三台机器：

192.168.29.190 安装kerberos、kafka、zookeeper服务

192.168.29.198 安装kafka、zookeeper服务

* + - 1. 安装kafka、zookeeper服务

1. Kerberos安装：

在192.168.29.190上安装kerberos服务

yum install krb5-server krb5-libs krb5-auth-dialog

在192.168.29.198、199上安装kerberos客户端

yum install krb5-devel krb5-workstation -y

2、安装完kerberos之后，会有两个配置文件  
/etc/krb5.conf 配置realm name  
/var/kerberos/krb5kdc/kdc.conf 配置domain-to-realm mappings，主机到域的映射。

配置/var/kerberos/krb5kdc/kdc.conf

说明：XUNCE.COM可随意取，代表一个realm，可配置多个realm，此处不讨论。此处的XUNCE.COM必须和/etc/krb5.conf中配置的保持一致。

[kdcdefaults]

kdc\_ports = 88

kdc\_tcp\_ports = 88

[realms]

XUNCE.COM = {

#master\_key\_type = aes256-cts

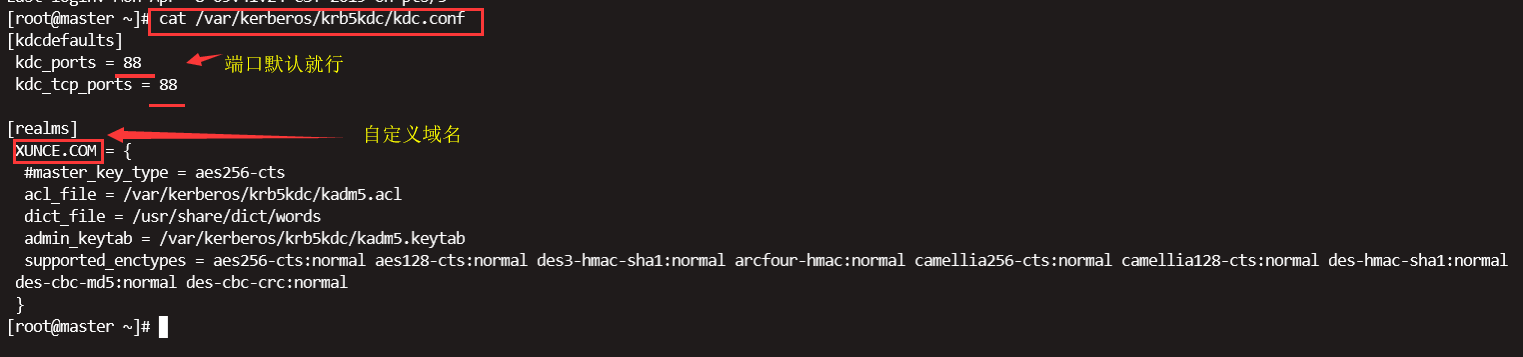
acl\_file = /var/kerberos/krb5kdc/kadm5.acl

dict\_file = /usr/share/dict/words

admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab

supported\_enctypes = 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

}



配置/etc/krb5.conf，并将文件copy到客户端(slave01、slave02)的对应位置。

[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

ticket\_lifetime = 1000d

renew\_lifetime = 1000d

forwardable = true

rdns = false

pkinit\_anchors = /etc/pki/tls/certs/ca-bundle.crt

default\_realm = XUNCE.COM

default\_ccache\_name = KEYRING:persistent:%{uid}

[realms]

XUNCE.COM = {

kdc = master

admin\_server = master

}

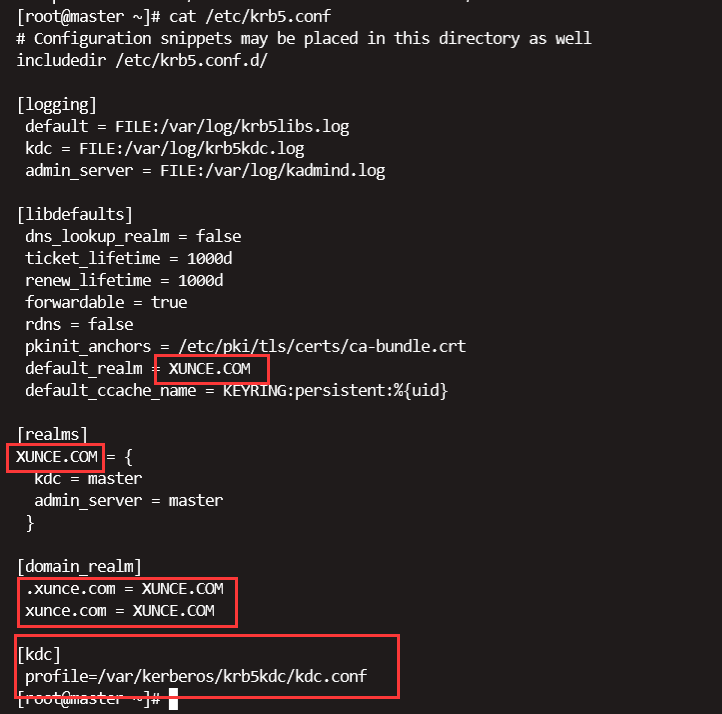
[domain\_realm]

