo mapred | passwd --stdin mapred
```

第 3 章 Hadoop Kerberos 配置

3.1 为 Hadoop 各服务创建 Kerberos 主体 (Principal)

主体格式如下: ServiceName/HostName@REALM, 例如 dn/hadoop102@EXAMPLE.COM

1. 各服务所需主体如下

环境: 3 台节点, 主机名分别为 hadoop102, hadoop103, hadoop104

服务	所在主机	主体 (Principal)
NameNode	hadoop102	nn/hadoop102
DataNode	hadoop102	dn/hadoop102
DataNode	hadoop103	dn/hadoop103
DataNode	hadoop104	dn/hadoop104
Secondary NameNode	hadoop104	sn/hadoop104
ResourceManager	hadoop103	rm/hadoop103
NodeManager	hadoop102	nm/hadoop102
NodeManager	hadoop103	nm/hadoop103
NodeManager	hadoop104	nm/hadoop104

JobHistory Server	hadoop102	jhs/hadoop102
Web UI	hadoop102	HTTP/hadoop102
Web UI	hadoop103	HTTP/hadoop103
Web UI	hadoop104	HTTP/hadoop104

2.创建主体说明

1) 路径准备

为服务创建的主体，需要通过密钥文件 **keytab** 文件进行认证，故需为各服务准备一个安全的路径用来存储 **keytab** 文件。

```
[root@hadoop102 ~]# mkdir /etc/security/keytab/  
[root@hadoop102 ~]# chown -R root:hadoop /etc/security/keytab/  
[root@hadoop102 ~]# chmod 770 /etc/security/keytab/
```

2) 管理员主体认证

为执行创建主体的语句，需登录 **Kerberos** 数据库客户端，登录之前需先使用 **Kerberos** 的管理员用户进行认证，执行以下命令并根据提示输入密码。

```
[root@hadoop102 ~]# kinit admin/admin
```

3) 登录数据库客户端

```
[root@hadoop102 ~]# kadmin
```

4) 执行创建主体的语句

```
kadmin: addprinc -randkey test/test  
kadmin: xst -k /etc/security/keytab/test.keytab test/test
```

说明：

(1) **addprinc test/test**：作用是新建主体

addprinc：增加主体

-randkey：密码随机，因 **hadoop** 各服务均通过 **keytab** 文件认证，故密码可随机生成

test/test：新增的主体

(2) **xst -k /etc/security/keytab/test.keytab test/test**：作用是将主体的密钥写入 **keytab** 文件

xst：将主体的密钥写入 **keytab** 文件

-k /etc/security/keytab/test.keytab：指明 **keytab** 文件路径和文件名

test/test：主体

(3) 为方便创建主体，可使用如下命令

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc  
-randkey test/test"  
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k  
/etc/security/keytab/test.keytab test/test"
```

说明：

-p：主体

-w: 密码

-q: 执行语句

(4) 操作主体的其他命令，可参考官方文档，地址如下：

http://web.mit.edu/kerberos/krb5-current/doc/admin/admin_commands/kadmin_local.html#commands

3.创建主体

1) 在所有节点创建 keytab 文件目录

```
[root@hadoop102 ~]# mkdir /etc/security/keytab/
[root@hadoop102 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop102 ~]# chmod 770 /etc/security/keytab/

[root@hadoop103 ~]# mkdir /etc/security/keytab/
[root@hadoop103 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop103 ~]# chmod 770 /etc/security/keytab/

[root@hadoop104 ~]# mkdir /etc/security/keytab/
[root@hadoop104 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop104 ~]# chmod 770 /etc/security/keytab/
```

2) 以下命令在 hadoop102 节点执行

NameNode (hadoop102)

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey nn/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/nn.service.keytab nn/hadoop102"
```

DataNode (hadoop102)

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey dn/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/dn.service.keytab dn/hadoop102"
```

NodeManager (hadoop102)

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey nm/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/nm.service.keytab nm/hadoop102"
```

JobHistory Server (hadoop102)

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey jhs/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/jhs.service.keytab jhs/hadoop102"
```

Web UI (hadoop102)

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey HTTP/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/spnego.service.keytab HTTP/hadoop102"
```

2) 以下命令在 hadoop103 执行

ResourceManager (hadoop103)


```
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey rm/hadoop103"
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/rm.service.keytab rm/hadoop103"
```

DataNode (hadoop103)

```
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey dn/hadoop103"
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/dn.service.keytab dn/hadoop103"
```

NodeManager (hadoop103)

```
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey nm/hadoop103"
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/nm.service.keytab nm/hadoop103"
```

Web UI (hadoop103)

```
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey HTTP/hadoop103"
[root@hadoop103 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/spnego.service.keytab HTTP/hadoop103"
```

3) 以下命令在 hadoop104 执行

DataNode (hadoop104)

```
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey dn/hadoop104"
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/dn.service.keytab dn/hadoop104"
```

Secondary NameNode (hadoop104)

```
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey sn/hadoop104"
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/sn.service.keytab sn/hadoop104"
```

NodeManager (hadoop104)

```
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey nm/hadoop104"
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/nm.service.keytab nm/hadoop104"
```

Web UI (hadoop104)

```
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"addprinc
-randkey HTTP/hadoop104"
[root@hadoop104 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/spnego.service.keytab HTTP/hadoop104"
```

4.修改所有节点 keytab 文件的所有者和访问权限

```
[root@hadoop102 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop102 ~]# chmod 660 /etc/security/keytab/*

[root@hadoop103 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop103 ~]# chmod 660 /etc/security/keytab/*

[root@hadoop104 ~]# chown -R root:hadoop /etc/security/keytab/
[root@hadoop104 ~]# chmod 660 /etc/security/keytab/*
```

3.2 修改 Hadoop 配置文件

需要修改的内容如下，修改完毕需要分发所改文件。

1.core-site.xml

```
[root@hadoop102 ~]# vim /opt/module/hadoop-3.1.3/etc/hadoop/core-site.xml
增加以下内容

<!-- Kerberos 主体到系统用户的映射机制 -->
<property>
  <name>hadoop.security.auth_to_local.mechanism</name>
  <value>MIT</value>
</property>

<!-- Kerberos 主体到系统用户的具体映射规则 -->
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1/$2@$0]([ndj]n\/*@EXAMPLE\.COM)s/*/*hdfs/
    RULE:[2:$1/$2@$0]([rn]m\/*@EXAMPLE\.COM)s/*/*yarn/
    RULE:[2:$1/$2@$0]([jhs]\/*@EXAMPLE\.COM)s/*/*mapred/
    DEFAULT
  </value>
</property>

<!-- 启用 Hadoop 集群 Kerberos 安全认证 -->
<property>
  <name>hadoop.security.authentication</name>
  <value>kerberos</value>
</property>

<!-- 启用 Hadoop 集群授权管理 -->
<property>
  <name>hadoop.security.authorization</name>
  <value>true</value>
</property>

<!-- Hadoop 集群间 RPC 通讯设为仅认证模式 -->
<property>
  <name>hadoop.rpc.protection</name>
  <value>authentication</value>
</property>
```

2.hdfs-site.xml

```
[root@hadoop102 ~]# vim /opt/module/hadoop-3.1.3/etc/hadoop/hdfs-site.xml
增加以下内容

<!-- 访问 DataNode 数据块时需通过 Kerberos 认证 -->
<property>
  <name>dfs.block.access.token.enable</name>
  <value>true</value>
</property>
```

```
<!-- NameNode 服务的 Kerberos 主体, _HOST 会自动解析为服务所在的主机名 -->
<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@EXAMPLE.COM</value>
</property>

<!-- NameNode 服务的 Kerberos 密钥文件路径 -->
<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytab/nn.service.keytab</value>
</property>

<!-- Secondary NameNode 服务的 Kerberos 主体 -->
<property>
  <name>dfs.secondary.namenode.keytab.file</name>
  <value>/etc/security/keytab/sn.service.keytab</value>
</property>

<!-- Secondary NameNode 服务的 Kerberos 密钥文件路径 -->
<property>
  <name>dfs.secondary.namenode.kerberos.principal</name>
  <value>sn/_HOST@EXAMPLE.COM</value>
</property>

<!-- NameNode Web 服务的 Kerberos 主体 -->
<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@EXAMPLE.COM</value>
</property>

<!-- WebHDFS REST 服务的 Kerberos 主体 -->
<property>
  <name>dfs.web.authentication.kerberos.principal</name>
  <value>HTTP/_HOST@EXAMPLE.COM</value>
</property>

<!-- Secondary NameNode Web UI 服务的 Kerberos 主体 -->
<property>
  <name>dfs.secondary.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@EXAMPLE.COM</value>
</property>

<!-- Hadoop Web UI 的 Kerberos 密钥文件路径 -->
<property>
  <name>dfs.web.authentication.kerberos.keytab</name>
  <value>/etc/security/keytab/spnego.service.keytab</value>
</property>

<!-- DataNode 服务的 Kerberos 主体 -->
<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@EXAMPLE.COM</value>
</property>
```

```
<!-- DataNode 服务的 Kerberos 密钥文件路径 -->
<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytab/dn.service.keytab</value>
</property>

<!-- 配置 NameNode Web UI 使用 HTTPS 协议 -->
<property>
  <name>dfs.http.policy</name>
  <value>HTTPS_ONLY</value>
</property>

<!-- 配置 DataNode 数据传输保护策略为仅认证模式 -->
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>
```

3.yarn-site.xml

