PTT Pro

Workcloud Communication

Okta Integration Guide

MN-004831-01EN Rev A

Copyright

2024/01/22

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Introduction

Workcloud Communication supports OAuth2 authentication using the Okta identity provider. Using OAuth2, multiple users can share a single device. This is referred to as the Shared Device Model.

When you install a Workcloud Communication solution with AD/ADFS services, User Authentication, User Provisioning, and Attribute Transformations are possible using LDAP services. The Okta integration only supports User Authentication. User Provisioning and Attribute Transformations are supported indirectly by using the Flat File Import functionality provided by Workcloud Communication.

The Okta integration uses the OpenID Connect (OIDC) protocol, which requires the Authentication Connection Service (ACS). ACS enables Workcloud Communication to authenticate with Okta using OIDC.

This document describes the Okta and ACS configuration required to support user authentication, and how this affects the configuration of PTT Pro for Android and Workcloud Communication Profile Manager.

Requirements

You must have an Okta instance configured and running and the PTT Pro for Android users must be provisioned in the Okta IdP.

- Okta IdP is installed and in operation.
- Users are provisioned in the Okta IdP.

Configuration Overview

The configuration process is divided into four phases. Complete each phase sequentially because each phase uses configuration elements from the previous phase.

Phase 1
Establish the Realm and Client on the ACS server. The Realm is required to create a client, and the client URL is required to configure Okta.

Phase 2
Create the Application in the Okta server. In this phase, the Authentication and Token Access URLs are created along with the Client ID and Client Secret. These elements are required to create the Identity Provider connection in the ACS.

Phase 3
Create the Identity Provider and finalize the ACS configuration.

Phase 4
Configure the PTT Pro Server OAuth connection with the URLs and Certificate needed to connect to the ACS.

Phase 1 – Configuring the ACS

In Phase 1 you will create the realm, copy the signing certificate, configure the client, and create the user property matcher.

In Phase 1, complete the following tasks:

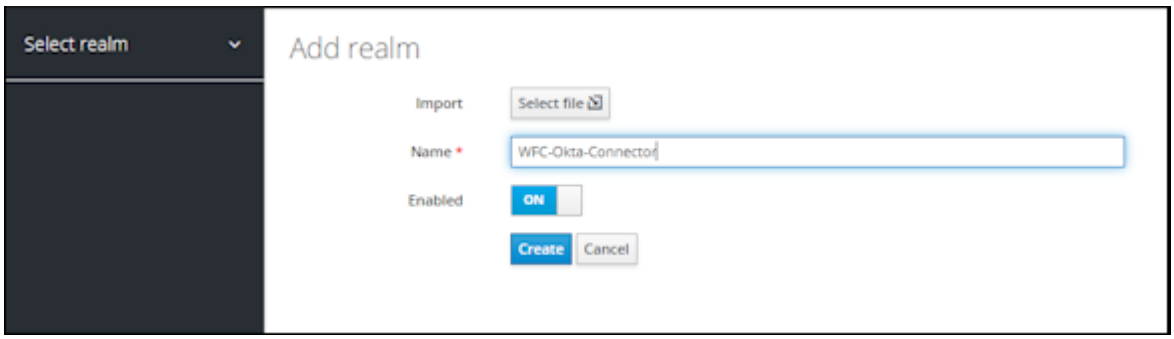
- Create the realm.
- Retrieve the signing certificate for the realm.
- Configure the client.
- Create the user property mapper.

- Capture the IdP redirect URL.

