# Kafka Installation

ReD Merchant Fraud

Exported on 06/05/2020

# Table of Contents

Last Updated: 20th Dec 2018

***All config details on document are added here from Limerick UAT servers. Same settings applied on Norcross UAT**

***systemctl script updated Mutiple times**

**6 servers in each DC (for both UAT and PROD)**

lmu22conkafv001 -- for confluent control center/c3

lmp22repkafv001 -- connect/replicator server

lmp22repkafv002 -- connect/replicator server

lmp22dbskaf001 -- Broker/zk server

lmp22dbskaf002 -- Broker/zk server

lmp22dbskaf003 -- Broker/zk server

lmu22scrkafv001 -- Schema Regisgtry1

lmu22scrkafv002 -- Schema Registry2

**1.Pre- checks:**

Make sure same RHEL version across cluster cat /etc/redhat-release

service ntpd status –active status

java -version (same version across nodes)

SELINUX = Permissive (cat /etc/selinux/config)

service firewalld status - inactive

service iptbales status – disable

Enable Password less SSH for root with in cluster

Make sure repos enabled

Check /etc/security/limits.d if kafka.conf profile is there

** script copied in to /root

-rwxr-xr-x. 1 root root 1122 Jun 5 06:24 **create_kafka_users_and_groups.sh** -- Make sure to run this script before installing Conflunet kafka

*This script creates required users and groups with unique UID/GID

Ex: cp-ksql:x:**10051**:10101:Confluent KSQL:/tmp:/sbin/nologin[1]

cp-schema-registry:x:**10052**:10101:Confluent Schema Registry:/tmp:/sbin/nologin[2]

cp-kafka-rest:x:**10053**:10101:Confluent Kafka REST proxy:/tmp:/sbin/nologin[3]

1 http://KSQL/tmp:/sbin/nologin
2 http://Registry/tmp:/sbin/nologin
3 http://proxy/tmp:/sbin/nologin

cp-kafka:x:**10054**:10101:Confluent Kafka:/var/lib/kafka:/sbin/nologin[4]

cp-kafka-connect:x:**10055**:10101:Confluent Kafka Connect:/tmp:/sbin/nologin[5]

cp-control-center:x:**10056**:10101:Confluent Control Center:/var/lib/confluent/control-center:/sbin/nologin[6]

**App version** : Confluent 4.1.1

1. **Installation:**

--To install confluent enterprise platform on brokers

yum install confluent-platform-2.11

Installed:

confluent-platform-2.11.noarch 0:4.1.1-1

Dependency Installed:

confluent-camus.noarch 0:4.1.1-1 confluent-cli.noarch 0:4.1.1-1 confluent-common.noarch 0:4.1.1-1

confluent-control-center.noarch 0:4.1.1-1 confluent-control-center-fe.noarch 0:4.1.1-1 confluent-kafka-2.11.noarch 0:1.1.1-1

confluent-kafka-connect-elasticsearch.noarch 0:4.1.1-1 confluent-kafka-connect-hdfs.noarch 0:4.1.1-1 confluent-kafka-connect-jdbc.noarch 0:4.1.1-1

confluent-kafka-connect-jms.noarch 0:4.1.1-1 confluent-kafka-connect-replicator.noarch 0:4.1.1-1 confluent-kafka-connect-s3.noarch 0:4.1.1-1

confluent-kafka-connect-storage-common.noarch 0:4.1.1-1 confluent-kafka-rest.noarch 0:4.1.1-1 confluent-ksql.noarch 0:4.1.1-1

confluent-rebalancer.noarch 0:4.1.1-1 confluent-rest-utils.noarch 0:4.1.1-1 confluent-schema-registry.noarch 0:4.1.1-1

confluent-support-metrics.noarch 0:4.1.1-1

Complete!

check directories are created

/etc/kafka

/etc/kafka-rest etc

**Repeat below steps on all brokers and connect workers

4 http://Kafka/var/lib/kafka:/sbin/nologin
5 http://Connect/tmp:/sbin/nologin
6 http://Center/var/lib/confluent/control-center:/sbin/nologin

1. **From /etc/kafka/server. Properties** amend below values accordingly on each broker

broker.id[7]: ex: 001 (unique for each broker)

Added broker.rack (ex: broker.rack=a31)

listeners=SASL_SSL://lmu22dbskaf001.ise.pos.net:9093(see page 3)

#listeners=PLAINTEXT://lmu22dbskaf001.ise.pos.net:9092,SSL://lmu22dbskaf001.ise.pos.net:9093

#advertised.listeners=PLAINTEXT://lmu22dbskaf001.ise.pos.net:9092,SSL://lmu22dbskaf001.ise.pos.net:9093

advertised.listeners=SASL_SSL://lmu22dbskaf001.ise.pos.net:9093(see page 3)

#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL

listener.security.protocol.map=SASL_SSL:SASL_SSL

log.dirs=/kafka/logs

offsets.topic.replication.factor=3

change below values

zookeeper.connect=lmu22dbskaf001.ise.pos.net[8]:2181,lmu22dbskaf002.ise.pos.net[9]:2181,lmu22dbskaf003.ise.pos.net[10]:2181

zookeeper session timeout: changed to zookeeper.connection.timeout.ms[11]=30000 (default is 6000)

Un-comment/add below lines

metric.reporters=io.confluent.metrics.reporter.ConfluentMetricsReporter

confluent.metrics.reporter.request.timeout.ms[12]=60000 (This was added as part of performance issue on 20th Dec, recommended by confluent kafka)

confluent.metrics.reporter.bootstrap.servers=lmu22dbskaf001.ise.pos.net[13]:9093,lmu22dbskaf002.ise.pos.net[14]:9093,lmu22dbskaf003.ise.pos.net[15]:9093

confluent.metrics.reporter.zookeeper.connect=lmu22dbskaf001.ise.pos.net[16]:2181,lmu22dbskaf002.ise.pos.net[17]:2181,lmu22dbskaf003.ise.pos.net[18]:2181