```
[root@hadoop102 ~]# vim /opt/module/hadoop-3.1.3/etc/hadoop/yarn-site.xml
增加以下内容

<!-- Resource Manager 服务的 Kerberos 主体 -->
<property>
  <name>yarn.resourcemanager.principal</name>
  <value>rm/_HOST@EXAMPLE.COM</value>
</property>

<!-- Resource Manager 服务的 Kerberos 密钥文件 -->
<property>
  <name>yarn.resourcemanager.keytab</name>
  <value>/etc/security/keytab/rm.service.keytab</value>
</property>

<!-- Node Manager 服务的 Kerberos 主体 -->
<property>
  <name>yarn.nodemanager.principal</name>
  <value>nm/_HOST@EXAMPLE.COM</value>
</property>

<!-- Node Manager 服务的 Kerberos 密钥文件 -->
<property>
  <name>yarn.nodemanager.keytab</name>
  <value>/etc/security/keytab/nm.service.keytab</value>
</property>
```

4.mapred-site.xml

```
[root@hadoop102 ~]# vim /opt/module/hadoop-3.1.3/etc/hadoop/mapred-site.xml
增加以下内容

<!-- 历史服务器的 Kerberos 主体 -->
<property>
  <name>mapreduce.jobhistory.keytab</name>
```

```
<value>/etc/security/keytab/jhs.service.keytab</value>
</property>

<!-- 历史服务器的 Kerberos 密钥文件 -->
<property>
  <name>mapreduce.jobhistory.principal</name>
  <value>jhs/_HOST@EXAMPLE.COM</value>
</property>
```

(5) 分发以上修改的配置文件

```
[root@hadoop102 ~]# xsync
/opt/module/hadoop-3.1.3/etc/hadoop/core-site.xml
[root@hadoop102 ~]# xsync
/opt/module/hadoop-3.1.3/etc/hadoop/hdfs-site.xml
[root@hadoop102 ~]# xsync
/opt/module/hadoop-3.1.3/etc/hadoop/yarn-site.xml
[root@hadoop102 ~]# xsync
/opt/module/hadoop-3.1.3/etc/hadoop/mapred-site.xml
```

3.3 配置 HDFS 使用 HTTPS 安全传输协议

1. 生成密钥对

Keytool 是 java 数据证书的管理工具，使用户能够管理自己的公/私钥对及相关证书。

-keystore 指定密钥库的名称及位置(产生的各类信息将存在.keystore 文件中)

-genkey(或者-genkeypair) 生成密钥对

-alias 为生成的密钥对指定别名，如果没有默认是 mykey

-keyalg 指定密钥的算法 RSA/DSA 默认是 DSA

1) 生成 keystore 的密码及相应信息的密钥库

```
[root@hadoop102 ~]# keytool -genkey -keystore
/etc/security/keytab/keystore -alias jetty -keyalg RSA
```

输入密钥库口令:

再次输入新口令:

您的名字与姓氏是什么?

[Unknown]:

您的组织单位名称是什么?

[Unknown]:

您的组织名称是什么?

[Unknown]:

您所在的城市或区域名称是什么?

[Unknown]:

您所在的省/市/自治区名称是什么?

[Unknown]:

该单位的双字母国家/地区代码是什么?

[Unknown]:

CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown 是否正确?

[否]: y

输入 <jetty> 的密钥口令
(如果和密钥库口令相同, 按回车):
再次输入新口令:

2) 修改 keystore 文件的所有者和访问权限

```
[root@hadoop102 ~]# chown -R root:hadoop /etc/security/keytab/keystore
[root@hadoop102 ~]# chmod 660 /etc/security/keytab/keystore
```

注意:

- (1) 密钥库的密码至少 6 个字符, 可以是纯数字或者字母或者数字和字母的组合等等
- (2) 确保 hdfs 用户 (HDFS 的启动用户) 具有对所生成 keystore 文件的读权限

3) 将该证书分发到集群中的每台节点的相同路径

```
[root@hadoop102 ~]# xsync /etc/security/keytab/keystore
```

4) 修改 hadoop 配置文件 ssl-server.xml.example,

该文件位于 \$HADOOP_HOME/etc/hadoop 目录

修改文件名为 ssl-server.xml

```
[root@hadoop102 ~]# mv $HADOOP_HOME/etc/hadoop/ssl-server.xml.example $HADOOP_HOME/etc/hadoop/ssl-server.xml
```

修改以下内容

```
[root@hadoop102 ~]# vim $HADOOP_HOME/etc/hadoop/ssl-server.xml
```

修改以下参数

```
<!-- SSL 密钥库路径 -->
<property>
  <name>ssl.server.keystore.location</name>
  <value>/etc/security/keytab/keystore</value>
</property>

<!-- SSL 密钥库密码 -->
<property>
  <name>ssl.server.keystore.password</name>
  <value>123456</value>
</property>

<!-- SSL 可信任密钥库路径 -->
<property>
  <name>ssl.server.truststore.location</name>
  <value>/etc/security/keytab/keystore</value>
</property>

<!-- SSL 密钥库中密钥的密码 -->
<property>
  <name>ssl.server.keystore.keypassword</name>
  <value>123456</value>
</property>

<!-- SSL 可信任密钥库密码 -->
<property>
```

```
<name>ssl.server.truststore.password</name>
<value>123456</value>
</property>
```

5) 分发 ssl-server.xml 文件

```
[root@hadoop102 ~]# xsync $HADOOP_HOME/etc/hadoop/ssl-server.xml
```

3.4 配置 Yarn 使用 LinuxContainerExecutor

1) 修改**所有节点**的 container-executor 所有者和权限，要求其所有者为 root，所有组为 hadoop（启动 NodeManger 的 yarn 用户的所属组），权限为 6050。其默认路径为 \$HADOOP_HOME/bin

```
[root@hadoop102 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/bin/container-executor
[root@hadoop102 ~]# chmod 6050
/opt/module/hadoop-3.1.3/bin/container-executor

[root@hadoop103 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/bin/container-executor
[root@hadoop103 ~]# chmod 6050
/opt/module/hadoop-3.1.3/bin/container-executor

[root@hadoop104 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/bin/container-executor
[root@hadoop104 ~]# chmod 6050
/opt/module/hadoop-3.1.3/bin/container-executor
```

2) 修改**所有节点**的 container-executor.cfg 文件的所有者和权限，要求该文件及其所有的上级目录的所有者均为 root，所有组为 hadoop（启动 NodeManger 的 yarn 用户的所属组），权限为 400。其默认路径为 \$HADOOP_HOME/etc/hadoop

```
[root@hadoop102 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg
[root@hadoop102 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc/hadoop
[root@hadoop102 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc
[root@hadoop102 ~]# chown root:hadoop /opt/module/hadoop-3.1.3
[root@hadoop102 ~]# chown root:hadoop /opt/module
[root@hadoop102 ~]# chmod 400
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg

[root@hadoop103 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg
[root@hadoop103 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc/hadoop
[root@hadoop103 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc
[root@hadoop103 ~]# chown root:hadoop /opt/module/hadoop-3.1.3
[root@hadoop103 ~]# chown root:hadoop /opt/module
[root@hadoop103 ~]# chmod 400
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg

[root@hadoop104 ~]# chown root:hadoop
```

```
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg
[root@hadoop104 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc/hadoop
[root@hadoop104 ~]# chown root:hadoop
/opt/module/hadoop-3.1.3/etc
[root@hadoop104 ~]# chown root:hadoop /opt/module/hadoop-3.1.3
[root@hadoop104 ~]# chown root:hadoop /opt/module
[root@hadoop104 ~]# chmod 400
/opt/module/hadoop-3.1.3/etc/hadoop/container-executor.cfg
```

3) 修改\$HADOOP_HOME/etc/hadoop/container-executor.cfg

```
[root@hadoop102 ~]# vim
$HADOOP_HOME/etc/hadoop/container-executor.cfg
```

内容如下

```
yarn.nodemanager.linux-container-executor.group=hadoop
banned.users=hdfs,yarn,mapred
min.user.id=1000
allowed.system.users=
feature.tc.enabled=false
```

4) 修改\$HADOOP_HOME/etc/hadoop/yarn-site.xml 文件

```
[root@hadoop102 ~]# vim $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

增加以下内容

```
<!-- 配置 Node Manager 使用 LinuxContainerExecutor 管理 Container -->
<property>
  <name>yarn.nodemanager.container-executor.class</name>

  <value>org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor</value>
</property>

<!-- 配置 Node Manager 的启动用户的所属组 -->
<property>
  <name>yarn.nodemanager.linux-container-executor.group</name>
  <value>hadoop</value>
</property>

<!-- LinuxContainerExecutor 脚本路径 -->
<property>
  <name>yarn.nodemanager.linux-container-executor.path</name>

  <value>/opt/module/hadoop-3.1.3/bin/container-executor</value>
</property>
```

5) 分发 container-executor.cfg 和 yarn-site.xml 文件

```
[root@hadoop102 ~]# xsync
$HADOOP_HOME/etc/hadoop/container-executor.cfg
[root@hadoop102 ~]# xsync $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

第 4 章 安全模式下启动 Hadoop 集群

4.1 修改特定本地路径权限

local	\$HADOOP_LOG_DIR	hdfs:hadoop	drwxrwxr-x
-------	------------------	-------------	------------

local	dfs.namenode.name.dir	hdfs:hadoop	drwx-----
local	dfs.datanode.data.dir	hdfs:hadoop	drwx-----
local	dfs.namenode.checkpoint.dir	hdfs:hadoop	drwx-----
local	yarn.nodemanager.local-dirs	yarn:hadoop	drwxrwxr-x
local	yarn.nodemanager.log-dirs	yarn:hadoop	drwxrwxr-x

1) \$HADOOP_LOG_DIR (所有节点)

该变量位于 `hadoop-env.sh` 文件，默认值为 `${HADOOP_HOME}/logs`