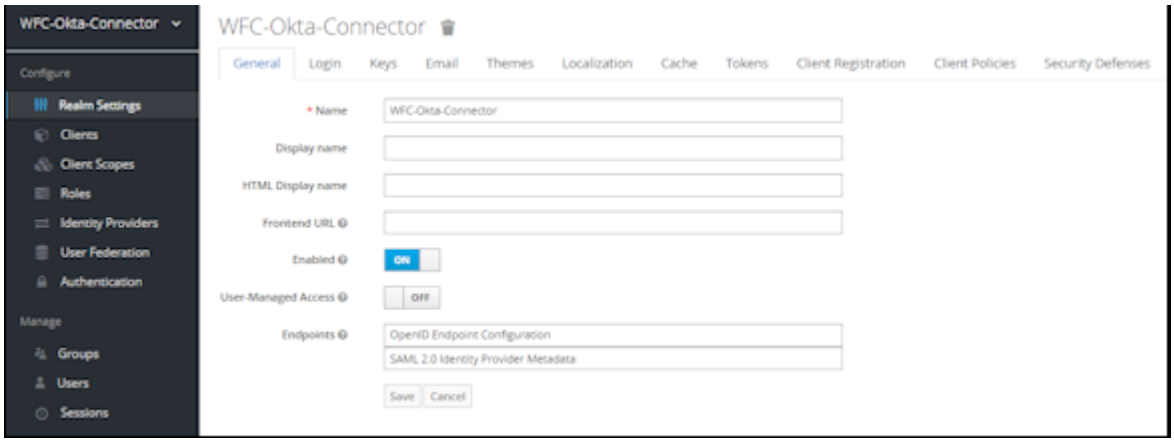
Creating the Realm

Create the realm and enter a meaningful name. The following example uses WFC-OKTA-Connector for the realm name.

1. Select **Add realm** from the **Select realm** menu.



2. In the **Name** text box, enter the name for the new realm.
3. Click **Create**.

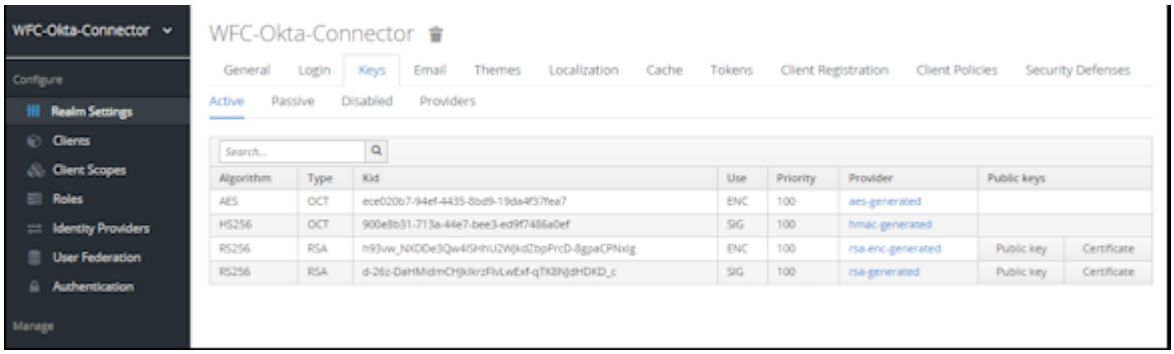


The realm is enabled, and the OpenID endpoint is created.

Retrieving the Signing Certification of the Realm

The signing certification is required to configure the PTT Pro Server.

1. Navigate to **Realm Settings** > **Keys**.



2. Copy the RS256, RSA signing certificate, to a convenient local location on your computer. You use this signing certificate to configure the PTT Pro Server.

NOTE: There are two RSA keys. One is for encoding, ENC, and the other for Signing, SIG. The PTT Pro Server uses the signing certificate.

3. Save the certificate in the X.509 format.

The Realm is established.

Sample Signing Certificate

NOTE: The line length is formatted to fit on the page.