.xunce.com = XUNCE.COM

xunce.com = XUNCE.COM

[kdc]

profile=/var/kerberos/krb5kdc/kdc.conf



参数说明：

[logging]：表示server端的日志的打印位置

[libdefaults]：每种连接的默认配置，需要注意以下几个关键的小配置

default\_realm = XUNCE.COM 默认的realm，必须跟要配置的realm的名称一致。

udp\_preference\_limit = 1 禁止使用udp

oticket\_lifetime表明凭证生效的时限，一般为24小时。

orenew\_lifetime表明凭证最长可以被延期的时限，一般为一个礼拜。当凭证过期之后，

对安全认证的服务的后续访问则会失败。

kdc：代表要kdc的位置。格式是 机器:端口，这里是服务器hosts，即master

admin\_server:代表admin的位置。格式是机器:端口

default\_domain：代表默认的域名

4.初始化kerberos并启动

kdb5\_util create -s -r XUNCE.COM

其中，[-s]表示生成stash file，并在其中存储master server key（krb5kdc）；还可以用[-r]来指定一个realm name —— 当krb5.conf中定义了多个realm时才是必要的。

创建过程中要求输入database的管理密码，此密码需要牢记。

当kerberos database创建好之后，可以在/var/kerberos/krb5kdc目录下看到

kadm5.acl

kdc.conf

principal

principal.kadm5

principal.kadm5.lock

principal.ok

6个文件

启动kerberos服务：

systemctl start krb5kdc.service  
systemctl start kadmin.service  
设置开机自启动  
systemctl enable krb5kdc.service  
systemctl enable kadmin.service

5.添加kerberos用户并生成密钥表文件：

kadmin.local -q "addprinc -randkey zookeeper/master@XUNCE.COM"

kadmin.local -q "addprinc -randkey kafka/master@XUNCE.COM"

kadmin.local -q "addprinc -randkey xunce/master@XUNCE.COM"

kadmin.local -q "addprinc -randkey zookeeper/slave01@XUNCE.COM"

kadmin.local -q "addprinc -randkey kafka/slave01@XUNCE.COM"

kadmin.local -q "addprinc -randkey xunce/slave01@XUNCE.COM"

kadmin.local -q "addprinc -randkey zookeeper/slave02@XUNCE.COM"

kadmin.local -q "addprinc -randkey kafka/slave02@XUNCE.COM"

kadmin.local -q "addprinc -randkey xunce/slave02@XUNCE.COM"

生成keytab密钥表文件：

kadmin.local -q "xst -norandkey -k xunce.keytab xunce/master@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab xunce/slave01@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab xunce/slave02@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab kafka/master@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab kafka/slave01@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab kafka/slave02@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab zookeeper/master@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab zookeeper/slave01@XUNCE.COM"

kadmin.local -q "xst -norandkey -k xunce.keytab zookeeper/slave02@XUNCE.COM"

将密钥copy到从节点(salve01、slave02)相应位置，这里是/usr/local/kafka/

scp /usr/local/kafka/xunce.keytab slave01: /usr/local/kafka/xunce.keytab

scp /usr/local/kafka/xunce.keytab slave02: /usr/local/kafka/xunce.keytab

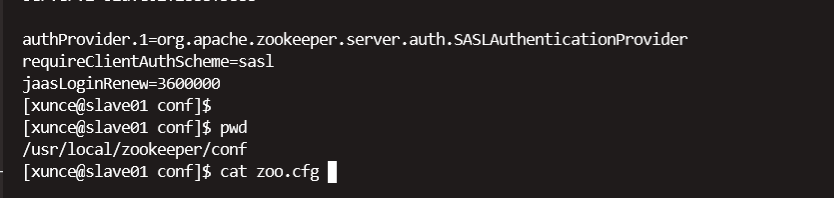
配置zookeeper的kerberos验证:

zoo.cfg新增（master、slave01、slave02）：

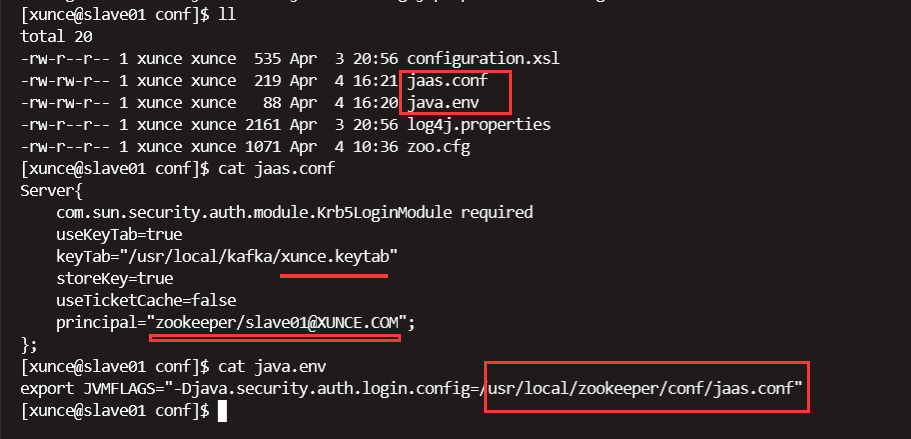
authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider

requireClientAuthScheme=sasl

jaasLoginRenew=3600000



新建文件jaas.conf、java.env，配置principal



将jaas.conf、java.env拷贝到slave01、slave02节点：

scp /usr/local/zookeeper/conf/jaas.conf slave01:/usr/local/zookeeper/conf/

scp /usr/local/zookeeper/conf/jaas.conf slave02:/usr/local/zookeeper/conf/

scp /usr/local/zookeeper/conf/java.env slave01:/usr/local/zookeeper/conf/

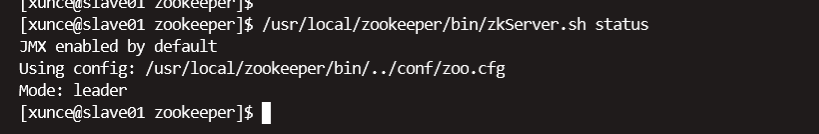
scp /usr/local/zookeeper/conf/java.env slave02:/usr/local/zookeeper/conf/

重启zookeeper服务:

/usr/local/zookeeper/bin/zkServer.sh stop

/usr/local/zookeeper/bin/zkServer.sh start

/usr/local/zookeeper/bin/zkServer.sh status



配置kafka的kerberos认证:

Kafka配置文件server.properties新增(master、slave01、slave02):

advertised.listeners=SASL\_PLAINTEXT://slave02:9092

listeners=SASL\_PLAINTEXT://slave02:9092

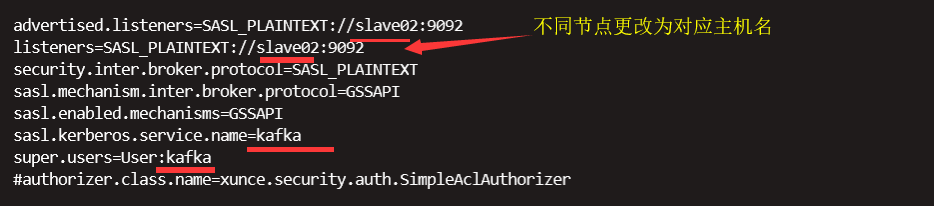
security.inter.broker.protocol=SASL\_PLAINTEXT

