Authorization in CC

Authorization via ACLs and RBAC

# Objectives

Learn how to assign permissions in Confluent Cloud via ACLs and RBAC. Gain practical experience with the different setups, and understand how to use authorization effectively.

# Labs

## Setup

* Ensure you have the two topics “test-topic” and “customer” set up.

## ACLs (Access Control Lists)

Use API Keys with granular access, linked to a service account, and ACLs to restrict access to specific resources such as topics.

* Create a new API Key
  + Choose the Service Account API option
  + Create a new service account. Name it, for example, “bootcamp-user”, and give it a description if you like.
  + When you press next, you can choose ACLs for this API Key.
    - Go to Topic
      * Pick, for example, the “test-topic”
      * Pattern Type “LITERAL”
      * Operation “Write”
      * Permission “Allow”
  + Give the API Key a description and download it
* Create a new Java client properties file, ensure that the API Key you just created is inserted (otherwise copy the key and secret from the downloaded JSON file).
* Use kafka-console-producer to write to the test-topic using the property file you just created. This should succeed!
* Try to use kafka-console-consumer to read from the topic. This should fail!
* You can edit the ACLs for the API Key you just created in the UI by choosing the Key and then clicking on access. You need to add at least two more ACLs to be able to use this key for the consumer:
  + The error message indicates that you cannot access a consumer group. You will require READ access.   
    Note that the name of the consumer group changes if you rerun the application, so you will have to pick a PREFIXED ACL.
  + Try to run the console-consumer again. It will still fail, but this time with a different error. Add the required ACL(s), be precise.
* Now try to create a new topic, for example, “another-topic” using the command line tool kafka-topics with the same API Key. This will fail, even after you figure out the syntax since you lack the correct authorization.
  + Add the missing ACL(s) to create this topic, again be precise.
* Try to list all topics using the kafka-topics command. Note that not all topics are visible, including your new “another-topic”. We need yet more ACLs.
  + Give your API Key permissions to describe your new topic, and see if it shows up in the list now.
  + You can use the kafka-topics command with the option --describe to get more details on a topic (specified with --topic <name>). Yet, although you gave yourself “describe” permissions, this will still fail. What is missing? Add the correct ACL(s).
* Create another Granular API Key for the same service user. Note how all the ACLs you defined have been recreated for you.
  + Create another property file with the new credentials and rerun some of the above commands to verify that you have access.

## RBAC (Role-base access control)

To be able to use role-based access control for service users for topics, you need to upgrade your cluster from BASIC to STANDARD, which we will do now.

* Navigate to Cluster Settings. Next to the cluster type is a Panel with “Upgrade available” written on the top. Press “See details”, then upgrade your cluster to Standard.

RBAC can provide authorization for many more resources than ACLs can, but it can also be used to provide access to topics and consumer groups.

* Create another service user by pressing the “hamburger” menu 🍔(≡) in the top right corner and choosing “Accounts and Access”.
  + Switch to “Service Account”, then add another service account, for example, “bootcamp-rbac-user”.
  + When you press “Next”, you get to the Access menu. Choose the correct cluster (if you have more than one), then pick topics.
  + Add a role assignment “Resource Owner” for your test-topic
  + Save the user
  + Create a Service Account API Key for the user. You will have to pick one ACL for this user to be able to complete the next panel, choose, for example, DESCRIBE on the cluster. You can delete this ACL later if you like, it is not needed.
* Use the new API Key to write to your test topic, which should succeed.
* Use the same key to read from the topic, which will fail until you add “Developer Read” to the appropriate consumer group.
* Now that you have seen the pattern, try some additional ACLs and RBAC combinations to deepen your understanding of how authorization in the Confluent Cloud works.

# References

<https://docs.confluent.io/cloud/current/access-management/access-control/rbac/manage-role-bindings.html#manage-role-bindings-of-principals>

<https://docs.confluent.io/cloud/current/access-management/access-control/rbac/use-acls-with-rbac.html#use-acls-with-rbac>

<https://developer.confluent.io/courses/cloud-security/rbac-and-acls/#:~:text=As%20opposed%20to%20ACLs%2C%20RBAC,confirm%20each%20individual%20ACL%20identity>.

# Expected Outcomes

Set up authorization via ACL and RBAC to provide service accounts tailored access to specific resources.

Understand how authorization works, which permissions are required and which errors to expect when the correct authorization is not available.

# Check your understanding

This colour marks advanced questions.

* What is an ACL?
* What is RBAC?
  + How does RBAC differ from ACL?
* What is the principle of least privilege?
* Why do you need a specific ACL or RBAC to access a consumer group?

# 