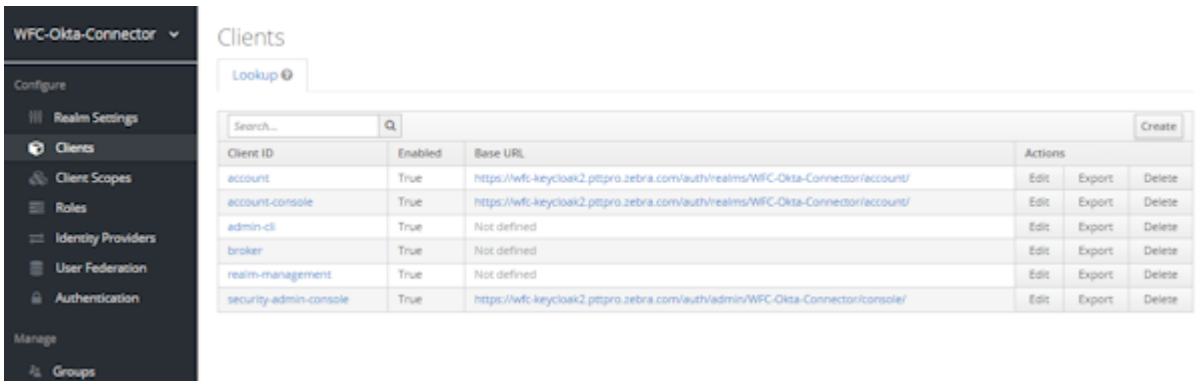
```
MIICszCCAZsCBgGBpcwvCzANBgkqhkiG9w0BAQsFADAdMRswGQ
YDVQQDDBJXRkMtT2t0YS1Db25uZWN0b3IwHhcNMjIwNjI3MTUzM
jEwWhcNMzIwNjI3MTUzMzUwWjAdMRswGQYDVQQDDBJXRkMtT2t0
YS1Db25uZWN0b3IwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggE
KAoIBAQCacby8lfRjEeDXn8VeKaxMBxxuQAkhiC/tnKC6q2MKCW
IHESQqK21HlZ/Pj1HbnDM7GGpBn7zvhQm+aCJ0XjyZiYxy3wkisB
JTqdc6JEzdEXwWxkZ58huenh/PTkpNQy0JLzwCOyzG+iQWWGTY
b/xBuBtY9QJoS2yhSHKU53c8txyytLTqurYNUPTfQC1bcZrQJpe
onAmmH4+FjM5sfOIQ28O7xD1g69IQ1hiW3+y1CsMoLp4F68SdQ+
pmkwH8jmIdLGWmIn6R1FsccnrdLDiydZc84sZcMCKTMnC9PYu3F
GQQzDHsT1goN+QDDfMhCUzbm1zQ4kSXcOkzHoOaG44zAgMBAAE
```

```
wDQYJKoZIhvcNAQELBQADggEBAFvl6+35OlKPsbEWPUPttNJpWf
WGTBpcVGohNmt8e2tfVj0GT7xh4zvNmQvxh+eaewhzuwKhpT/J
G8dyuQVhF4O2Os2W8YPZqvtLWS0cOY9kljqRl1A3z1o2wO1IfDU
+D5aaGSkylyBxL7HkuJsPoWtwUMyfBZNH14Xp4Scwb25BfddECP
SBNCGJ+j4s1rwfac5YVKTswjcePF+r4VsHzEfTgdMhjJha1wI7G
KgzrBXOagZCA6ZfeQMINLTkBSXW6m+xkkcU/owmMXsGJOTEQOTT
0HefiBXq0Jt/0h/NReuc6Qk4AlJHh0Cj9FhAT2OTPvPbn7Yj3vB
7Tne+dMk+p1A=
```

Creating the Client

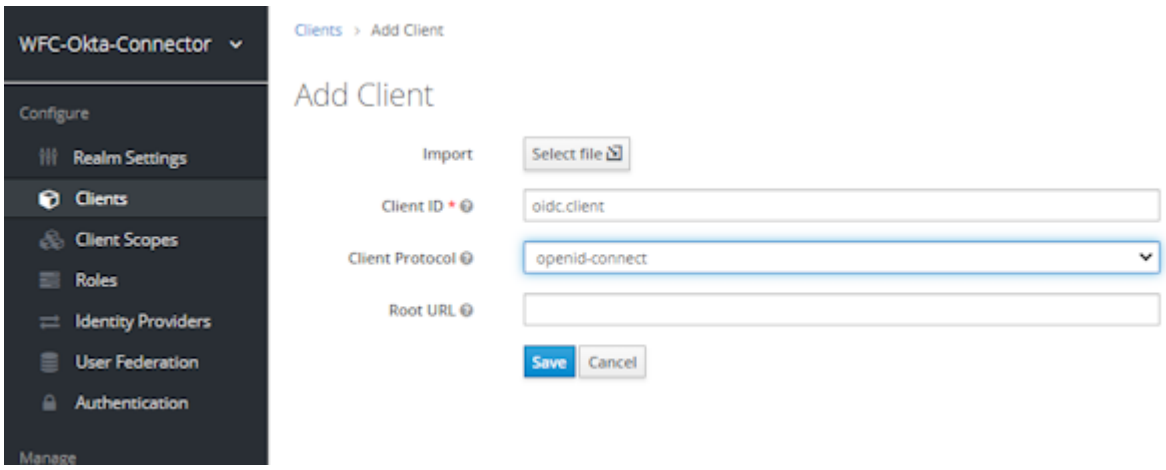
Create a client so that PTT Pro for Android users can authenticate using Okta.

1. Select **Clients** to display the list of default clients.



2. Click **Create**.
3. In the **Client ID** text box, enter a name for the client.

   The client name in this example is oidc.client.