---

7 http://broker.id
8 http://lmu22dbskaf001.ise.pos.net
9 http://lmu22dbskaf002.ise.pos.net
10 http://lmu22dbskaf003.ise.pos.net
11 http://zookeeper.connection.timeout.ms
12 http://confluent.metrics.reporter.request.timeout.ms
13 http://lmu22dbskaf001.ise.pos.net
14 http://lmu22dbskaf002.ise.pos.net
15 http://lmu22dbskaf003.ise.pos.net
16 http://lmu22dbskaf001.ise.pos.net
17 http://lmu22dbskaf002.ise.pos.net
18 http://lmu22dbskaf003.ise.pos.net

confluent.metrics.reporter.security.protocol=SASL_SSL

confluent.metrics.reporter.ssl.truststore.location=/etc/security/certs/truststore.jks

confluent.metrics.reporter.ssl.truststore.password=xxxx

confluent.metrics.reporter.ssl.keystore.location=/etc/security/certs/keystore.jks

confluent.metrics.reporter.ssl.keystore.password=xxxx

confluent.metrics.reporter.sasl.mechanism=GSSAPI

confluent.metrics.reporter.sasl.kerberos.service.name[19]=kafka

confluent.metrics.reporter.sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required \

useKeyTab=true \

storeKey=true \

keyTab="/etc/security/keytabs/kafka_client.keytab" \

principal="kafkaclient@AOD.LOCAL[20]";

confluent.support.metrics.enable=false

Add below parameters

############################ SASL_SSL Properties ###############################

ssl.keystore.location=/etc/security/certs/keystore.jks

ssl.keystore.password=xxxx

ssl.keystore.type=jks

ssl.key.password=xxxxx

ssl.truststore.location=/etc/security/certs/truststore.jks

ssl.truststore.type=jks

ssl.truststore.password=xxxxx

security.inter.broker.protocol=SASL_SSL

ssl.client.auth=required

ssl.enabled.protocols=TLSv1.2

# List of enabled mechanisms, can be more than one

sasl.enabled.mechanisms=GSSAPI

# Specify one of of the SASL mechanisms

sasl.mechanism.inter.broker.protocol=GSSAPI

sasl.kerberos.service.name[21]=kafka

19 http://confluent.metrics.reporter.sasl.kerberos.service.name
20 mailto:principal=%22kafkaclient@AOD.LOCAL
21 http://sasl.kerberos.service.name

-

-

1. **Configuring Brokers: -- steps for all brokers**

--From /etc/kafka/connect-distributed.properties

Add below to bootstratp servers:

bootstrap.servers=lmu22dbskaf001.ise.pos.net[22]:9092,lmu22dbskaf002.ise.pos.net[23]:9092,lmu22dbskaf003.ise.pos.net[24]:9092

Enable stream Monitoring in C3 add below to /etc/kafka/connect-distributed.properties

producer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringProducerInterceptor

consumer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringConsumerInterceptor

1. **Start Services:**

Start zookeeper on zookeeper brokers and kafka on all brokers

To start Zookeeper:

sudo systemctl start confluent-zookeeper

Check status : systemctl status confluent-zookeeper

systemctl status confluent-zookeeper

â confluent-zookeeper.service - Apache Kafka - ZooKeeper

Loaded: loaded (/usr/lib/systemd/system/confluent-zookeeper.service; enabled; vendor preset: disabled)

To start Kafka:

sudo systemctl start confluent-kafka

Check status : systemctl status confluent-kafka

systemctl status confluent-kafka

â confluent-kafka.service - Apache Kafka - broker

Loaded: loaded (/usr/lib/systemd/system/confluent-kafka.service; enabled; vendor preset: disabled)

1. **To enable auto –start during boot**

sudo systemctl enable confluent-zookeeper --- enabled

sudo systemctl enable confluent-kafka -- enabled

22 http://lmu22dbskaf001.ise.pos.net
23 http://lmu22dbskaf002.ise.pos.net
24 http://lmu22dbskaf003.ise.pos.net

Other things we can check:

jps | grep -i Quorum

jps | grep -i SupportedKafka

zookeeper-shell hostname:2181 ls /brokers/ids

1. **Error:**

ERROR [KafkaServer id=3] Fatal error during KafkaServer startup. Prepare to shutdown (kafka.server.KafkaServer)

kafka.common.KafkaException: Found directory /var/lib/kafka/.oracle_jre_usage, '.oracle_jre_usage' is not in the form of topic-partition or topic-partition.uniqueId-delete (if marked for deletion).

Work around:

Created a file usagetracker.properties

with com.oracle.usagetracker.track.last.usage = false and copied in to

/usr/java/jdk1.8.0_131/jre/lib/management on all brokers

1. **Zookeeper config:**

From zookeeper.properties

dataDir=/zookeeper

Under /zookeeper create folder version-2and own by cp-kafka user and confluent group

chown cp-kafka:confluent version-2/ (permissions taken from previous path /var/lib/zookeeper)

Also create a file with myid with integer value and unique to each zookeper node (001, 002, 003) and this next to server.xxx on below

Add below values for all zookeeper nodes (these values can be amended but make sure same values across the zookeeper nodes)

tickTime=2000

initLimit=15 (15 x 2000 = 30000ms.)

syncLimit=2

server.001=lmu22dbskaf001.ise.pos.net[25]:2888:3888

server.002=lmu22dbskaf002.ise.pos.net[26]:2888:3888

server.003=lmu22dbskaf003.ise.pos.net[27]:2888:3888

autopurge.snapRetainCount=3

autopurge.purgeInterval=24

25 http://lmu22dbskaf001.ise.pos.net
26 http://lmu22dbskaf002.ise.pos.net
27 http://lmu22dbskaf003.ise.pos.net

zookeeper.set.acl=true (for kerberos)

1. **Redirect kafka data(logs)**

Stop kafka service on all brokers

From server.properties

Change this parameter: log.dirs=/kafka/logs (previous path :/var/lib/kafka)

Under /kafka

chown cp-kafka:confluent logs/ (permissions taken from previous path /var/lib/kafka)

Start kafka service on all brokers

We can see logs in /kafka/logs

-

1. **C3 installation:**

dedicated server for confluent control center lmu22conkafv001

yum install confluent-control-center

Installed:

confluent-control-center.noarch 0:4.1.1-1

Dependency Installed:

confluent-common.noarch 0:4.1.1-1 confluent-control-center-fe.noarch 0:4.1.1-1 confluent-rebalancer.noarch 0:4.1.1-1 confluent-rest-utils.noarch 0:4.1.1-1

Complete!