```
[root@hadoop102 ~]# chown hdfs:hadoop /opt/module/hadoop-3.1.3/logs/
[root@hadoop102 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/

[root@hadoop103 ~]# chown hdfs:hadoop /opt/module/hadoop-3.1.3/logs/
[root@hadoop103 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/

[root@hadoop104 ~]# chown hdfs:hadoop /opt/module/hadoop-3.1.3/logs/
[root@hadoop104 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/
```

2) dfs.namenode.name.dir (NameNode 节点)

该参数位于 `hdfs-site.xml` 文件，默认值为 `file://${hadoop.tmp.dir}/dfs/name`

```
[root@hadoop102 ~]# chown -R hdfs:hadoop /opt/module/hadoop-3.1.3/data/dfs/name/
[root@hadoop102 ~]# chmod 700 /opt/module/hadoop-3.1.3/data/dfs/name/
```

3) dfs.datanode.data.dir (DataNode 节点)

该参数为于 `hdfs-site.xml` 文件，默认值为 `file://${hadoop.tmp.dir}/dfs/data`

```
[root@hadoop102 ~]# chown -R hdfs:hadoop /opt/module/hadoop-3.1.3/data/dfs/data/
[root@hadoop102 ~]# chmod 700 /opt/module/hadoop-3.1.3/data/dfs/data/

[root@hadoop103 ~]# chown -R hdfs:hadoop /opt/module/hadoop-3.1.3/data/dfs/data/
[root@hadoop103 ~]# chmod 700 /opt/module/hadoop-3.1.3/data/dfs/data/

[root@hadoop104 ~]# chown -R hdfs:hadoop /opt/module/hadoop-3.1.3/data/dfs/data/
[root@hadoop104 ~]# chmod 700 /opt/module/hadoop-3.1.3/data/dfs/data/
```

4) dfs.namenode.checkpoint.dir (SecondaryNameNode 节点)

该参数位于 `hdfs-site.xml` 文件，默认值为 `file://${hadoop.tmp.dir}/dfs/namesecondary`

```
[root@hadoop104 ~]# chown -R hdfs:hadoop /opt/module/hadoop-3.1.3/data/dfs/namesecondary/
[root@hadoop104 ~]# chmod 700 /opt/module/hadoop-3.1.3/data/dfs/namesecondary/
```

5) yarn.nodemanager.local-dirs (NodeManager 节点)

该参数位于 yarn-site.xml 文件，默认值为 file://\${hadoop.tmp.dir}/nm-local-dir

```
[root@hadoop102 ~]# chown -R yarn:hadoop /opt/module/hadoop-3.1.3/data/nm-local-dir/
[root@hadoop102 ~]# chmod -R 775 /opt/module/hadoop-3.1.3/data/nm-local-dir/

[root@hadoop103 ~]# chown -R yarn:hadoop /opt/module/hadoop-3.1.3/data/nm-local-dir/
[root@hadoop103 ~]# chmod -R 775 /opt/module/hadoop-3.1.3/data/nm-local-dir/

[root@hadoop104 ~]# chown -R yarn:hadoop /opt/module/hadoop-3.1.3/data/nm-local-dir/
[root@hadoop104 ~]# chmod -R 775 /opt/module/hadoop-3.1.3/data/nm-local-dir/
```

6) yarn.nodemanager.log-dirs (NodeManager 节点)

该参数位于 yarn-site.xml 文件，默认值为 \$HADOOP_LOG_DIR/userlogs

```
[root@hadoop102 ~]# chown yarn:hadoop /opt/module/hadoop-3.1.3/logs/userlogs/
[root@hadoop102 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/userlogs/

[root@hadoop103 ~]# chown yarn:hadoop /opt/module/hadoop-3.1.3/logs/userlogs/
[root@hadoop103 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/userlogs/

[root@hadoop104 ~]# chown yarn:hadoop /opt/module/hadoop-3.1.3/logs/userlogs/
[root@hadoop104 ~]# chmod 775 /opt/module/hadoop-3.1.3/logs/userlogs/
```

4.2 启动 HDFS

需要注意的是，启动不同服务时需要使用对应的用户

1. 单点启动

(1) 启动 NameNode

```
[root@hadoop102 ~]# sudo -i -u hdfs hdfs --daemon start namenode
```

(2) 启动 DataNode

```
[root@hadoop102 ~]# sudo -i -u hdfs hdfs --daemon start datanode
[root@hadoop103 ~]# sudo -i -u hdfs hdfs --daemon start datanode
[root@hadoop104 ~]# sudo -i -u hdfs hdfs --daemon start datanode
```

(3) 启动 SecondaryNameNode

```
[root@hadoop104 ~]# sudo -i -u hdfs hdfs --daemon start secondarynamenode
```

说明：

- -i: 重新加载环境变量
- -u: 以特定用户的身份执行后续命令

2.群起

- 1) 在主节点 (hadoop102) 配置 **hdfs** 用户到所有节点的免密登录。
- 2) 修改主节点 (hadoop102) 节点的\$HADOOP_HOME/sbin/start-dfs.sh 脚本, 在顶部增加以下环境变量。

```
[root@hadoop102 ~]# vim $HADOOP_HOME/sbin/start-dfs.sh
```

在顶部增加如下内容

```
HDFS_DATANODE_USER=hdfs
HDFS_NAMENODE_USER=hdfs
HDFS_SECONDARYNAMENODE_USER=hdfs
```

注: \$HADOOP_HOME/sbin/stop-dfs.sh 也需在顶部增加上述环境变量才可使用。

- 3) 以 root 用户执行群起脚本, 即可启动 HDFS 集群。

```
[root@hadoop102 ~]# start-dfs.sh
```

3.查看 HDFS web 页面

访问地址为 <https://hadoop102:9871>

4.3 修改 HDFS 特定路径访问权限

hdfs	/	hdfs:hadoop	drwxr-xr-x
hdfs	/tmp	hdfs:hadoop	drwxrwxrwt
hdfs	/user	hdfs:hadoop	drwxrwxr-x
hdfs	yarn.nodemanager.remote-app-log-dir	yarn:hadoop	drwxrwxrwt
hdfs	mapreduce.jobhistory.intermediate-done-dir	mapred:hadoop	drwxrwxrwt
hdfs	mapreduce.jobhistory.done-dir	mapred:hadoop	drwxrwx---

说明:

若上述路径不存在, 需手动创建

- 1) 创建 hdfs/hadoop 主体, 执行以下命令并按照提示输入密码

```
[root@hadoop102 ~]# kadmin.local -q "addprinc hdfs/hadoop"
```

- 2) 认证 hdfs/hadoop 主体, 执行以下命令并按照提示输入密码

```
[root@hadoop102 ~]# kinit hdfs/hadoop
```

- 3) 按照上述要求修改指定路径的所有者和权限

- (1) 修改 /、/tmp、/user 路径

```
[root@hadoop102 ~]# hadoop fs -chown hdfs:hadoop / /tmp /user
[root@hadoop102 ~]# hadoop fs -chmod 755 /
[root@hadoop102 ~]# hadoop fs -chmod 1777 /tmp
[root@hadoop102 ~]# hadoop fs -chmod 775 /user
```

- (2) 参数 yarn.nodemanager.remote-app-log-dir 位于 yarn-site.xml 文件, 默认值/tmp/logs

```
[root@hadoop102 ~]# hadoop fs -chown yarn:hadoop /tmp/logs
[root@hadoop102 ~]# hadoop fs -chmod 1777 /tmp/logs
```

- (3) 参数 mapreduce.jobhistory.intermediate-done-dir 位于 mapred-site.xml 文件, 默认值为/tmp/hadoop-yarn/staging/history/done_intermediate, 需保证该路径的所有上级目录(除/tmp)

更多 **Java** - 大数据 - 前端 - **python** 人工智能资料下载, 可百度访问: 尚硅谷官网

的所有者均为 `mapred`，所属组为 `hadoop`，权限为 `770`

```
[root@hadoop102 ~]# hadoop fs -chown -R mapred:hadoop /tmp/hadoop-yarn/staging/history/done_intermediate
[root@hadoop102 ~]# hadoop fs -chmod -R 1777 /tmp/hadoop-yarn/staging/history/done_intermediate

[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/staging/history/
[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/staging/
[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/

[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/staging/history/
[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/staging/
[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/
```

(4) 参数 `mapreduce.jobhistory.done-dir` 位于 `mapred-site.xml` 文件，默认值为 `/tmp/hadoop-yarn/staging/history/done`，需保证该路径的所有上级目录（除 `tmp`）的所有者均为 `mapred`，所属组为 `hadoop`，权限为 `770`

```
[root@hadoop102 ~]# hadoop fs -chown -R mapred:hadoop /tmp/hadoop-yarn/staging/history/done
[root@hadoop102 ~]# hadoop fs -chmod -R 750 /tmp/hadoop-yarn/staging/history/done

[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/staging/history/
[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/staging/
[root@hadoop102 ~]# hadoop fs -chown mapred:hadoop /tmp/hadoop-yarn/

[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/staging/history/
[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/staging/
[root@hadoop102 ~]# hadoop fs -chmod 770 /tmp/hadoop-yarn/
```

4.4 启动 Yarn

1. 单点启动

启动 ResourceManager