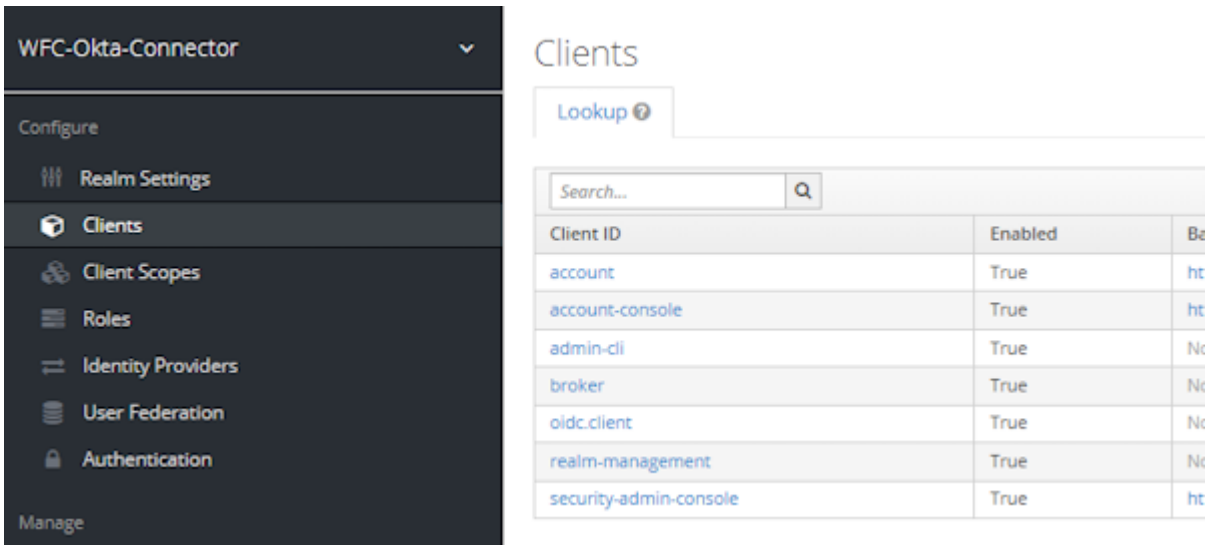
4. Verify that the **Client Protocol** is openid-connect.
5. Click **Save** to proceed and configure the client.

Configuring the Client

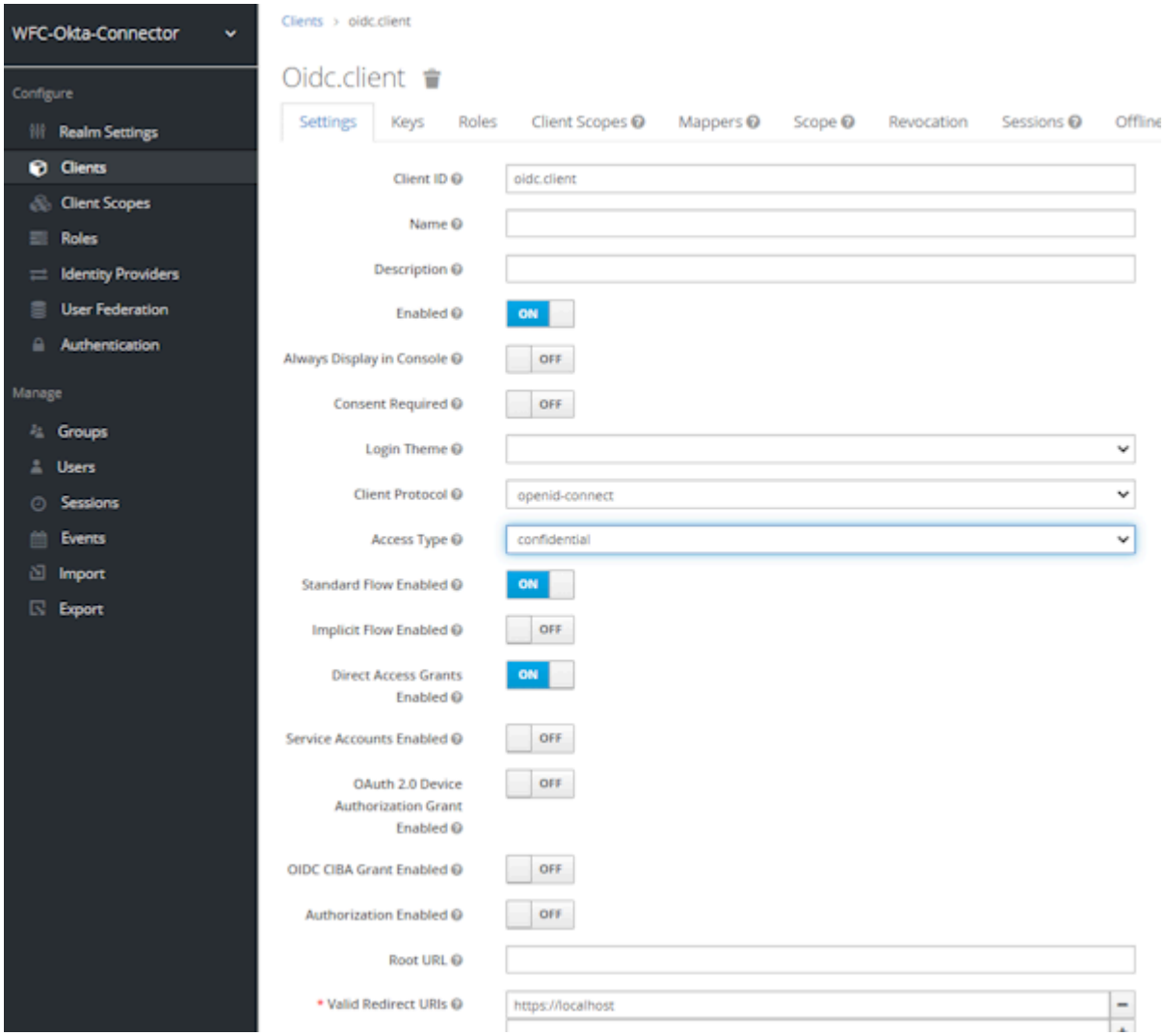
Complete the configuration of the client to ensure the integration with PTT Pro for Android.