To Enable **auto –start during boot**

sudo systemctl enable confluent-control-center --- enabled

systemctl start confluent-control-center

**10.1 Configuration changes on control-center-production.properties:**

bootstrap.servers=lmu22dbskaf001.ise.pos.net[28]:9093,lmu22dbskaf002.ise.pos.net[29]:9093,lmu22dbskaf003.ise.pos.net[30]:9093

28 http://lmu22dbskaf001.ise.pos.net
29 http://lmu22dbskaf002.ise.pos.net
30 http://lmu22dbskaf003.ise.pos.net

zookeeper.connect=lmu22dbskaf001.ise.pos.net[31]:2181,lmu22dbskaf002.ise.pos.net[32]:2181,lmu22dbskaf003.ise.pos.net[33]:2181

confluent.controlcenter.id[34]=**UATLIM**

confluent.controlcenter.data.dir=/opt/lib/confluent/control-center --- more space required – can be redirected to required path

Licnese key

xxxxxxxxxxxxxxxxx

confluent.controlcenter.connect.cluster=lmu22conkafv001.ise.pos.net[35]:8083,lmu22conkafv002.ise.pos.net[36]:8083

confluent.controlcenter.internal.topics.replication=3

confluent.controlcenter.internal.topics.partitions=2

confluent.controlcenter.command.topic.replication=3

confluent.monitoring.interceptor.topic.partitions=2

confluent.monitoring.interceptor.topic.replication=3

confluent.metrics.topic.partitions=2

confluent.metrics.topic.replication=3

Add below

##############################SASL_SSL Properties##############################

confluent.controlcenter.streams.security.protocol=SASL_SSL

confluent.controlcenter.streams.ssl.truststore.location=/etc/security/certs/truststore.jks

confluent.controlcenter.streams.ssl.truststore.password=xxxxx

confluent.controlcenter.streams.ssl.keystore.location=/etc/security/certs/keystore.jks

confluent.controlcenter.streams.ssl.keystore.password=xxxxx

confluent.controlcenter.streams.sasl.mechanism=GSSAPI

confluent.controlcenter.streams.sasl.kerberos.service.name[37]=kafka

confluent.controlcenter.streams.sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required \

useKeyTab=true \

storeKey=true \

keyTab="/etc/security/keytabs/kafka_client.keytab" \

principal="kafkacontrolcenter@AOD.LOCAL[38]";

######################## Sync C3 with Open LDAP ------ WIP

---

31 http://lmu22dbskaf001.ise.pos.net
32 http://lmu22dbskaf002.ise.pos.net
33 http://lmu22dbskaf003.ise.pos.net
34 http://confluent.controlcenter.id
35 http://lmu22conkafv001.ise.pos.net
36 http://lmu22conkafv002.ise.pos.net
37 http://confluent.controlcenter.streams.sasl.kerberos.service.name
38 mailto:principal=%22kafkacontrolcenter@AOD.LOCAL

########################## Additional properties

confluent.support.metrics.enable=false

########################## HTTPS UI ##########################

confluent.controlcenter.rest.listeners=https://0.0.0.0:9021

confluent.controlcenter.rest.ssl.keystore.location=/etc/security/certs/keystore.jks

confluent.controlcenter.rest.ssl.keystore.password=xxxxxx

confluent.controlcenter.rest.ssl.key.password=xxxxx

confluent.controlcenter.rest.ssl.truststore.location=/etc/security/certs/truststore.jks

confluent.controlcenter.rest.ssl.truststore.password=xxxxx

confluent.controlcenter.rest.ssl.keystore.type=JKS

confluent.controlcenter.rest.ssl.truststore.type=JKS

confluent.controlcenter.rest.ssl.protocol=TLSv1.2

**10.2 control-center_conf.jass --- WIP**

1. **kafka-connect-replicator:**

dedicated serves for kafka-connect : lmu22repkafv001 and lmu22repkafv002

yum install confluent-kafka-connect-replicator.noarch ----- * replicator app not working as expected we gone ahead with full package installation so we can use connect service and replicator

yum install confluent-platform-2.11

And install jq on connect servers

yum install jq

Installed:

jq.x86_64 0:1.5-1.el7

Dependency Installed:

oniguruma.x86_64 0:5.9.5-3.el7

Complete!

1. **Create keystore and truststore** and copy across servers @ /etc/security/certs

**Keystore** contains specific to server with rootca:

**Commands**:

keytool -importkeystore -srckeystore lmu22dbskaf001[39].pfx -srcstoretype pkcs12 -destkeystore keystore.jks -deststoretype JKS -storepass xxxx

39 mailto:zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL

keytool -import -alias rootca -file /home/polisettym/Hadoop_Prod_Root_CA.cer -keystore keystore.jks -storepass xxxx

To change name if Alias if required

keytool -changealias -keystore /home/polisettym/lmu22dbskaf001[40]/keystore.jks -alias 1 -destalias lmu22dbskaf001[41].ise.pos.net[42]

**Truststore** contains SSL certs for both Limerick and Norcross servers, ldaprootca and rootca:

**Commnads**:

keytool -import -noprompt -alias lmp22hdpwrk010.ise.pos.net[43] -file /home/xxxxx/xxxxxx/lmu22dbskaf001[44].crt -keystore truststore.jks -storepass xx

Similarly add all server certs to truststore

keytool -import -noprompt -alias **RootCA** -file /home/polisettym/Hadoop_Prod_Root_CA.cer -keystore truststore.jks -storepass xx

keytool -import -file /home/polisettym/**ldap.aod.local-ca.pem** -keystore truststore.jks -storepass xx

*Because we are using replication certs form both DC's nodes added to truststore

Added ldaprootca to java truststore on all servers

**Commnad:**

keytool -import -trustcacerts -keystore /usr/java/jdk1.8.0_131/jre/lib/security/cacerts -storepass changeit -noprompt -alias ldprootca -file /etc/openldap/cacerts/ldap.aod.local-ca.pem

**Keytabs:**

Also, for Kerberos create **keytabs** with one principle name on each kaytab (security team will generate this keytabs)

zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[45]

Zookeeper/lmu22dbskaf002.ise.pos.net@AOD.LOCAL[46]

zookeeper/lmu22dbskaf003.ise.pos.net@AOD.LOCAL[47]

kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[48]

kafka/lmu22dbskaf002.ise.pos.net@AOD.LOCAL[49]

kafka/lmu22dbskaf003.ise.pos.net@AOD.LOCAL[50]