sasl.mechanism.inter.broker.protocol=GSSAPI

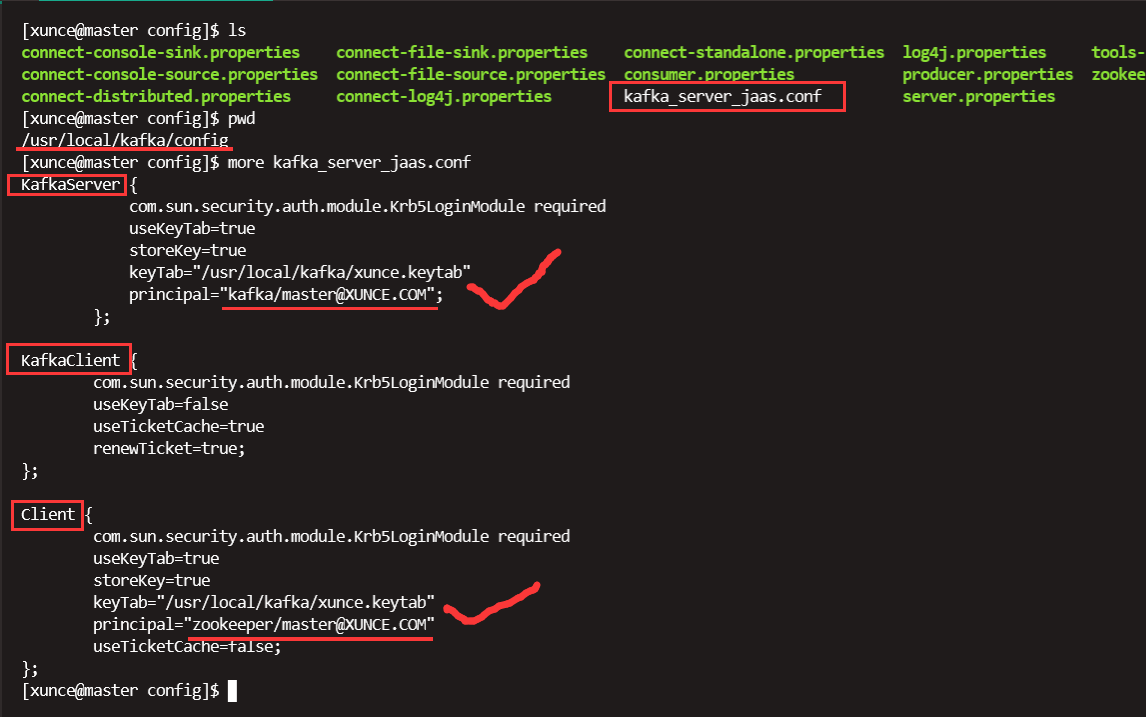
sasl.enabled.mechanisms=GSSAPI

sasl.kerberos.service.name=kafka

super.users=User:kafka

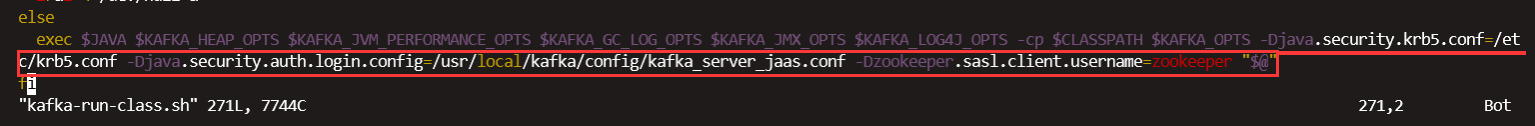


在/usr/local/kafka/config下新建文件kafka\_server\_jaas.conf，配置如下：



在/usr/local/kafka/bin下kafka-run-class.sh，exec $JAVA行，新增配置如下：

-Djava.security.krb5.conf=/etc/krb5.conf -Djava.security.auth.login.config=/usr/local/kafka/config/kafka\_server\_jaas.conf -Dzookeeper.sasl.client.username=zookeeper



将kafka\_server\_jaas.conf、kafka-run-class.sh拷贝到slave01、slave02节点：

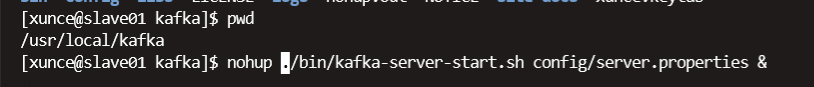
scp /usr/local/kafka/conf/kafka\_server\_jaas.confslave01:/usr/local/kafka/conf/

scp /usr/local/kafka/conf/kafka\_server\_jaas.conf slave02:/usr/local/kafka/conf/

scp /usr/local/kafka/bin/ka-run-class.sh slave01:/usr/local/kafka/bin/

scp /usr/local/kafka/bin/ka-run-class.sh slave02:/usr/local/kafka/bin/

重启kafka集群：master、slave01、slave02



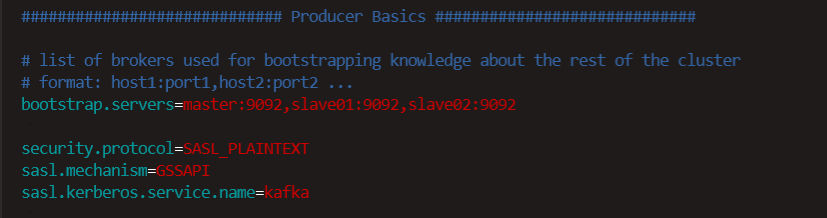
修改生产者配置：

bootstrap.servers=master:9092,slave01:9092,slave02:9092

security.protocol=SASL\_PLAINTEXT

sasl.mechanism=GSSAPI

sasl.kerberos.service.name=kafka



修改消费者配置：

bootstrap.servers=master:9092,slave01:9092,slave02:9092

zookeeper.connect=master:2181,slave01:2181,slave02:2181

#kerberos

security.protocol=SASL\_PLAINTEXT

sasl.mechanism=GSSAPI

sasl.kerberos.service.name=kafka