1. Click **Clients** under the **Configure** menu.
2. Select the client.

   The client name in this example is oidc.client

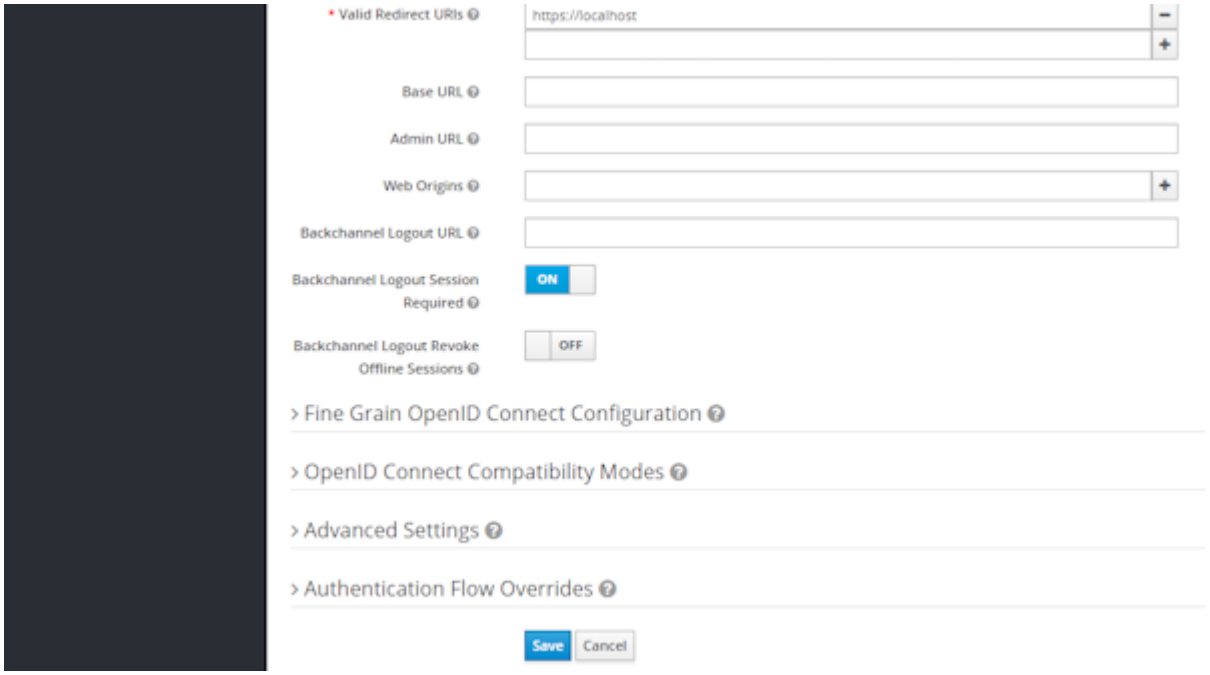


3. Select **Settings** and verify the following settings.



- **Client Protocol** is openid-connect.
- **Access Type** is confidential
  - The access type public works but does not use a client secret.
  - Switching from confidential to public and back to confidential resets the client's secret to a new value.
  - The access type bearer-only does not work.
- **Standard Flow Enabled** is ON. If not selected, PTT Pro for Android displays a blank screen and does not prompt for credentials.
- **Direct Access Grants Enabled** is ON. If not selected, PTT Pro for Android displays a login screen but then displays a blank screen and does not complete the connection.
- Enter `https://localhost` in the **Valid Redirect URIs** field. An invalid URI generates an error on the device when connecting to the ACS; the login screen does not display.