Above keytabs will be copied to @ /etc/security/keytabs on relevant servers with name of zookeeper.keytab and kafka_server.keytab

---

40 mailto:zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
41 mailto:zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
42 http://ise.pos.net
43 http://lmp22hdpwrk010.ise.pos.net
44 mailto:zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
45 mailto:zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
46 mailto:Zookeeper/lmu22dbskaf002.ise.pos.net@AOD.LOCAL
47 mailto:zookeeper/lmu22dbskaf003.ise.pos.net@AOD.LOCAL
48 mailto:kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
49 mailto:kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
50 mailto:kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL

Replicator.keytab (replicator@AOD.LOCAL[51]) copied in to both connect/replicator nodes @ /etc/security/keytabs

Controlcenter.keytab (controlcenter@AOD.LOCAL[52] ) copied in to C3 node @ /etc/security/keytabs

And kafka_client.keytab -- this keytab created with 5 below principles and copies across the cluster (all nodes) @ /etc/security/keytabs

kafkaclient@AOD.LOCAL[53], kafkaconnect@AOD.LOCAL[54], kafkarestproxy@AOD.LOCAL[55], kafkareplicator@AOD.LOCAL[56], kafkacontrolcenter@AOD.LOCAL[57]

Permissions on keytabs

-rw-r--r--. 1 root root 706 Jun 13 15:25 kafka_server.keytab

-rw-r--r--. 1 root root 738 Jun 13 15:26 zookeeper.keytab

-rw-r--r--. 1 root root 2762 Jun 20 17:56 kafka_client.keytab

1. **zookeeper_jaas.conf -- on all zk nodes**

Create zookeeper_jaas.conf file @ /etc/kafka with below details

```
// Specifies a unique keytab and principal name for each ZooKeeper node
Server {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="/etc/security/keytabs/zookeeper.keytab"
  storeKey=true
  useTicketCache=false
  principal="zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[58]";
};
```

1. **kafka_server_jaas.conf -- on all broker nodes**

```
[root@lmu22dbskaf001 kafka]# cat kafka_server_jaas.conf
// Specifies a unique keytab and principal name for each broker
KafkaServer {
  com.sun.security.auth.module.Krb5LoginModule required
```

51 mailto:replicatorprod@AOD.LOCAL
52 mailto:controlcenterprod@AOD.LOCAL
53 mailto:kafkaclient@AOD.LOCAL
54 mailto:kafkaconnect@AOD.LOCAL
55 mailto:kafkarestproxy@AOD.LOCAL
56 mailto:kafkareplicator@AOD.LOCAL
57 mailto:kafkacontrolcenter@AOD.LOCAL
58 mailto:principal=%22zookeeper/lmu22dbskaf001.ise.pos.net@AOD.LOCAL

```
  useKeyTab=true
  storeKey=true
  keyTab="/etc/security/keytabs/kafka_server.keytab"
  principal="kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[59]";
};

// ZooKeeper client authentication
Client {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  storeKey=true
  keyTab="/etc/security/keytabs/kafka_server.keytab"
  principal="kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[60]";
};
```

1. **create producer_ssl.properties** @/etc/kafka on all brokers

```
[root@nxu22dbskaf001 kafka]# cat producer_ssl.properties
security.protocol=SASL_SSL
ssl.truststore.location=/etc/security/certs/truststore.jks
ssl.truststore.password=xxxxx
ssl.keystore.location=/etc/security/certs/keystore.jks
ssl.keystore.password=xxxxt
ssl.key.password=xxxx
sasl.mechanism=GSSAPI
sasl.kerberos.service.name[61]=kafka
sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required \
  useKeyTab=true \
  storeKey=true \
  keyTab="/etc/security/keytabs/kafka_client.keytab" \
  principal="kafkaclient@AOD.LOCAL[62]";
```

1. **Create consumer_ssl.properties** @ /etc/kafka on all brokers

59 mailto:principal=%22kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
60 mailto:principal=%22kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL
61 http://sasl.kerberos.service.name
62 mailto:principal=%22kafkaclient@AOD.LOCAL

[root@nxu22dbskaf001 kafka]# cat consumer_ssl.properties

security.protocol=SASL_SSL

ssl.truststore.location=/etc/security/certs/truststore.jks

ssl.truststore.password=xxxx

ssl.keystore.location=/etc/security/certs/keystore.jks

ssl.keystore.password=xxxxx

ssl.key.password=xxxxx

sasl.mechanism=GSSAPI

sasl.kerberos.service.name[63]=kafka

sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required \

useKeyTab=true \

storeKey=true \

keyTab="/etc/security/keytabs/kafka_client.keytab" \

principal="kafkaclient@AOD.LOCAL[64]";

1. **JCE files**

**Replace** local_policy.jar and US_export_policy.jar JCE files form

/usr/java/jdk1.8.0_131/jre/lib/security/

After replacing

-rw-r--r--. 1 root root 3023 Jun 12 13:36 US_export_policy.jar

-rw-r--r--. 1 root root 3035 Jun 12 13:37 local_policy.jar

1. **Update systemctl script from Systemd**

Path : /usr/lib/systemd/system/

For all Zookpeer nodes

Add Environment="KAFKA_OPTS=-Djava.security.auth.login.config=/etc/kafka/zookeeper_jaas.conf" to confluent-zookeeper.service file

For all Broker nodes

Add Environment="KAFKA_OPTS=-Djava.security.auth.login.config=/etc/kafka/kafka_server_jaas.conf" to confluent-kafka.service file

systemctl daemon-reload and restart services

**[root@lmu22dbskaf001 ~]#systemctl cat confluent-kafka**
**# /usr/lib/systemd/system/confluent-kafka.service**
**[Unit]**

63 http://sasl.kerberos.service.name
64 mailto:principal=%22kafkaclient@AOD.LOCAL

**Description=Apache Kafka - broker**
**Documentation=**http://docs.confluent.io/
[65]**After=network.target confluent-zookeeper.target**

[Service]
Type=simple
User=cp-kafka
Group=confluent
Environment="KAFKA_OPTS=-Djava.security.auth.login.config=/etc/kafka/kafka_server_jaas.conf"
ExecStart=/usr/bin/kafka-server-start /etc/kafka/server.properties
TimeoutStopSec=180
Restart=no

