

SASL

(SIMPLE AUTHENTICATION AND SECURITY LAYER)

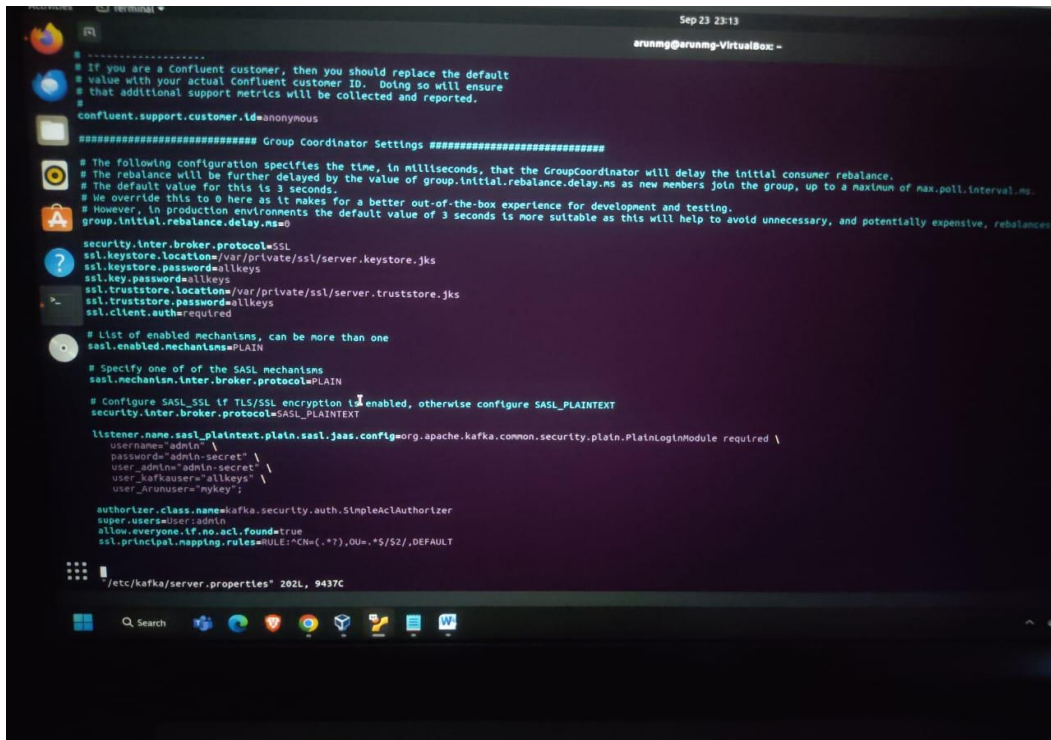
SASL is mainly used for Authentication. We are implementing SASL in the kafka for client authorization. Which is used for secure communication between client and brokers. Kafka supports several authentication mechanisms via SASL and we are using PLAIN mechanism.

So we create SASL security layer in kafka in my **local** system. Which is similar to as creating Ssl so here are the steps :

there are some SASL mechanisms in kafka here we are using PLAIN mechanism by which clients are authenticated by their **username and passwords**.

Other mechanisms are used for different purposes so used in production environment and enterprises that these mechanisms use many techniques like hashing ,encoding etc. Here we are using just simple PLAIN (username and password).

First step : To edit the server.properties in local host :-



```
##### Group Coordinator Settings #####
# If you are a Confluent customer, then you should replace the default
# value with your actual Confluent customer ID. Doing so will ensure
# that additional support metrics will be collected and reported.
confluent.support.customer.id=anonymous

# The following configuration specifies the time, in milliseconds, that the GroupCoordinator will delay the initial consumer rebalance.
# The rebalance will be further delayed by the value of group.initial.rebalance.delay.ms as new members join the group, up to a maximum of max.poll.interval.ms.
# The default value for this is 3 seconds.
# We override this to 0 here as it makes for a better out-of-the-box experience for development and testing.
# However, in production environments the default value of 3 seconds is more suitable as this will help to avoid unnecessary, and potentially expensive, rebalances.
group.initial.rebalance.delay.ms=0

security.inter.broker.protocol=SSL
ssl.keystore.location=/var/private/ssl/server.keystore.jks
ssl.keystore.password=allkeys
ssl.key.password=allkeys
ssl.truststore.location=/var/private/ssl/server.truststore.jks
ssl.truststore.password=allkeys
ssl.client.auth=required

# List of enabled mechanisms, can be more than one
sasl.enabled.mechanisms=PLAIN

# Specify one of the SASL mechanisms
sasl.mechanism.inter.broker.protocol=PLAIN

# Configure SASL_SSL if TLS/SSL encryption is enabled, otherwise configure SASL_PLAINTEXT
security.inter.broker.protocol=SASL_PLAINTEXT

listener.name.sasl_plaintext.plain.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
  username=admin \
  password=admin-secret \
  user_admin=admin-secret \
  user_kafkauser=allkeys \
  user_arunuser=mykey;

authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer
super.users=User:admin
allow.everyone.if.no.acl.found=true
ssl.principal.mapping.rules=RULE:*CN=(*?),OU=,*S*/,DEFAULT
```