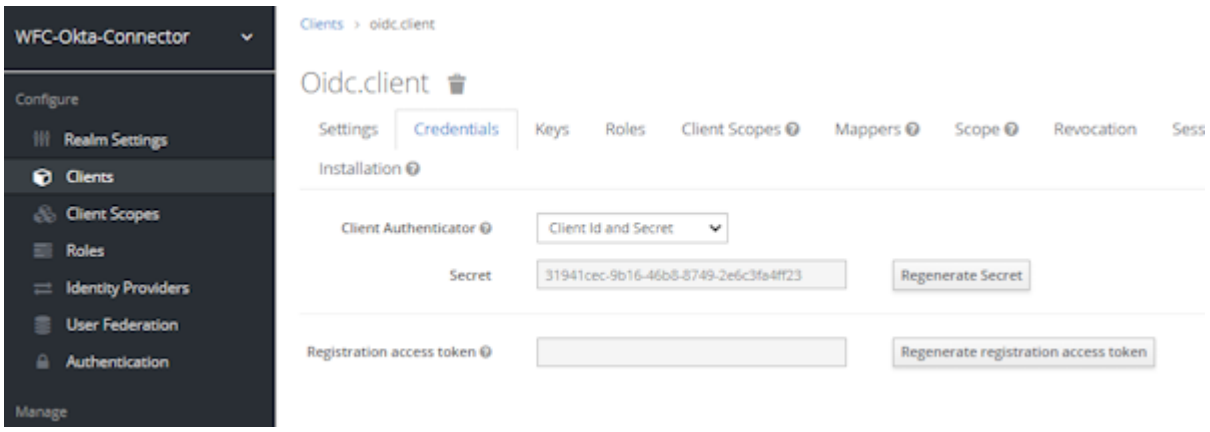
4. Click **Save** to continue.



Copying the Client Secret

The client secret is used to configure PTT Pro for Android. Copy the secret and the Client ID.

1. Click the **Credentials** tab to access the client secret.



2. Copy the client secret.

   The client secret is automatically generated.

3. Save the client secret and the client ID, oidc.client, in this example, to a convenient location.

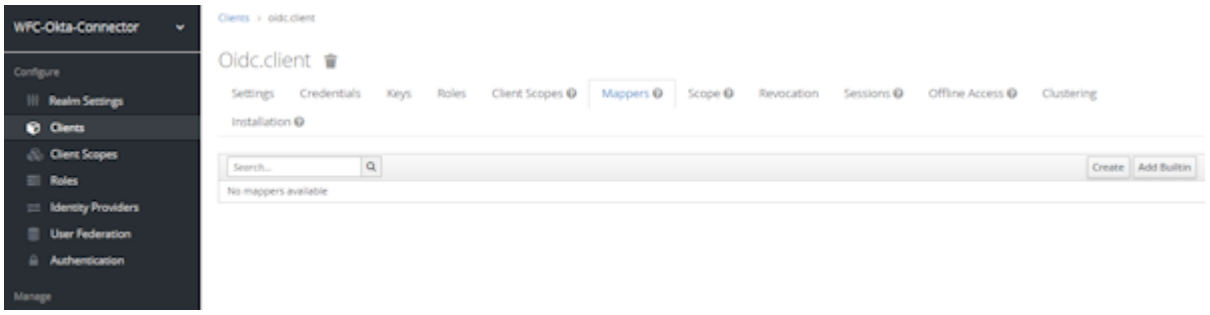
Example Client Secret

```
31941cec-9b16-46b8-8749-2e6c3fa4ff23
```