[Install]
WantedBy=multi-user.target
**[root@lmu22dbskaf001 ~]#systemctl cat confluent-zookeeper**
**# /usr/lib/systemd/system/confluent-zookeeper.service**
**[Unit]**
**Description=Apache Kafka - ZooKeeper**
**Documentation=**http://docs.confluent.io/
[66]**After=network.target**

[Service]
Type=simple
User=cp-kafka
Group=confluent
Environment="KAFKA_OPTS=-Djava.security.auth.login.config=/etc/kafka/zookeeper_jaas.conf"
ExecStart=/usr/bin/zookeeper-server-start /etc/kafka/zookeeper.properties
TimeoutStopSec=180
Restart=no

[Install]
WantedBy=multi-user.target

1. **Replicator/Connect Nodes configuration**

cp /etc/kafka-connect-replicator/replicator-connect-distributed.properties /etc/kafka/**connect-distributed.properties**

cp /etc/kafka-connect-replicator/**replicator.json** /etc/kafka/

**19.1 replicator.json** file details (update on both **connect/replicator** servers)

Rename format will be **-replica**

**19.2 connect-distributed.properties** file details (update on both **connect/replicator** servers)

65 http://docs.confluent.io/
66 http://docs.confluent.io/

sudo systemctl enable confluent-kafka-connect.service --- enabled

[root@lmu22repkafv001 kafka]# cat connect-distributed.properties

1. **confluent-kafka-connect.service systemctl** details

1. **Start services :**

Before running replicator make sure confluent-kafka-connect.service running on replicator/connect services on both nodes

systemctl start confluent-kafka-connect.service

**For replicator** :

To Post :

curl -X POST -d @/etc/kafka/replicator.json http://localhost:8083/connectors --header "content-Type:application/json" | jq

To delete:

curl -X DELETE http://localhost:8083/connectors/replicator --header "content-Type:application/json"

Status:

curl http://localhost:8083/connectors/replicator/status | jq

For Zookeeper:

systemctl start confluent-zookeeper

For Broker:

systemctl start confluent-kafka

1. **WIP – Sync confluent control center C3 with LDAP**

1. **consumer groups –list**

To get more details

bash -x /bin/kafka-run-class kafka.admin.ConsumerGroupCommand --bootstrap-server lmu22dbskaf002.ise.pos.net[67]:9093 –list

Java heap error message

Before

KAFKA_HEAP_OPTS=-Xmx**256**M

Edit kafka run class  and change Heap size

vi /bin/kafka-run-class

After

KAFKA_HEAP_OPTS=-Xmx**512**M

Use required parameters with –command-config (all parameters are defined in producer_ssl.properties)

Example:

[root@lmu22dbskaf002 ~]# /bin/kafka-run-class kafka.admin.ConsumerGroupCommand --bootstrap-server lmu22dbskaf001.ise.pos.net[68]:9093 --command-config /etc/kafka/producer_ssl.properties --list

Note: This will not show information about old Zookeeper-based consumers.

_confluent-controlcenter-4-1-1-UATLIM-command

_confluent-controlcenter-4-1-1-UATLIM

1. **Update Log4j with file Appender and Retentions:**

After some testing with different file appenders configured with below values and /var/logs are redirected

*used RollingFileAppender

Reason:

Deleting old log files automatically is desirable

Predictability in the space on disk occupied by logs is required

**After below changes** systemctl daemon-reload and restart services

**C3**:

Systemctl

Environment="LOG_DIR=/var/log/confluent/control-center"

Logs location: /var/log/confluent/control-center

**Kafka:**

67 http://lmu22dbskaf002.ise.pos.net
68 http://lmu22dbskaf001.ise.pos.net

Log locaiton: /var/log/kafka/
kafkaAppender and stateChangeAppender are key files so index files number set to 20 others set to 10
log4j.rootLogger=INFO, kafkaAppender

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=[%d] %p %m (%c)%n

log4j.appender.kafkaAppender=org.apache.log4j.RollingFileAppender
log4j.appender.kafkaAppender.File=/var/log/kafka/server.log
log4j.appender.kafkaAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.kafkaAppender.layout.ConversionPattern=[%d] %p %m (%c)%n
log4j.appender.kafkaAppender.Append=true
log4j.appender.kafkaAppender.MaxBackupIndex=20
log4j.appender.kafkaAppender.MaxFileSize=50MB

log4j.appender.stateChangeAppender=org.apache.log4j.RollingFileAppender
log4j.appender.stateChangeAppender.File=/var/log/kafka/state-change.log
log4j.appender.stateChangeAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.stateChangeAppender.layout.ConversionPattern=[%d] %p %m (%c)%n
log4j.appender.stateChangeAppender.Append=true
log4j.appender.stateChangeAppender.MaxBackupIndex=20
log4j.appender.stateChangeAppender.MaxFileSize=50MB

log4j.appender.requestAppender=org.apache.log4j.RollingFileAppender
log4j.appender.requestAppender.File=/var/log/kafka/kafka-request.log
log4j.appender.requestAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.requestAppender.layout.ConversionPattern=[%d] %p %m (%c)%n
log4j.appender.requestAppender.Append=true
log4j.appender.requestAppender.MaxBackupIndex=10
log4j.appender.requestAppender.MaxFileSize=50MB

log4j.appender.cleanerAppender=org.apache.log4j.RollingFileAppender
log4j.appender.cleanerAppender.File=/var/log/kafka/log-cleaner.log
log4j.appender.cleanerAppender.layout=org.apache.log4j.PatternLayout

log4j.appender.cleanerAppender.layout.ConversionPattern=[%d] %p %m (%c)%n

log4j.appender.cleanerAppender.Append=true

log4j.appender.cleanerAppender.MaxBackupIndex=10

log4j.appender.cleanerAppender.MaxFileSize=50MB

log4j.appender.controllerAppender=org.apache.log4j.RollingFileAppender

log4j.appender.controllerAppender.File=/var/log/kafka/controller.log

log4j.appender.controllerAppender.layout=org.apache.log4j.PatternLayout

log4j.appender.controllerAppender.layout.ConversionPattern=[%d] %p %m (%c)%n

log4j.appender.controllerAppender.Append=true

log4j.appender.controllerAppender.MaxBackupIndex=10

log4j.appender.controllerAppender.MaxFileSize=50MB

log4j.appender.authorizerAppender=org.apache.log4j.RollingFileAppender

log4j.appender.authorizerAppender.File=/var/log/kafka/kafka-authorizer.log

log4j.appender.authorizerAppender.layout=org.apache.log4j.PatternLayout

log4j.appender.authorizerAppender.layout.ConversionPattern=[%d] %p %m (%c)%n

log4j.appender.authorizerAppender.Append=true

log4j.appender.authorizerAppender.MaxBackupIndex=10

log4j.appender.authorizerAppender.MaxFileSize=50MB

By default systmctl will use log4j.properties file – no need to change Environment from systemctl