```
[root@hadoop103 ~]# sudo -i -u yarn yarn --daemon start resourcemanager
```

启动 NodeManager

```
[root@hadoop102 ~]# sudo -i -u yarn yarn --daemon start nodemanager
[root@hadoop103 ~]# sudo -i -u yarn yarn --daemon start nodemanager
[root@hadoop104 ~]# sudo -i -u yarn yarn --daemon start nodemanager
```

2. 群起

1) 在 Yarn 主节点（`hadoop103`）配置 **yarn** 用户到所有节点的免密登录。

2) 修改主节点 (hadoop103) 的\$HADOOP_HOME/sbin/start-yarn.sh, 在顶部增加以下环境变量。

```
[root@hadoop103 ~]# vim $HADOOP_HOME/sbin/start-yarn.sh
在顶部增加如下内容
```

```
YARN_RESOURCEMANAGER_USER=yarn
YARN_NODEMANAGER_USER=yarn
```

注: stop-yarn.sh 也需在顶部增加上述环境变量才可使用。

3) 以 root 用户执行\$HADOOP_HOME/sbin/start-yarn.sh 脚本即可启动 yarn 集群。

```
[root@hadoop103 ~]# start-yarn.sh
```

3.访问 Yarn web 页面

访问地址为 <http://hadoop103:8088>

4.5 启动 HistoryServer

1.启动历史服务器

```
[root@hadoop102 ~]# sudo -i -u mapred mapred --daemon start
historyserver
```

2.查看历史服务器 web 页面

访问地址为 <http://hadoop102:19888>

第 5 章 安全集群使用说明

5.1 用户要求

1.具体要求

以下使用说明均基于普通用户, 安全集群对用户有以下要求:

- 1) 集群中的每个节点都需要创建该用户
- 2) 该用户需要属于 hadoop 用户组
- 3) 需要创建该用户对应的 Kerberos 主体

2.实操

此处以 atguigu 用户为例, 具体操作如下

- 1) 创建用户 (存在可跳过), 须在所有节点执行

```
[root@hadoop102 ~]# useradd atguigu
[root@hadoop102 ~]# echo atguigu | passwd --stdin atguigu

[root@hadoop103 ~]# useradd atguigu
[root@hadoop103 ~]# echo atguigu | passwd --stdin atguigu

[root@hadoop104 ~]# useradd atguigu
[root@hadoop104 ~]# echo atguigu | passwd --stdin atguigu
```

2) 加入 hadoop 组, 须在所有节点执行

```
[root@hadoop102 ~]# usermod -a -G hadoop atguigu
[root@hadoop103 ~]# usermod -a -G hadoop atguigu
[root@hadoop104 ~]# usermod -a -G hadoop atguigu
```

3) 创建主体

```
[root@hadoop102 ~]# kadmin -p admin/admin -wadmin -q"addprinc -pw
atguigu atguigu"
```

5.2 访问 HDFS 集群文件

5.2.1 Shell 命令

1. 认证

```
[atguigu@hadoop102 ~]$ kinit atguigu
```

2. 查看当前认证用户

```
[atguigu@hadoop102 ~]$ kinit atguigu
```

```
[atguigu@hadoop102 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: atguigu@EXAMPLE.COM
```

Valid starting	Expires	Service principal
2021-04-07T21:20:39	2021-04-08T21:20:39	krbtgt/EXAMPLE.COM@EXAMPLE.COM

3. 执行命令

```
[atguigu@hadoop102 ~]$ hadoop fs -ls /
```

```
[atguigu@hadoop102 ~]$ hadoop fs -ls /
Found 10 items
drwxr-xr-x - atguigu supergroup          0 2021-01-13 10:48 /hbase
drwxr-xr-x - atguigu supergroup          0 2021-01-13 11:24 /kylin
drwxr-xr-x - atguigu supergroup          0 2020-11-26 19:09 /origin_data
drwxr-xr-x - atguigu supergroup          0 2021-04-07 10:35 /spark-history
drwxr-xr-x - atguigu supergroup          0 2020-12-28 10:21 /spark-jars
drwxr-xr-x - atguigu supergroup          0 2021-01-18 23:15 /system
drwxr-xr-x - atguigu supergroup          0 2020-12-31 08:50 /test
drwxrwxrwt - hdfs      hadoop           0 2021-01-15 09:04 /tmp
drwxrwxrwx - hdfs      hadoop           0 2021-04-07 14:30 /user
drwxr-xr-x - atguigu supergroup          0 2020-12-28 14:38 /warehouse
```

4. 注销认证

```
[atguigu@hadoop102 ~]$ kdestroy
```

5. 再次执行查看命令

```
[atguigu@hadoop102 ~]$ hadoop fs -ls /
```

```
[atguigu@hadoop102 ~]$ hadoop fs -ls /
2021-04-07 21:22:51,819 WARN ipc.Client: Exception encountered while
connecting to the server : org.apache.hadoop.security.AccessControlEx
ception: Client cannot authenticate via:[TOKEN, KERBEROS]
ls: DestHost:destPort hadoop102:8020 , LocalHost:localPort hadoop102/
192.168.10.102:0. Failed on local exception: java.io.IOException: org
.apache.hadoop.security.AccessControlException: Client cannot authent
icate via:[TOKEN, KERBEROS]
```

5.2.2 web 页面

1. 安装 Kerberos 客户端

下载地址：<http://web.mit.edu/kerberos/dist/kfw/4.1/kfw-4.1-amd64.msi>

1) 下载之后按照提示安装

2) 编辑 C:\ProgramData\MIT\Kerberos5\krb5.ini 文件，内容如下

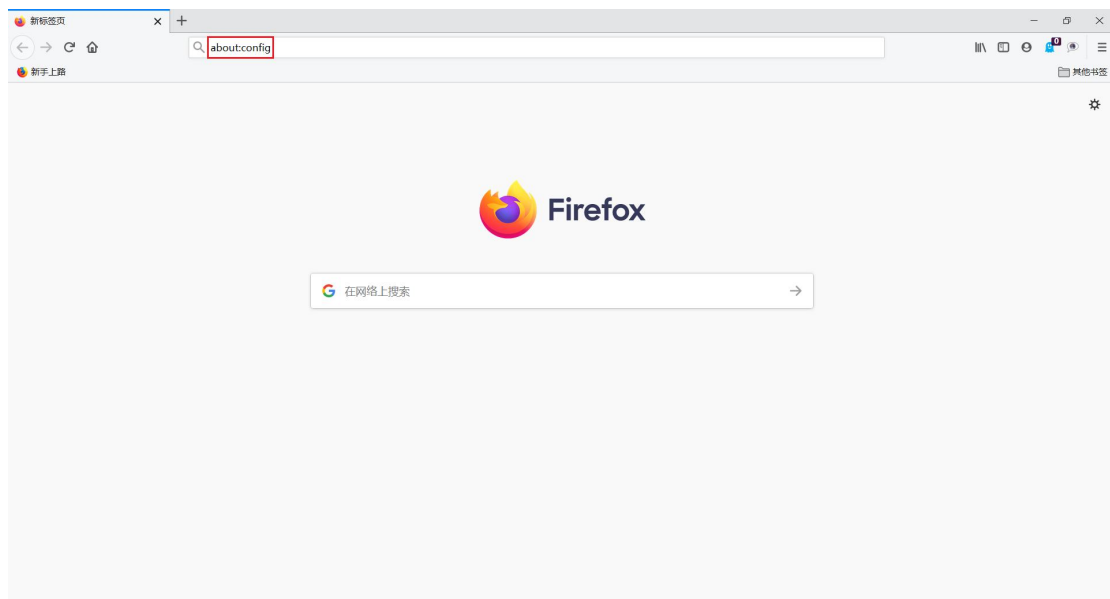
```
[libdefaults]
    dns_lookup_realm = false
    ticket_lifetime = 24h
    forwardable = true
    rdns = false
    default_realm = EXAMPLE.COM

[realms]
    EXAMPLE.COM = {
        kdc = hadoop102
        admin_server = hadoop102
    }