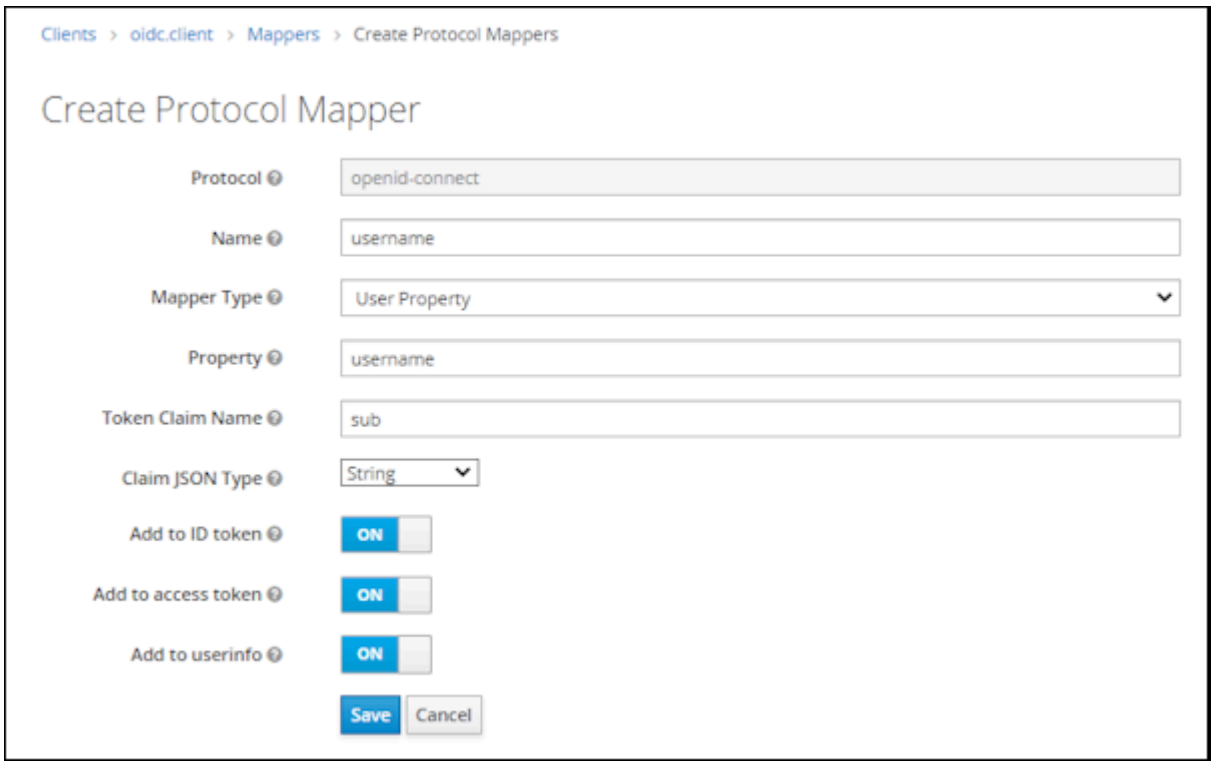
Creating a User Property Mapper

This task maps the token claim `sub` to the `username` attribute used by PTT Pro for Android. If this procedure is not completed, the device user is presented with the credentials screen, but after entering the credentials, the device displays a blank screen.

1. Click the **Mappers** tab in the client definition.



2. Click **Create** to define a mapping.
   - Enter `username` in the **Name** field.
   - Select **User Property** from the **Mapper Type** menu.
   - Enter `username` in the **Property** field.
   - Enter `sub` in the **Token Claim Name** field.
   - Select **String** from the **Claim JSON Type** menu.