As you see in the picture there are lot of configurations of server there are SSL configurations , SASL configurations , and also ACL setting is there

List of enabled mechanisms, can be more than one

sasl.enabled.mechanisms=PLAIN

Specify one of the SASL mechanisms

sasl.mechanism.inter.broker.protocol=PLAIN

Configure SASL_SSL if TLS/SSL encryption is enabled, otherwise configure SASL_PLAINTEXT

security.inter.broker.protocol=SASL_PLAINTEXT

listener.name.sasl_plaintext.plain.sasl.jaas.config=org.apache.kafka.common.security.plain.

PlainLoginModule required \

username="admin" \

password="admin-secret" \

user_admin="admin-secret" \

user_kafkauser="allkeys" \

user_Arunuser="mykey";

explanation : In the first property we are just enabling different mechanisms for further uscases.

In the second property we are specifying that which mechanism we are actually going to use here that is PLAIN which is used in simple use cases like local environments that clients are authenticated by username and password.

In the third property setting we are mentioned that security protocol to use that is SASL_PLAINTEXT .

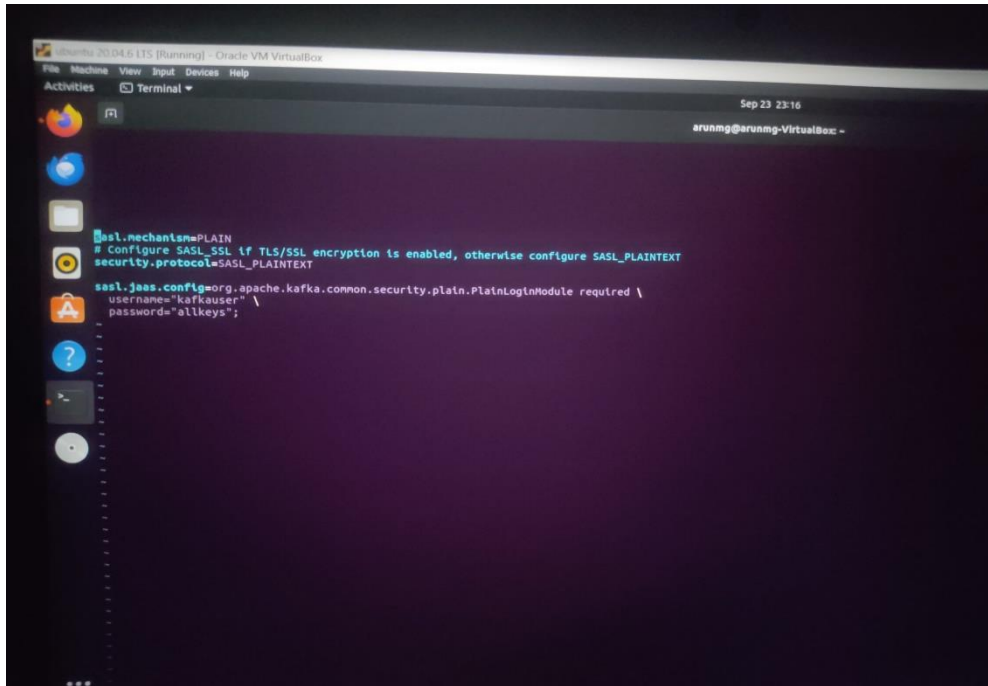
In fourth property that **listener.name.sasl_plaintext.plain.sasl.jaas.config** this specifies that jaas configuration file for SASL authentication of PLAINTEXT mechanism in which we explain that credentials of **ADMIN** and **USERS** (clients) .