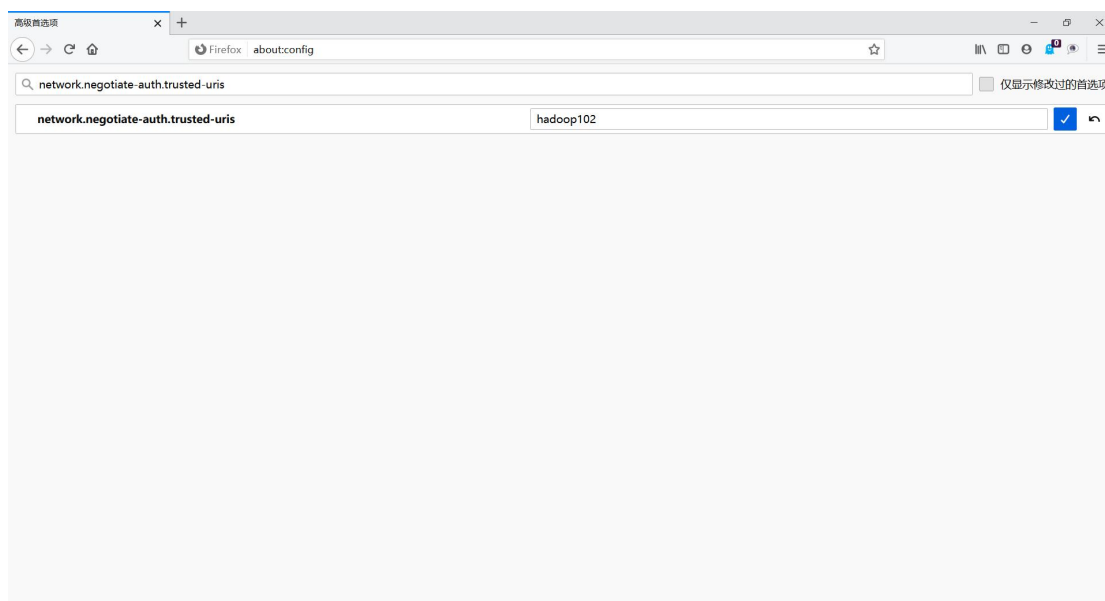
[domain_realm]
    .EXAMPLE.COM = EXAMPLE.COM
```

2. 配置火狐浏览器

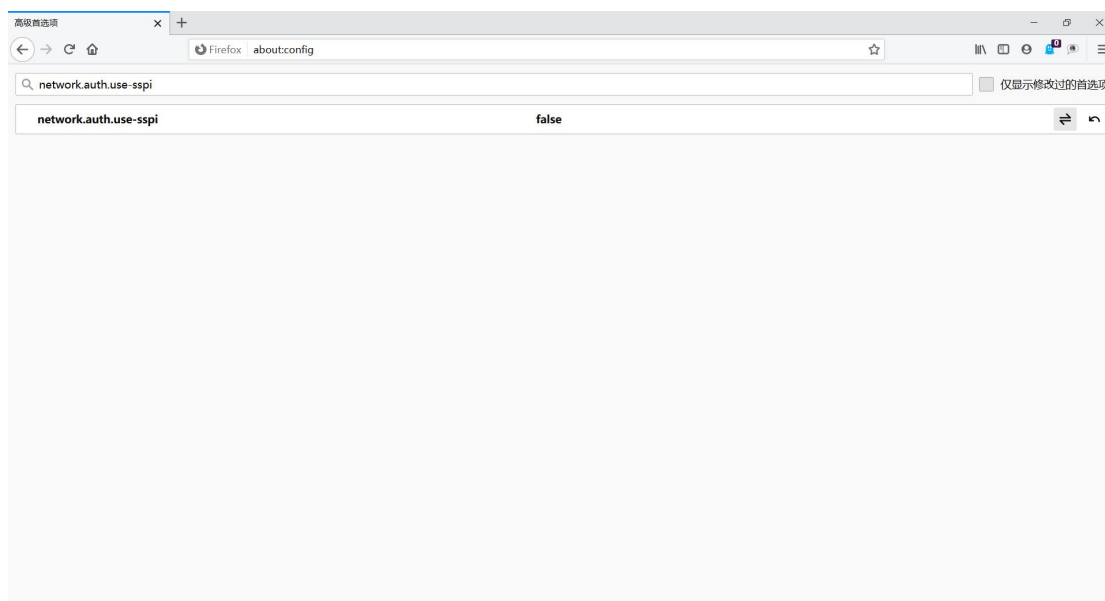
1) 打开浏览器，在地址栏输入“about:config”，点击回车



2) 搜索“network.negotiate-auth.trusted-uris”，修改值为要访问的主机名（hadoop102）

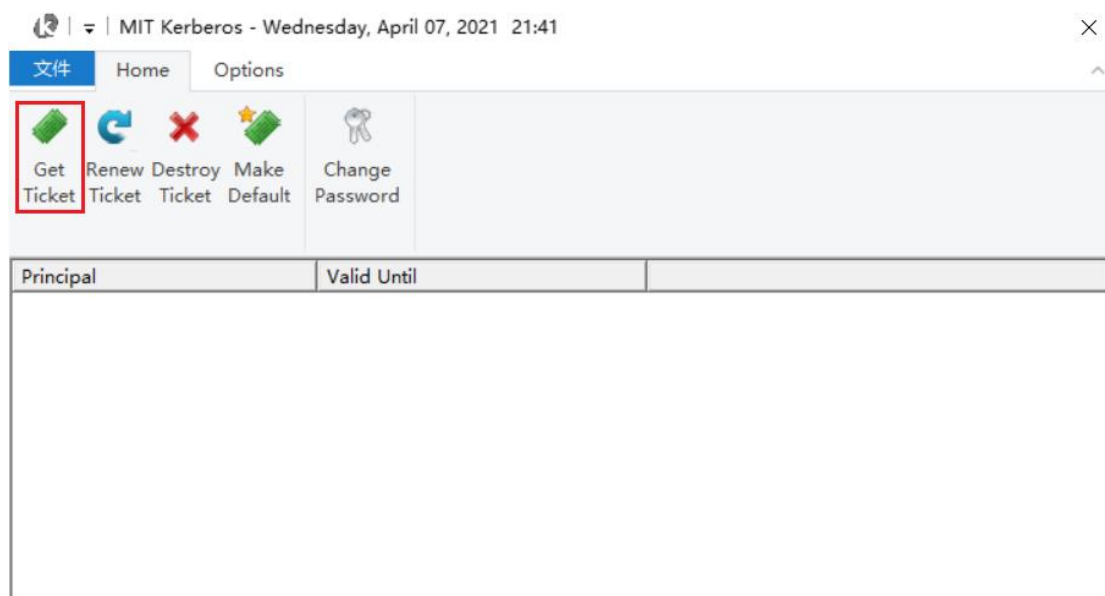


3) 搜索“network.auth.use-sspi”，双击将值变为 false

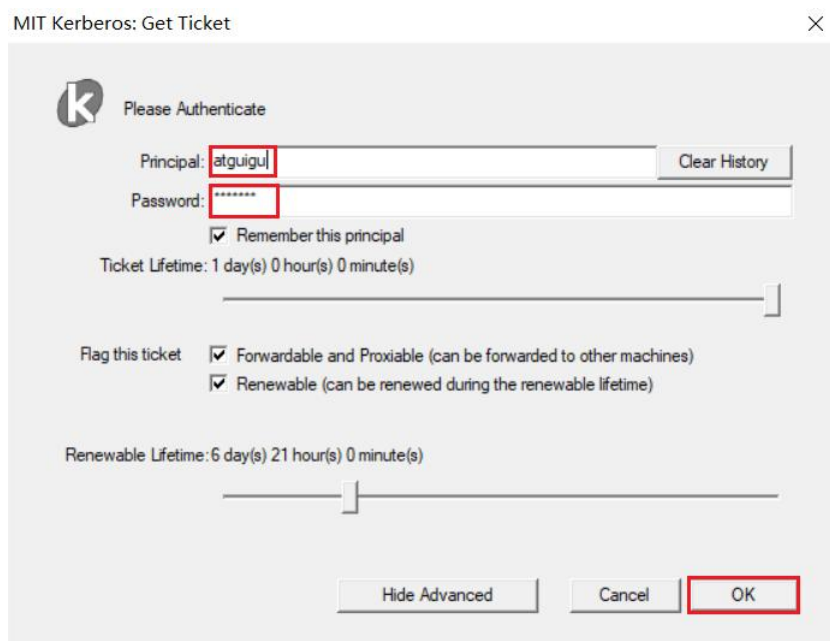


3.认证

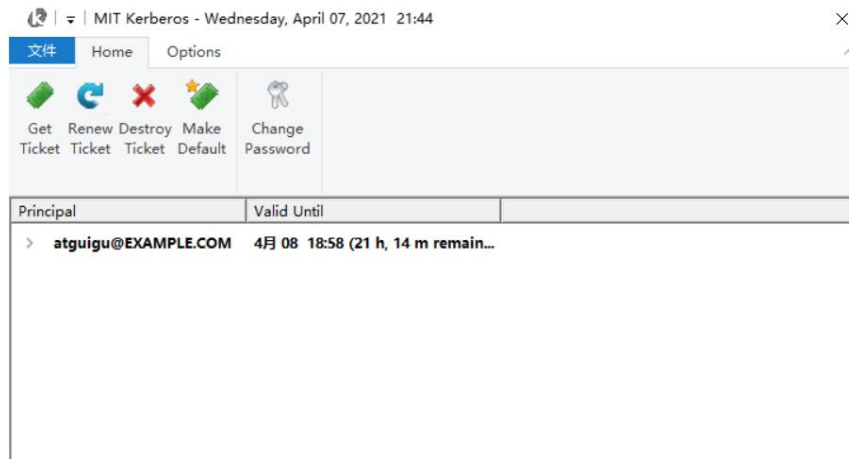
1) 启动 Kerberos 客户端，点击 Get Ticket



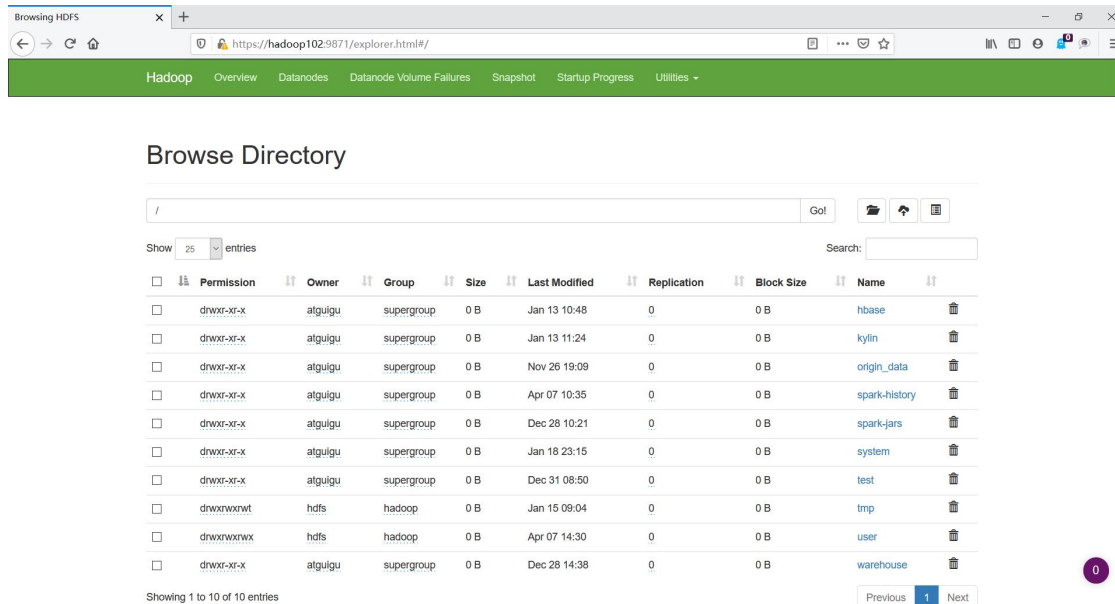
2) 输入主体名和密码，点击 OK



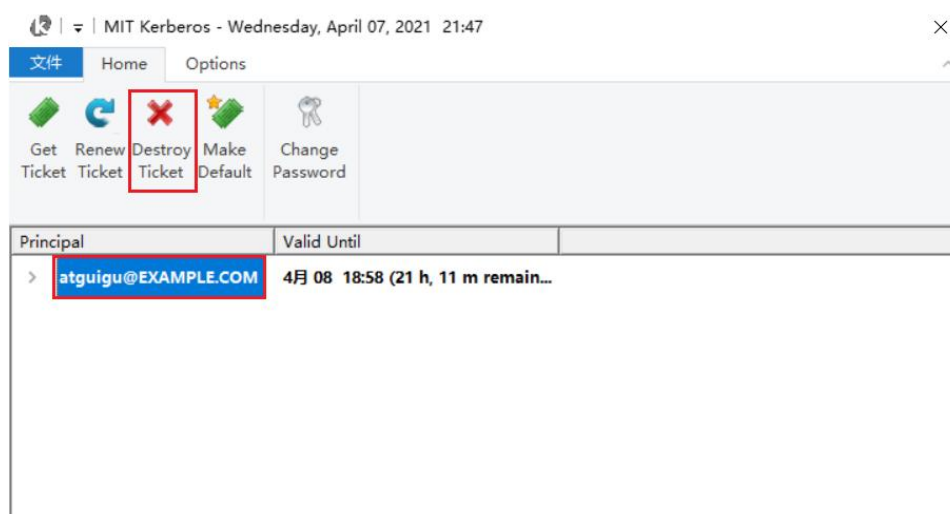
3) 认证成功



4.访问 HDFS

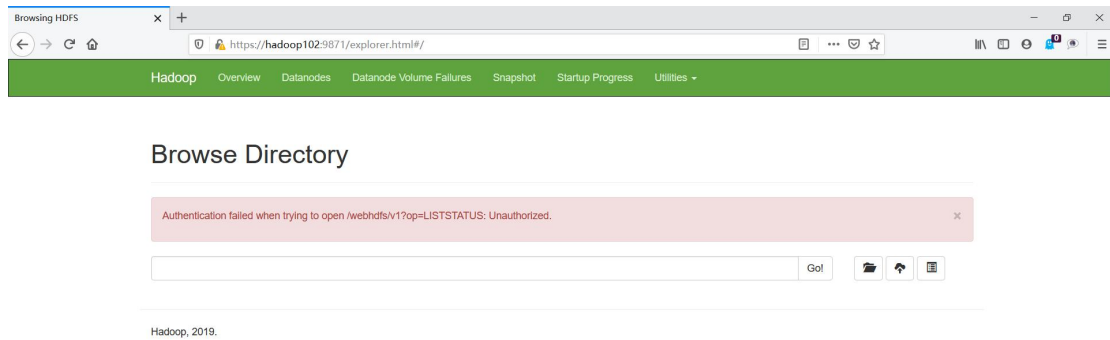


5.注销认证



6.重启浏览器，再次访问 HDFS

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网



5.3 提交 MapReduce 任务

1. 认证