**Zookeeper:**

By default ZK serive will use log4j.properties (as per recommondation zk should run on spearete nodes), because we are running both zookeeper and kafka on same server required to use anohter log4j file

Log location: /var/log/kafka/

Created new file with name zk-log4j.properties on all zk server

Add parameters to systemctl

Environment="KAFKA_LOG4J_OPTS=-Dlog4j.configuration=file:/etc/kafka/zk-log4j.properties[69]"

[root@lmu22dbskaf002 kafka]# cat zk-log4j.properties

############## zkAppender #################

69 http://file/etc/kafka/zk-log4j.properties

log4j.rootLogger=INFO, zkAppender

log4j.appender.zkAppender=org.apache.log4j.RollingFileAppender

log4j.appender.zkAppender.File=/var/log/kafka/zkserver.log

log4j.appender.zkAppender.layout=org.apache.log4j.PatternLayout

log4j.appender.zkAppender.layout.ConversionPattern=[%d] %p %m (%c)%n

log4j.appender.zkAppender.Append=true

log4j.appender.zkAppender.MaxBackupIndex=10

log4j.appender.zkAppender.MaxFileSize=50MB

**Connect/replicator service:**

Create folder kafka-connect -- /var/log/kafka-connect/ with below permisisons

drwxr-x---. 2 cp-kafka-connect  confluent

Add parameters to systemctl

Environment="KAFKA_LOG4J_OPTS=-Dlog4j.configuration=file:/etc/kafka/connect-log4j.properties[70]"

Cat connect-log4j.properties:

log4j.rootLogger=INFO, fileLog

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=[%d] %p %m (%c:%L)%n

log4j.appender.fileLog=org.apache.log4j.RollingFileAppender

log4j.appender.fileLog.File=/var/log/kafka-connect/file.log

log4j.appender.fileLog.layout=org.apache.log4j.PatternLayout

log4j.appender.fileLog.layout.ConversionPattern=[%d] [%t] [%m]%n

log4j.appender.fileLog.MaxBackupIndex=10

log4j.appender.fileLog.MaxFileSize=50MB

log4j.logger.org[71].apache.zookeeper=ERROR

log4j.logger.org[72].I0Itec.zkclient=ERROR

70 http://file/etc/kafka/connect-log4j.properties
71 http://log4j.logger.org
72 http://log4j.logger.org

log4j.logger.org[73].reflections=ERROR

1. **JMX changes for Mbeans:**

For Simple Authorization Created jmx.access and jmx.password with chmod 600 under /etc/jmx/

And chown my user and group from systemctl

**cat /etc/jmx/jmx.access**

user readonly

admin readwrite

**cat /etc/jmx/jmx.password**

//jmx.password

user xxxxxx

admin xxxxx

For **Brokers**

JMX files (**jmx.access and jmx.password**) will be owned by cp-kafka:confluent

Add Below config to systemctl

"KAFKA_JMX_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8181 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxremote.access.file=/etc/jmx/jmx.access -Dcom.sun.management.jmxremote.password.file=/etc/jmx/jmx.password"

For **C3**

JMX files (**jmx.access and jmx.password**) will be owned by cp-control-center:confluent

Add Below config to systemctl

"CONTROL_CENTER_JMX_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8181 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxremote.access.file=/etc/jmx/jmx.access -Dcom.sun.management.jmxremote.password.file=/etc/jmx/jmx.password"

For **Connect**

JMX files (**jmx.access and jmx.password**) will be owned by cp-kafka-connect:confluent

Add below to systemctl

"KAFKA_JMX_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8181 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxremote.access.file=/etc/jmx/jmx.access -Dcom.sun.management.jmxremote.password.file=/etc/jmx/jmx.password"

73 http://log4j.logger.org

For Schema Registry

JMX files (**jmx.access and jmx.password**) will be owned by cp-schema-registry:confluent

Add below to systemctl

"SCHEMA_REGISTRY_JMX_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8181 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxremote.access.file=/etc/jmx/jmx.access -Dcom.sun.management.jmxremote.password.file=/etc/jmx/jmx.password"

Restart services for effective changes

1. **Process to Enable ACL:**

Add below to /etc/kafka/server.properties on all brokers

###################### ACL ######################

authorizer.class.name[74]=kafka.security.auth.SimpleAclAuthorizer

allow.everyone.if.no[75].acl.found=false

super.users=User:kafka

################################################

In a secure cluster, both client requests and inter-broker operations require authorization

Error:

ERROR Error checking or creating metrics topic (io.confluent.metrics.reporter.ConfluentMetricsReporter)

org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [Topic authorization failed.])

To resolve:

Add below to **kafka_server_jaas.conf** file (principal name should be relevant to host) on all brokers

(already contains KafkaServer and Zk client details in **kafka_server_jaas.conf** file)

// Kafka client authentication (Metrics Reporter)

KafkaClient {

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab=true

storeKey=true

keyTab="/etc/security/keytabs/kafka_server.keytab"

74 http://authorizer.class.name
75 http://allow.everyone.if.no

principal="kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL[76]";
};

Restart confluent-kafka service

User:kafka -- not found any errors related to this

found access errors related to Users: kafkacontrolcenter and kafkaconnect

error related to kafkaclient (internal metrics/c3 metrics)

Errors also found related to replcicator, not showind any streams from conenct service and data flow is not working --- to resolve this provide access to kafkareplicator

So, required to provide ACLs manually

kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:**kafkaconnect** --operation All --cluster

kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:**kafkaconnect** --operation All --topic '*' --cluster

kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:**kafkaconnect** --operation All --topic '*' --group '*' --cluster

Repeat above steps for user **kafkaclient** , **kafkacontrolcenter** , **kafkaschemaregistry** and **kafkareplicator**

Because we are using kafka replicator for replicaiton (found auth errors in kafka loga)