We also add listeners for SASL_PLAIN ie

listeners=PLAINTEXT://localhost:9092,SSL://localhost:9093,SASL_PLAINTEXT://localhost:9094

On which port kafka will hear SASL communication

Then for client side configuration ex : client-sasl.properties



In the above picture we have seen that client—sasl.properties for kafkauser in local host
We have to mention authentication mechanism type that is PLAIN.

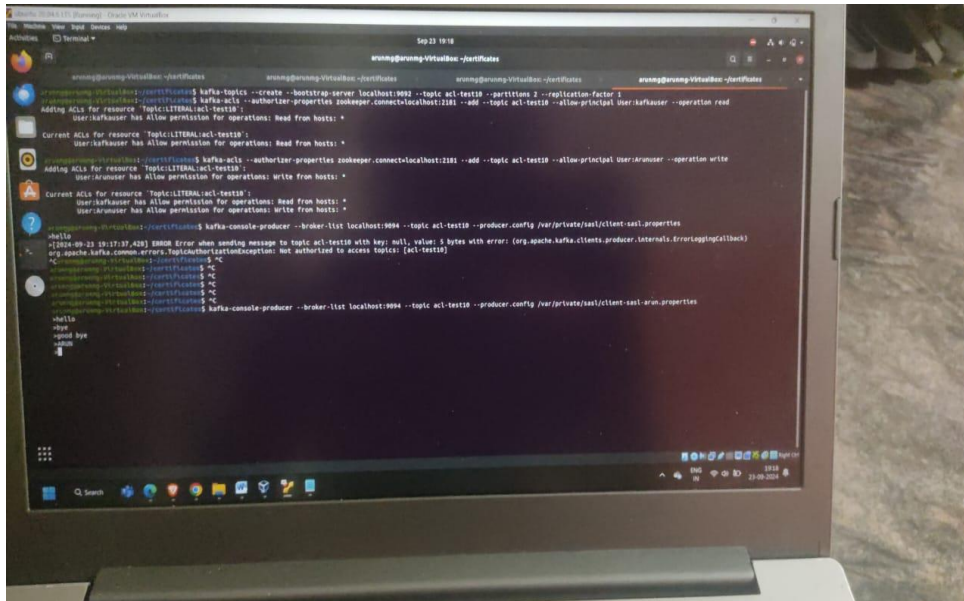
And we have to mention the Security.protocol that we are using for secure authentication
SASL_PLAINTEXT.

Finally we are using the property of jaas file in that jaas file we have to mention username and
password of user i.e

```
Username="kafkauser"\  
Password="allkeys" ;
```

This will be authenticated when we will be producing messages to topic upon our credentials
Which will be used in when **ACLs** are created on topic or any other resources of kafka. SASL
will authenticate user and then proceed it .

SASL USE CASES :



First we created two client-SASL properties in the directory of /var/private/kafka/sasl/client-sasl.properties for **kafkauser** and mentioned credentials for kafkauser

second we created two client-SASL properties in the directory of /var/private/kafka/sasl/client-sasl.properties for **Arunuser** and mentioned credentials for Arunuser

For first command : Here we are creating acl-test10 named topic to check SASL authentication

Insecond command : we are creating acl on that topic acl-test10 topic and mentioned that principal as USER: kafkauser and mentioned operation that allowed for kafkauser to read only

In third command : we are creating another acl on sam topic acl-test10 that we are mentioned principal as USER: Arunuser and allowed operation only to write to topic

Now we are just producing messages to topic using client-sasl.properties as you seen in the above picture then it had prompted me to write when we trying to message and hit the enter it thrown the error bcz : kafka user have only allowed to read operation not to write to topic

In other hnd when we tried to produce message with client-sasl-arun.properties then it will prompted me to write message when I hit the enter it wil further prompted me to send message that's mean Arun user hve write operation to send or pulish mssages to topic .

But when we tried to consume messages using `client-sasl-arun.properties` of Arunuser on topic we got error of AUTHORIZATION that's bcz Aunuser has only produce messages to topic not read or consume messages from topic.