```
[atguigu@hadoop102 ~]$ kinit atguigu
```

2. 提交任务

```
[atguigu@hadoop102 ~]$ hadoop jar /opt/module/hadoop-3.1.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar pi 1 1
```

第 6 章 Hive 用户认证配置

6.1 前置要求

6.1.1 Hadoop 集群启动 Kerberos 认证

按照上述步骤为 Hadoop 集群开启 Kerberos 安全认证。

6.1.2 创建 Hive 系统用户和 Kerberos 主体

1. 创建系统用户

```
[root@hadoop102 ~]# useradd hive -g hadoop
[root@hadoop102 ~]# echo hive | passwd --stdin hive

[root@hadoop103 ~]# useradd hive -g hadoop
[root@hadoop103 ~]# echo hive | passwd --stdin hive

[root@hadoop104 ~]# useradd hive -g hadoop
[root@hadoop104 ~]# echo hive | passwd --stdin hive
```

2. 创建 Kerberos 主体并生成 keytab 文件

创建 hive 用户的 Kerberos 主体

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc
```

```
-randkey hive/hadoop102"
```

在 Hive 所部署的节点生成 keytab 文件

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k  
/etc/security/keytab/hive.service.keytab hive/hadoop102"
```

3. 修改 keytab 文件所有者和访问权限

```
[root@hadoop102 ~]# chown -R root:hadoop /etc/security/keytab/  
[root@hadoop102 ~]# chmod 660  
/etc/security/keytab/hive.service.keytab
```

6.2 配置认证

1. 修改 \$HIVE_HOME/conf/hive-site.xml 文件，增加如下属性

```
[root@hadoop102 ~]# vim $HIVE_HOME/conf/hive-site.xml  
  
<!-- HiveServer2 启用 Kerberos 认证 -->  
<property>  
  <name>hive.server2.authentication</name>  
  <value>kerberos</value>  
</property>  
  
<!-- HiveServer2 服务的 Kerberos 主体 -->  
<property>  
  <name>hive.server2.authentication.kerberos.principal</name>  
  <value>hive/hadoop102@EXAMPLE.COM</value>  
</property>  
  
<!-- HiveServer2 服务的 Kerberos 密钥文件 -->  
<property>  
  <name>hive.server2.authentication.kerberos.keytab</name>  
  <value>/etc/security/keytab/hive.service.keytab</value>  
</property>  
  
<!-- Metastore 启动认证 -->  
<property>  
  <name>hive.metastore.sasl.enabled</name>  
  <value>true</value>  
</property>  
<!-- Metastore Kerberos 密钥文件 -->  
<property>  
  <name>hive.metastore.kerberos.keytab.file</name>  
  <value>/etc/security/keytab/hive.service.keytab</value>  
</property>  
<!-- Metastore Kerberos 主体 -->  
<property>  
  <name>hive.metastore.kerberos.principal</name>  
  <value>hive/hadoop102@EXAMPLE.COM</value>  
</property>
```

2. 修改 \$HADOOP_HOME/etc/hadoop/core-site.xml 文件，具体修改如下

```
[root@hadoop102 ~]# vim $HADOOP_HOME/etc/hadoop/core-site.xml
```

1) 删除以下参数

```
<property>  
  <name>hadoop.http.staticuser.user</name>  
  <value>atguigu</value>
```

```
</property>

<property>
  <name>hadoop.proxyuser.atguigu.hosts</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.atguigu.groups</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.atguigu.users</name>
  <value>*</value>
</property>
```

2) 增加以下参数

```
<property>
  <name>hadoop.proxyuser.hive.hosts</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.hive.groups</name>
  <value>*</value>
</property>

<property>
  <name>hadoop.proxyuser.hive.users</name>
  <value>*</value>
</property>
```

3.分发配置 core-site.xml 文件

```
[root@hadoop102 ~]# xsync $HADOOP_HOME/etc/hadoop/core-site.xml
```

4.重启 Hadoop 集群

```
[root@hadoop102 ~]# stop-dfs.sh
[root@hadoop103 ~]# stop-yarn.sh

[root@hadoop102 ~]# start-dfs.sh
[root@hadoop103 ~]# start-yarn.sh
```

6.3 启动 hiveserver2

注：需使用 hive 用户启动

```
[root@hadoop102 ~]# sudo -i -u hive hiveserver2
```

第 7 章 Hive Kerberos 认证使用说明

以下说明均基于普通用户

7.1 beeline 客户端

1.认证，执行以下命令，并按照提示输入密码

