OpenID Connect (OIDC)

Authentication via OIDC

# Objectives

Learn to set up and configure an OIDC provider with Confluent Cloud, and use it to authenticate a client.

# Labs

## Set up OIDC in Confluent Cloud

* Log into your Confluent Cloud account. Navigate to the Hamburger menu on the top right and choose “Accounts and Access”, then “Workload identities”.
* Add a new provider, choose Okta and press Next
* Give your provider a name and description
  + Add “dev-84536545.okta.com” as the Domain and leave the Authorization server as “default”
  + Press “Import from configuration”
  + Validate and save the configuration
* Now, you need to add an identity pool. Click on your newly created Identity Provider.
  + Press “Add Identity Pool”
  + Give the pool a name and a description, and leave the claim as “claims.sub”
  + Switch the filter to “Advanced”
  + In Okta, scopes are an array, not a single value. Add the filter expression. "ccloud" in claims.scp
* The next step is to provide some permissions to this identity.
  + Either choose “Add New Permissions” and provide the permissions yourself
  + Alternatively, import the permissions from a service account you configured earlier

You should now have the Okta account as an identity provider in your Organisation.

## Use OIDC in the clients

* Create a (Java) configuration file on your local machine with the following content:

| bootstrap.servers=<your CC bootstrap server>  security.protocol=SASL\_SSL  sasl.oauthbearer.token.endpoint.url=https://dev-84536545.okta.com/oauth2/default/v1/token  sasl.login.callback.handler.class=org.apache.kafka.common.security.oauthbearer.secured.OAuthBearerLoginCallbackHandler  sasl.mechanism=OAUTHBEARER  sasl.jaas.config= \  org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \  clientId='0oadgg8ushBMjLBHj5d7' \  clientSecret='Bo1w7o1FlpcUBRDG2LHtO6g-xaZN6Y5rp6HAoCHAOi7kdrYSw4SviJqpotBuwlH\_' \  scope='ccloud' \  extension\_logicalCluster='<your cluster id>' \  extension\_identityPoolId='<your pool id>'; |
| --- |

* Your cluster id can be found under Cluster Settings in the Identification box. It starts with “lkc-”.
* The pool id starts with “pool-” and can be found in the Identity Provider overview.
* Test your new configuration file with a Java application
* Create a librdkafka (C/C++ clients like kafkacat) configuration with the same parameters, following this template:

| bootstrap.servers=<your CC bootstrap server>  security.protocol=SASL\_SSL  sasl.mechanisms=OAUTHBEARER  sasl.oauthbearer.method=oidc  sasl.oauthbearer.scope=ccloud  sasl.oauthbearer.token.endpoint.url=https://dev-84536545.okta.com/oauth2/default/v1/token  sasl.oauthbearer.client.id=0oadgg8ushBMjLBHj5d7  sasl.oauthbearer.client.secret=Bo1w7o1FlpcUBRDG2LHtO6g-xaZN6Y5rp6HAoCHAOi7kdrYSw4SviJqpotBuwlH\_  sasl.oauthbearer.extensions=logicalCluster=<logical-cluster>,identityPoolId=<pool-id> |
| --- |

* Test your new configuration file with kcat/kafkacat application

# References

<https://docs.confluent.io/cloud/current/access-management/authenticate/oauth/identity-providers.html#add-an-oauth-oidc-identity-provider-for-ccloud>

# Expected Outcomes

Successfully configure an existing Okta account as a OIDC provider.

Successfully test out the provider with Java and C/C++ applications.

# Check your understanding

This colour marks advanced questions.

* TBD - something about which resources need to be created/maintained for each user?
* TBD - how about creating your own OpenID provider? Okta? AWS Cognito? Azure AD?

# 