All above principals (kafka, kafkaconnect, kafakreplicator etc )are service principals --- not supposed to share with users

Errors resolved after adding ACL to kafkaclient user

*For users required to create new principal and provide relevant access for producer and consumer

**After enabling check, the ACL list**

[root@lmu22dbskaf001 certs]# kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --list

Current ACLs for resource `Topic:midhun-after-acl`:

User:mpolisetty has Allow permission for operations: All from hosts: *

Current ACLs for resource `Group:*`:

User:kafkacontrolcenter has Allow permission for operations: All from hosts: *

User:kafkaclient has Allow permission for operations: All from hosts: *

User:kafkaconnect has Allow permission for operations: All from hosts: *

Current ACLs for resource `Topic:*`:

User:kafkacontrolcenter has Allow permission for operations: All from hosts: *

76 mailto:principal=%22kafka/lmu22dbskaf001.ise.pos.net@AOD.LOCAL

User:kafkaclient has Allow permission for operations: All from hosts: *

User:kafkaconnect has Allow permission for operations: All from hosts: *

Current ACLs for resource `Cluster:kafka-cluster`:

User:kafkaclient has Allow permission for operations: All from hosts: *

User:kafkaconnect has Allow permission for operations: All from hosts: *

**From C3 server:**

control center streams application experienced an exception related to its state store.

To resolve this

[root@lmu22conkafv001 control-center]# pwd

/var/lib/confluent/control-center

[root@lmu22conkafv001 control-center]# ls -ltr

total 0

drwxr-xr-x. 4 cp-control-center confluent 43 Jun 6 22:05 1

drwxr-xr-x. 4 cp-control-center confluent 43 Jun 7 16:44 UATLIM

[root@lmu22conkafv001 control-center]# rm -rf ./*

**Restart confluent-control-center.service**

1. **Forward aggregated logs to ELK with log4j**

WIP

1. **Oracle to OpenJDK:**

Joe Chambers from Infra team installed OpenJDK on all servers and made changes to /etc/yum.conf so this won't be updated

[root@nxu22dbskaf003 yum.repos.d]# cat /etc/yum.conf

[main]

cachedir=/var/cache/yum/$basearch/$releasever

keepcache=0

debuglevel=2

logfile=/var/log/yum.log

exactarch=1

obsoletes=1

gpgcheck=1

plugins=1

installonly_limit=3

exclude=java-1.8.0*

Changes required for both alternatives --config java and alternatives --config javac

select option 2 for both java and javac

**After changes** check below :

[root@lmu22repkafv002 ~]# alternatives --config java

There are 2 programs which provide 'java'.

```
  Selection    Command
-----------------------------------------------
   1           /usr/java/jdk1.8.0_131/jre/bin/java
*+ 2           java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.161-2.b14.el7_4.x86_64/jre/bin/java)
```

and

[root@lmu22repkafv002 ~]# alternatives --config javac

There are 2 programs which provide 'javac'.

```
  Selection    Command
-----------------------------------------------
   1           /usr/java/jdk1.8.0_131/bin/javac
*+ 2           java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.161-2.b14.el7_4.x86_64/bin/javac)
```

verify

[root@lmu22repkafv002 java]# rpm -qf /usr/share/java

javapackages-tools-3.4.1-11.el7.noarch

And

ls -ltr /usr/share/java

Add ldaprootca and rootca to java truststore

keytool -import -trustcacerts -keystore /etc/pki/ca-trust/extracted/java/cacerts -storepass changeit -noprompt -alias ldaprootca -file /etc/openldap/cacerts/ldap.aod.local-ca.pem

keytool -import -trustcacerts -keystore /etc/pki/ca-trust/extracted/java/cacerts -storepass changeit -noprompt -alias rootca -file /etc/security/certs/Hadoop_QA_Root_CA.cer

Test with java -jar my-bench-jar-all and restart all services without errors

1. **Schema Registry setup in HA:**

Complete prechecks and update kafka_clinet keytab with kafascheamregistry principal

*kafkaschemaregistry@AOD.LOCAL(see page 3) added to existing kafka_client.keytab

create users with create_kafka_users_and_groups.sh

yum install confluent-platform-2.11

sudo systemctl enable confluent-schema-registry

Create new certs for servers

Create keystoere and update truststore accordingly (with all server certs)

copy keystore/truststore @ /etc/security/certs

copy kafak_client to schema registry servers @ /etc/security/keytabs

from /etc/schema-registryschema-registry.properties (example for server1 – relevant details should be added to server2 like listener etc)

listeners=https://lmu22scrkafv001.ise.pos.net:8081

kafkastore.connection.url=lmu22dbskaf001.ise.pos.net[77]:2181,lmu22dbskaf002.ise.pos.net[78]:2181,lmu22dbskaf003.ise.pos.net[79]:2181

kafkastore.bootstrap.servers=SASL_SSL://lmu22dbskaf001.ise.pos.net:9093,SASL_SSL://lmu22dbskaf002.ise.pos.net:9093,SASL_SSL://lmu22dbskaf003.ise.pos.net:9093(see page 3)

kafkastore.topic=_schemas

kafkastore.topic.replication.factor=3

debug=false

###### SASL_SSL ########

kafkastore.ssl.truststore.location=/etc/security/certs/truststore.jks

kafkastore.ssl.truststore.password=xxxx

kafkastore.ssl.truststore.type=jks

kafkastore.ssl.keystore.location=/etc/security/certs/keystore.jks

kafkastore.ssl.keystore.password=xxxx

kafkastore.ssl.key.password=xxx

kafkastore.ssl.keystore.type=jks

kafkastore.security.protocol=SASL_SSL

77 http://lmu22dbskaf001.ise.pos.net
78 http://lmu22dbskaf002.ise.pos.net
79 http://lmu22dbskaf003.ise.pos.net

kafkastore.sasl.mechanism=GSSAPI

kafkastore.ssl.enabled.protocols=TLSv1.2

kafkastore.sasl.kerberos.service.name[80]=kafka

kafkastore.sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required \

useKeyTab=true \

storeKey=true \

keyTab="/etc/security/keytabs/kafka_client.keytab" \

principal="kafkaschemaregistry@AOD.LOCAL[81]";