```
[atguigu@hadoop102 ~]$ kinit atguigu
```

2.使用 beeline 客户端连接 hiveserver2

```
[atguigu@hadoop102 ~]$ beeline
```

使用如下 url 进行连接

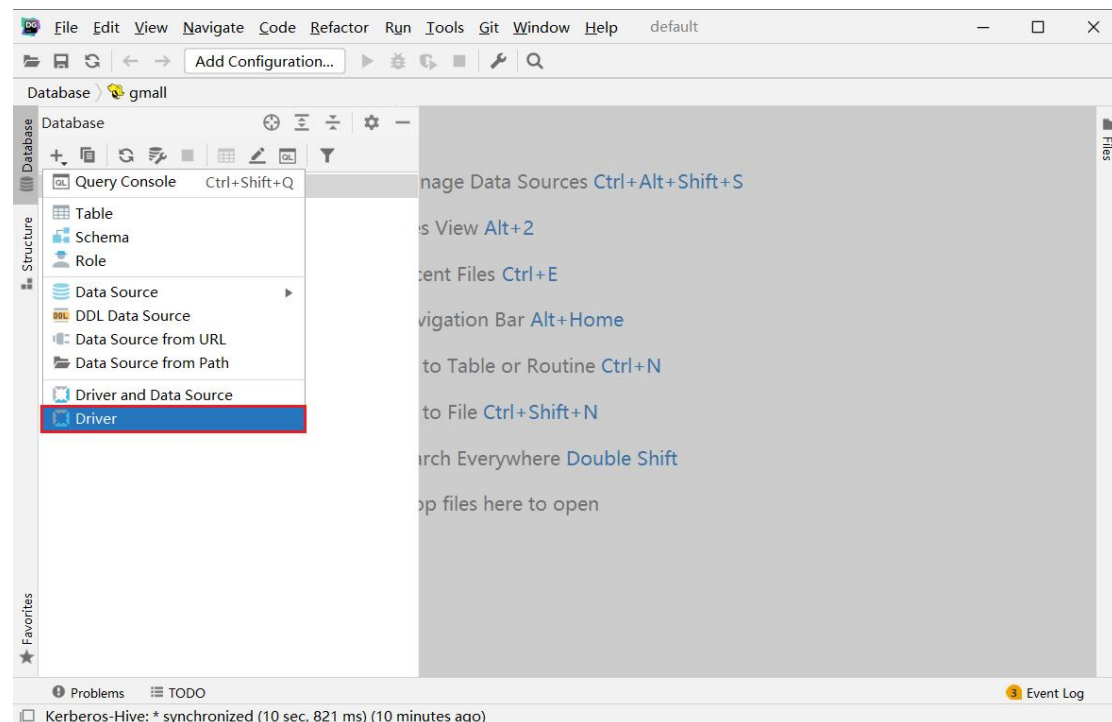
```
> !connect jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM
0: jdbc:hive2://hadoop102:10000/> !connect jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM
Connecting to jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
```

3.测试查询

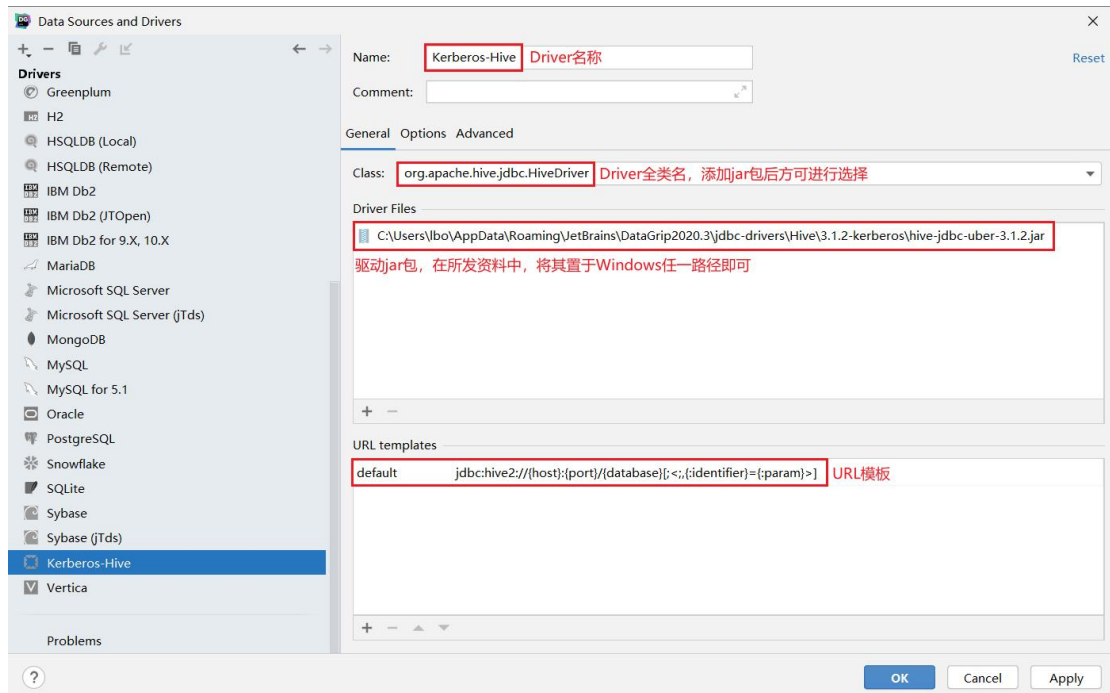
7.2 DataGrip 客户端

7.2.1 新建 Driver

1.创建 Driver



2.配置 Driver

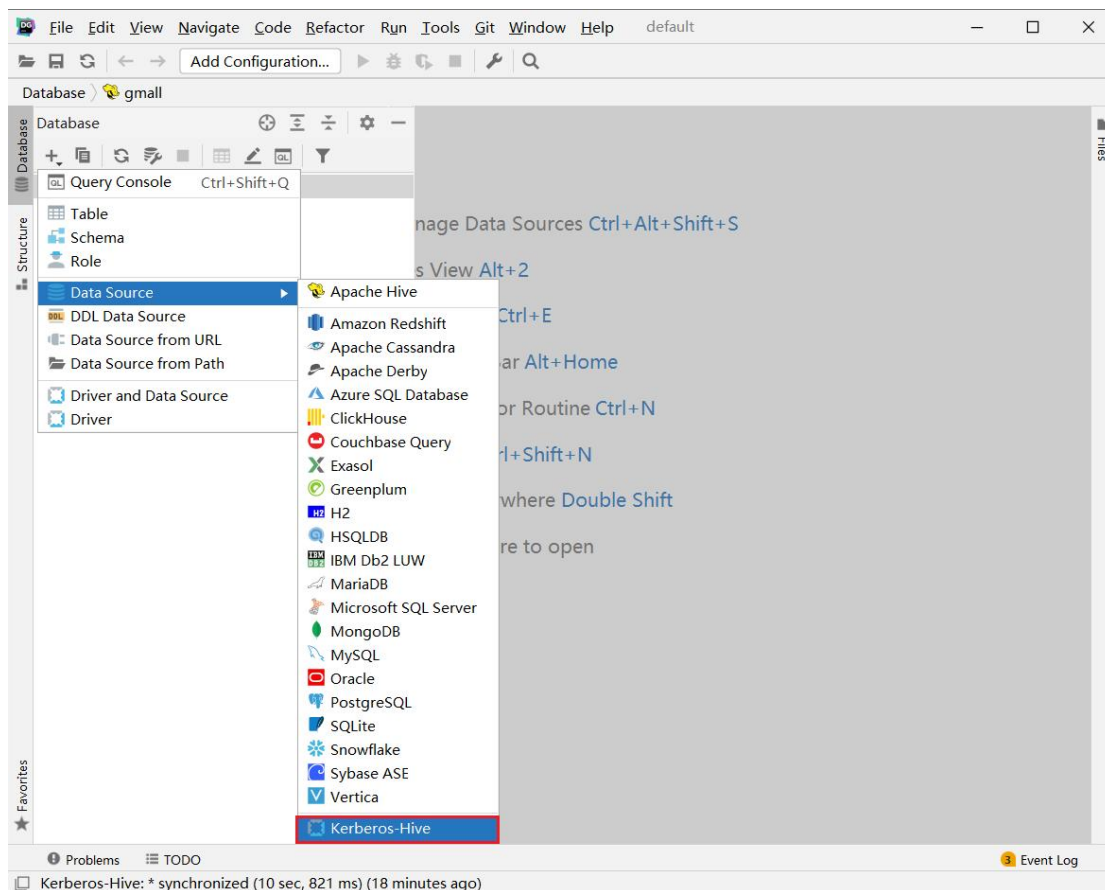


注：

url 模板： jdbc:hive2://{host}:{port}/{database}[,<,{:identifier}={:param}>]

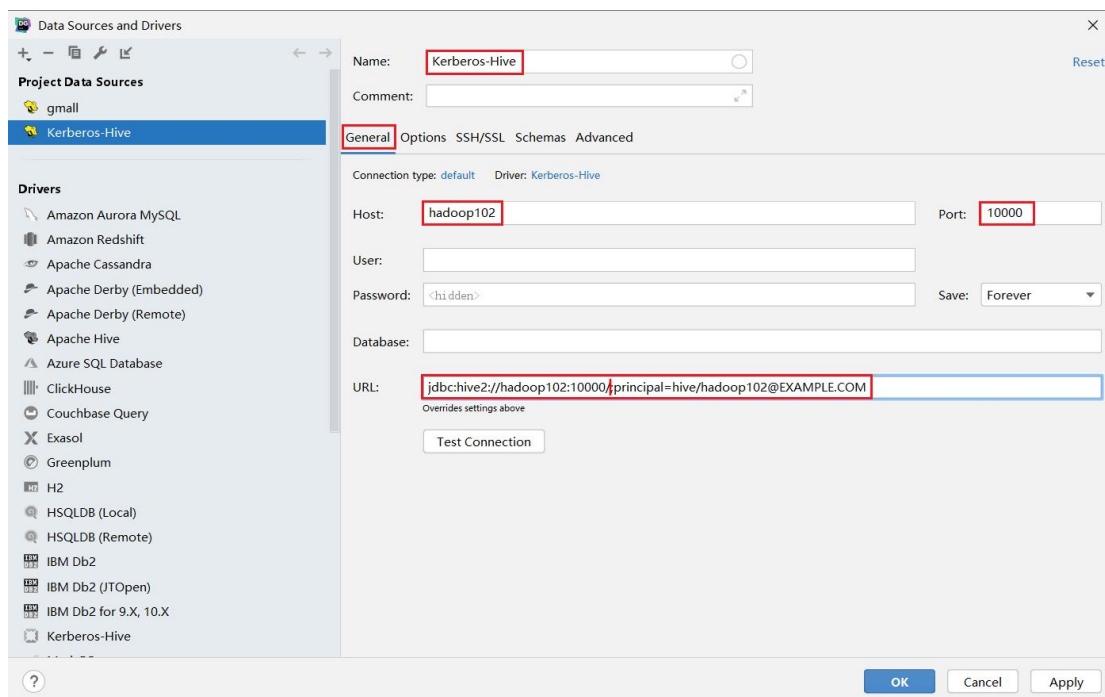
7.2.2 新建连接

1.创建连接



2.配置连接

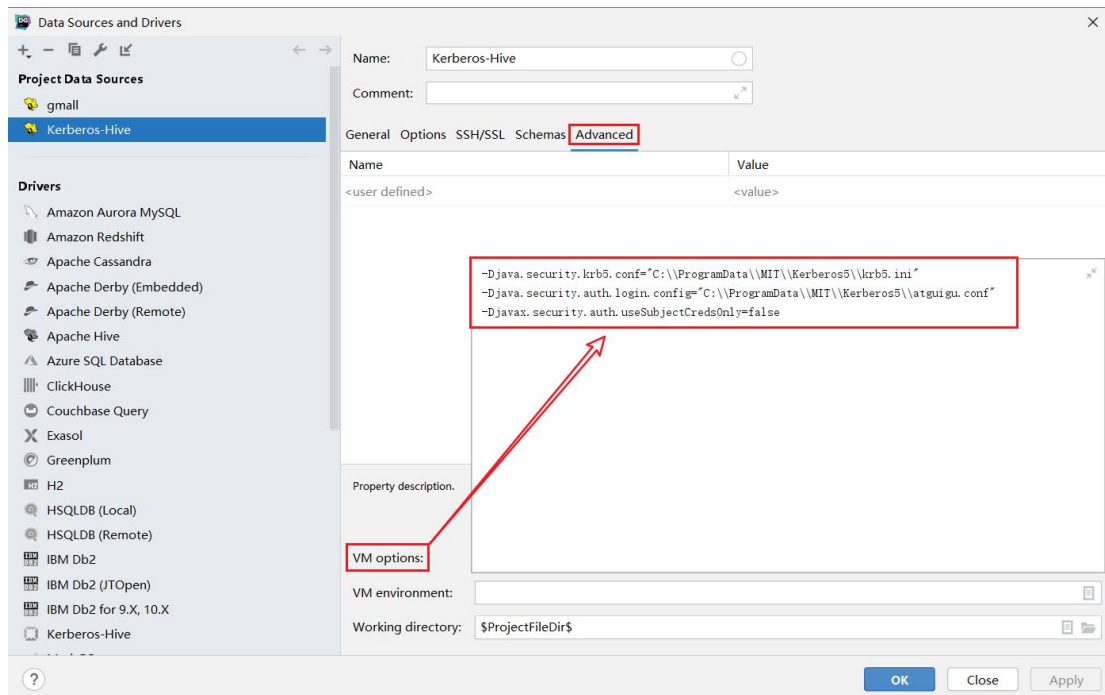
1) 基础配置



注:

url: jdbc:hive2://hadoop102:10000/;principal=hive/hadoop102@EXAMPLE.COM

2) 高级配置



注:

配置参数:

-Djava.security.krb5.conf="C:\\ProgramData\\MIT\\Kerberos5\\krb5.ini"

-Djava.security.auth.login.config="C:\\ProgramData\\MIT\\Kerberos5\\atguigu.conf"

-Djavax.security.auth.useSubjectCredsOnly=false

3) 编写 JAAS (Java 认证授权服务) 配置文件, 内容如下, 文件名和路径须和上图中 java.security.auth.login.config 参数的值保持一致。

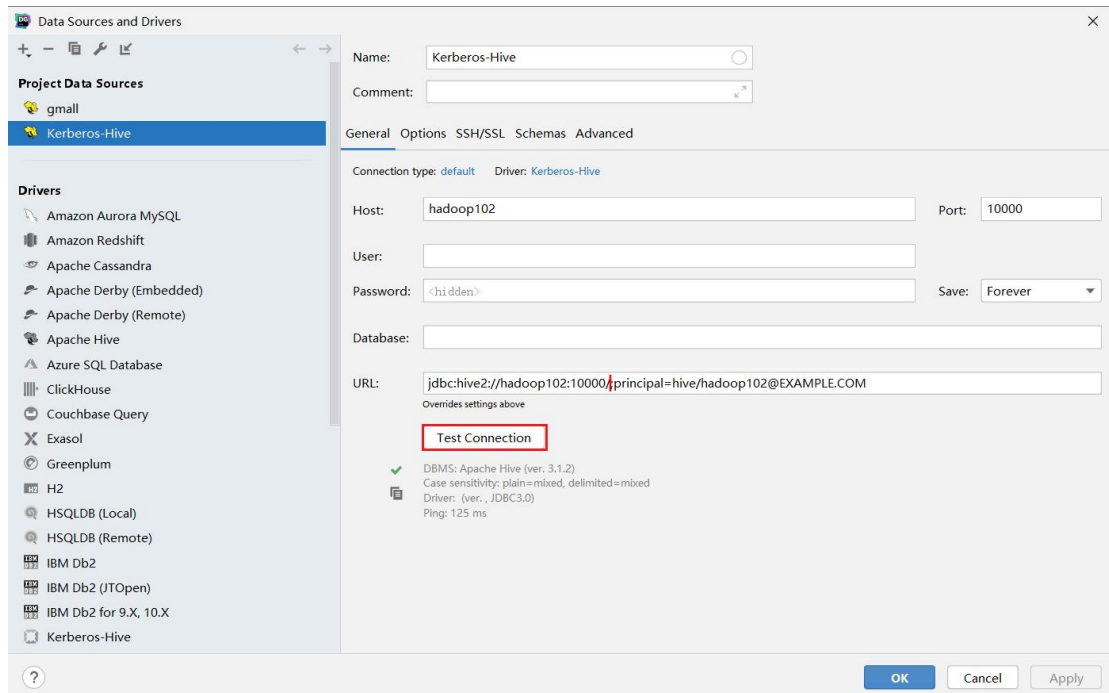
```
com.sun.security.jgss.initiate{
    com.sun.security.auth.module.Krb5LoginModule required
    useKeyTab=true
    useTicketCache=false
    keyTab="C:\\ProgramData\\MIT\\Kerberos5\\atguigu.keytab"
    principal="atguigu@EXAMPLE.COM";
};
```

4) 为用户生成 keytab 文件, 在 krb5kdc 所在节点 (hadoop102) 执行以下命令

