#### ACL (access controle lists) DOCUMENT

**1.Create a Test Topic**

kafka-topics --bootstrap-server zoo1:9092 --create --topic acl-test --partitions 1 --replication-factor 1

* --bootstrap-server zoo1:9092: Connect to Kafka broker running on zoo1:9092.
* --create --topic acl-test: Create a new topic named acl-test.
* --partitions 1: Create a single partition for the topic.
* --replication-factor 1: Use one broker for topic replication (this can be adjusted as needed).

In this command we are creating topic as acl-test for testing acl .

**2. Enable ACL Authorization in server.properties**

On **all three brokers**, you need to configure the server.properties file to enable ACL authorization.

sudo vi /etc/kafka/server.properties

Adding the following lines to enable authorization:

properties

Copy code

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=.\*$/$1/,DEFAULT

* authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer: Enables Kafka's simple ACL authorizer.
* super.users=User:admin: Configures the superuser (e.g., admin) who will have access to everything.
* allow.everyone.if.no.acl.found=true: Allows access when no ACL exists for a resource.
* ssl.principal.mapping.rules: A rule for mapping SSL principals (e.g., certificates) to usernames.

**3. Restarting Kafka Services**

After editing the configuration, we have to restart the Kafka services on each broker:

sudo systemctl restart confluent-kafka

To check the status to confirm that Kafka is running without any issues

sudo systemctl status confluent-kafka

**4. Write Data to the Topic**

Now, write data to the acl-test topic. Since no ACL is set yet and allow.everyone.if.no.acl.found is set to true, this should succeed.

kafka-console-producer --broker-list zoo1:9093 --topic acl-test --producer.config client-ssl.properties

* --broker-list zoo1:9093: Specify the Kafka broker.
* --topic acl-test: The topic to write data to.
* --producer.config client-ssl.properties: SSL configuration file for secure communication.

Here we are connecting to cluster with .9093 port which is secure port to connect kafka cluster so to cnnect to 9093 port we have to produce client ssl properties by using with command --producer.configg

**5. Add an ACL for otheruser**

kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:otheruser --operation Write --topic acl-test

* --authorizer-properties zookeeper.connect=localhost:2181: Connects to Zookeeper.
* --add --allow-principal User:otheruser: Grants write permission to otheruser.
* --operation Write: The operation allowed (in this case, Write).
* --topic acl-test: Specifies that the ACL applies to the acl-test topic.

Here we are using above command to create acl on kafka soursces here we are we are creating acl on topic (which is kafka source) and we are mentioned that –allow-principal that User:Otheruser it’s mans the acl permitted otheruser role to perform write to topic acl-test

**6. Attempt to Write to the Topic as Another User**

Now, attempt to write to the acl-test topic using a different user, such as kafkauser. Since the ACL only allows otheruser to write, this should fail.

kafka-console-producer --broker-list zoo1:9093 --topic acl-test --producer.config client-ssl.properties

.we have created acl on topic acl-test soo .If we would be producing messages to acl-test topic the kafka with 9093 port that willl throws AUTHORIZATION ERROR bcz acl has been created on that topic

**7. Add an ACL for kafkauser**

kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:kafkauser --operation Write --topic acl-test

Now we are adding another acl for kafkaUser principalname this acl is permitting kafkausr to write operation on acl-test topic . Here Kafka user is nothing but admin that now he can easily write to that toopic .

**8. Test Writing to the Topic Again**

kafka-console-producer --broker-list zoo1:9093 --topic acl-test --producer.config client-ssl.properties

After adding the proper ACLs, kafkauser should now be able to produce messages to the acl-test topic.

**Explanation of Key Concepts**

* **ACLs (Access Control Lists)**: These are rules used to define which users or groups have permission to perform specific operations on Kafka topics (e.g., Read, Write, Delete).
* **Superusers**: Superusers (defined by super.users) have full access to the Kafka cluster and bypass ACLs.
* **SSL Authentication**: Kafka uses SSL to authenticate users. In this example, the ssl.principal.mapping.rules map SSL certificates to Kafka user identities (e.g., CN=kafkauser).
* **Zookeeper**: Zookeeper is required to manage ACLs, as Kafka stores ACLs in Zookeeper.