###### HTTPS #########

ssl.keystore.location=/etc/security/certs/keystore.jks

ssl.keystore.password=xxxx

ssl.keystore.type=jks

ssl.key.password=xxxx

ssl.truststore.location=/etc/security/certs/truststore.jks

ssl.truststore.type=jks

ssl.truststore.password=xxxx

ssl.client.auth=false

####### Additional ########

host.name[82]=nxu22scrkafv001.ise.pos.net[83]

zookeeper.set.acl=false

master.eligibility=true

schema.registry.inter.instance.protocol=https

Add permisisons to kafkaschemaregistry - details added to ACL section

add trsustsotre to java

keytool -import -trustcacerts -keystore /etc/pki/ca-trust/extracted/java/cacerts -storepass changeit -noprompt -alias ldaprootca -file /etc/openldap/cacerts/ldap.aod.local-ca.pem

keytool -import -trustcacerts -keystore /etc/pki/ca-trust/extracted/java/cacerts -storepass changeit -noprompt -alias rootca -file /etc/security/certs/Hadoop_QA_Root_CA.cer

80 http://kafkastore.sasl.kerberos.service.name
81 mailto:principal=%22kafkaschemaregistry@AOD.LOCAL
82 http://host.name
83 http://nxu22scrkafv001.ise.pos.net

**Changes to Log4j:**

[root@lmu22scrkafv001 schema-registry]# cat log4j.properties

log4j.rootLogger=INFO, stdout, file

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=[%d] %p %m (%c:%L)%n

log4j.logger.kafka=ERROR, stdout

log4j.logger.org[84].apache.zookeeper=ERROR, stdout

log4j.logger.org[85].apache.kafka=ERROR, stdout

log4j.logger.org[86].I0Itec.zkclient=ERROR, stdout

log4j.additivity.kafka.server=false

log4j.additivity.kafka.consumer.ZookeeperConsumerConnector=false

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.maxBackupIndex=20

log4j.appender.file.maxFileSize=50MB

log4j.appender.file.File=${schema-registry.log.dir}/schema-registry.log

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=[%d] %p %m (%c)%n

**Testing:**

root@nxu22scrkafv001 schema-registry]# kafka-avro-console-producer --broker-list nxu22dbskaf001.ise.pos.net[87]:9093,nxu22dbskaf002.ise.pos.net[88]:9093,nxu22dbskaf003.ise.pos.net[89]:9093 --topic

grinder_test --property schema.registry.url=https://nxu22scrkafv001.ise.pos.net:8081 --property value.schema='{"type":"record","name":"myrecord","fields":[{"name":"f1","type":"string"}]}'

--producer.config /etc/schema-registry/producer_ssl.properties

84 http://log4j.logger.org
85 http://log4j.logger.org
86 http://log4j.logger.org
87 http://nxu22dbskaf001.ise.pos.net
88 http://nxu22dbskaf002.ise.pos.net
89 http://nxu22dbskaf003.ise.pos.net

{"f1":"value1"}

[2018-09-14 09:36:31,477] DEBUG Sending POST with input {"schema":"{\"type\":\"record\",\"name\":\"myrecord\",\"fields\":[{\"name\":\"f1\",\"type\":\"string\"}]}"} to

https://nxu22scrkafv001.ise.pos.net:8081/subjects/grinder_test-value/versions
(io.confluent.kafka.schemaregistry.client.rest.RestService:146)

{"f1":"value2"}

[root@nxu22scrkafv002 schema-registry]# kafka-avro-console-consumer --bootstrap-server nxu22dbskaf001.ise.pos.net[90]:9093,nxu22dbskaf002.ise.pos.net[91]:9093,nxu22dbskaf003.ise.pos.net[92]:9093 --topic

grinder_test --property schema.registry.url=https://nxu22scrkafv001.ise.pos.net:8081 --consumer.config /etc/schema-registry/consumer_ssl.properties

[2018-09-14 09:36:32,937] DEBUG Sending GET with input null to https://nxu22scrkafv001.ise.pos.net:8081/schemas/ids/81 (io.confluent.kafka.schemaregistry.client.rest.RestService:146)

{"f1":"value1"}

{"f1":"value2"}

[root@lmu22scrkafv001 schema-registry]# kafka-avro-console-producer --broker-list lmu22dbskaf001.ise.pos.net[93]:9093,lmu22dbskaf002.ise.pos.net[94]:9093,lmu22dbskaf003.ise.pos.net[95]:9093 --topic

grinderlimerick_test --property schema.registry.url=https://lmu22scrkafv001.ise.pos.net:8081 --property value.schema='{"type":"record","name":"myrecord","fields":

[{"name":"f1","type":"string"}]}' --producer.config /etc/schema-registry/producer_ssl.properties

{"f1":"value1"}

[2018-09-14 16:15:46,067] DEBUG Sending POST with input {"schema":"{\"type\":\"record\",\"name\":\"myrecord\",\"fields\":[{\"name\":\"f1\",\"type\":\"string\"}]}"} to

https://lmu22scrkafv001.ise.pos.net:8081/subjects/grinderlimerick_test-value/versions
(io.confluent.kafka.schemaregistry.client.rest.RestService:146)

{"f1":"value2"}

90 http://nxu22dbskaf001.ise.pos.net
91 http://nxu22dbskaf002.ise.pos.net
92 http://nxu22dbskaf003.ise.pos.net
93 http://lmu22dbskaf001.ise.pos.net
94 http://lmu22dbskaf002.ise.pos.net
95 http://lmu22dbskaf003.ise.pos.net

[root@lmu22scrkafv002 schema-registry]# kafka-avro-console-consumer --bootstrap-server lmu22dbskaf001.ise.pos.net[96]:9093,lmu22dbskaf002.ise.pos.net[97]:9093,lmu22dbskaf003.ise.pos.net[98]:9093 --topic

grinderlimerick_test --property schema.registry.url=https://lmu22scrkafv001.ise.pos.net:8081 --consumer.config /etc/schema-registry/consumer_ssl.properties

[2018-09-14 16:15:53,553] DEBUG Sending GET with input null to https://lmu22scrkafv001.ise.pos.net:8081/schemas/ids/41 (io.confluent.kafka.schemaregistry.client.rest.RestService:146)

{"f1":"value1"}

….....

….....

96 http://lmu22dbskaf001.ise.pos.net
97 http://lmu22dbskaf002.ise.pos.net
98 http://lmu22dbskaf003.ise.pos.net