```
[root@hadoop102]# kadmin.local -q"xst -norandkey -k /home/atguigu/atguigu.keytab atguigu"
```

5) 将上一步生成的 atguigu.keytab 文件, 置于 Windows 中的特定路径, 该路径须与 3) 中的 keyTab 属性的值保持一致。

6) 测试连接



The image shows a 'Data Sources and Drivers' dialog box with a sidebar on the left and a main configuration area on the right. The sidebar contains 'Project Data Sources' (with 'gmail' and 'Kerberos-Hive') and 'Drivers' (listing various database drivers like Amazon Aurora MySQL, Amazon Redshift, Apache Cassandra, etc., with 'Kerberos-Hive' selected). The main area has tabs for 'General', 'Options', 'SSH/SSL', 'Schemas', and 'Advanced'. The 'General' tab is active, showing fields for Name ('Kerberos-Hive'), Comment, Connection type ('default'), Driver ('Kerberos-Hive'), Host ('hadoop102'), Port ('10000'), User, Password ('<hidden>'), Save ('Forever'), Database, and URL ('jdbc:hive2://hadoop102:10000/?principal=hive/hadoop102@EXAMPLE.COM'). A 'Test Connection' button is highlighted with a red box. Below it, a green checkmark indicates a successful connection test with details: 'DBMS: Apache Hive (ver. 3.1.2)', 'Case sensitivity: plain=mixed, delimited=mixed', 'Driver: (ver., JDBC3.0)', and 'Ping: 125 ms'. At the bottom right are 'OK', 'Cancel', and 'Apply' buttons.

Data Sources and Drivers

Project Data Sources

- gmail
- Kerberos-Hive**

Drivers

- Amazon Aurora MySQL
- Amazon Redshift
- Apache Cassandra
- Apache Derby (Embedded)
- Apache Derby (Remote)
- Apache Hive
- Azure SQL Database
- ClickHouse
- Couchbase Query
- Exasol
- Greenplum
- H2
- HSQldb (Local)
- HSQldb (Remote)
- IBM Db2
- IBM Db2 (JOpen)
- IBM Db2 for 9.X, 10.X
- Kerberos-Hive**

General Options SSH/SSL Schemas Advanced

Connection type: default Driver: Kerberos-Hive

Host: hadoop102 Port: 10000

User:

Password: <hidden> Save: Forever

Database:

URL: jdbc:hive2://hadoop102:10000/?principal=hive/hadoop102@EXAMPLE.COM

Overrides settings above

Test Connection

✓ DBMS: Apache Hive (ver. 3.1.2)
Case sensitivity: plain=mixed, delimited=mixed
Driver: (ver., JDBC3.0)
Ping: 125 ms

OK Cancel Apply