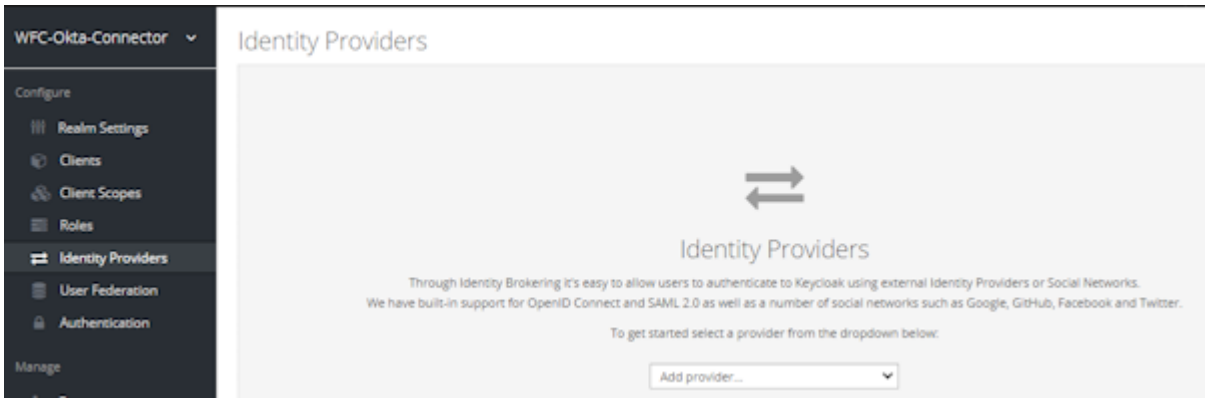


3. Click **Save** to return to the client definition screen.

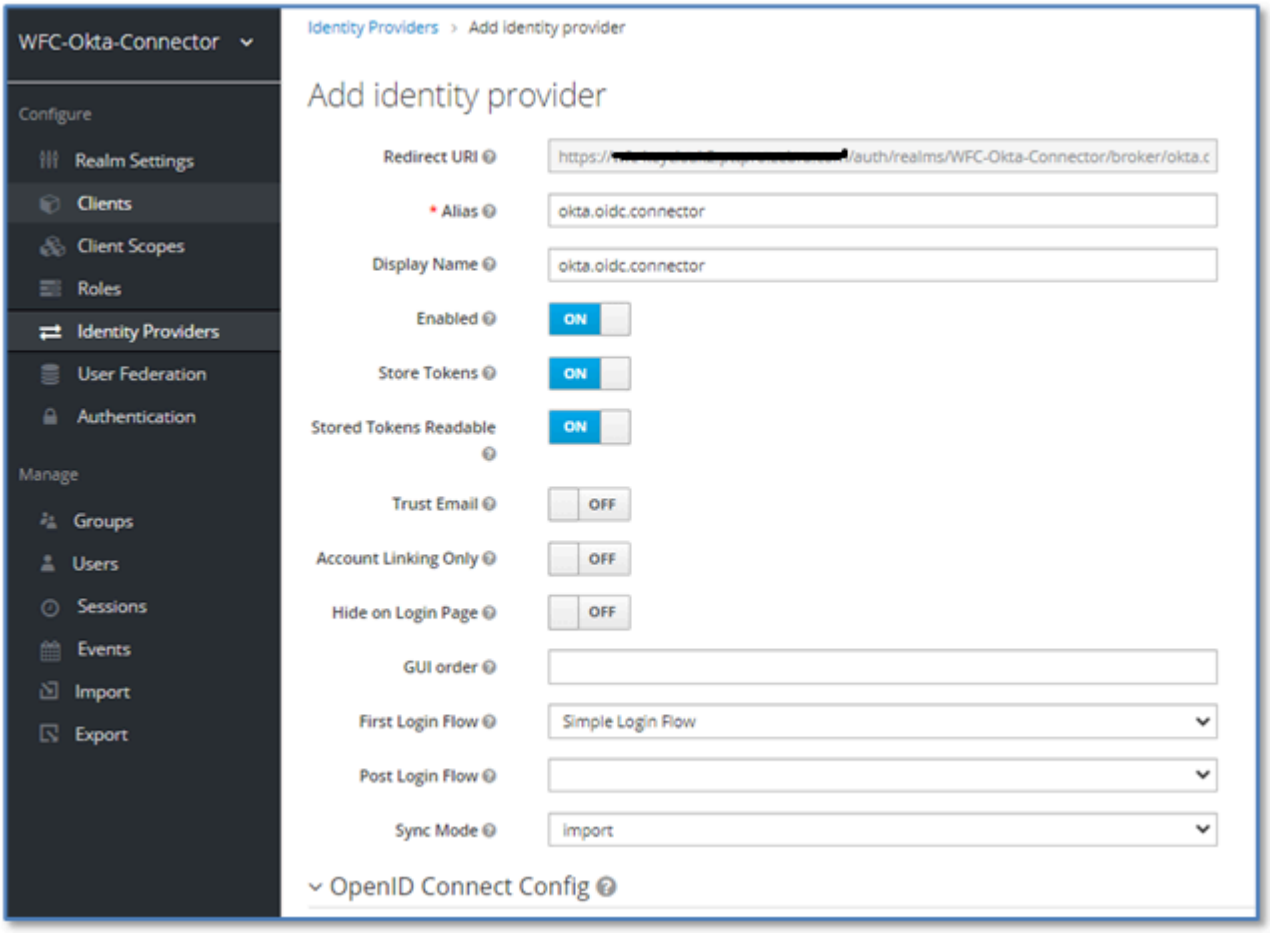
Copying the Identity Provider Redirect URI

Copy the Redirect URI from the ACS to complete the Okta configuration. The ACS IdP requires information from the Okta application configuration, which you complete after the Okta Application is established.

1. Select **Identity Providers** from the client configuration.



2. Select **Open ID Connect 1.0** from the menu to display the redirect URI.
3. Copy the redirect URI from the **Redirect URI** field.



Example Redirect URI

In this example, the redirect URI is: `https://<ACS-server-name>/auth/realm/WFC-Okta-Connector/broker/okta.oidc.connector/endpoint`.

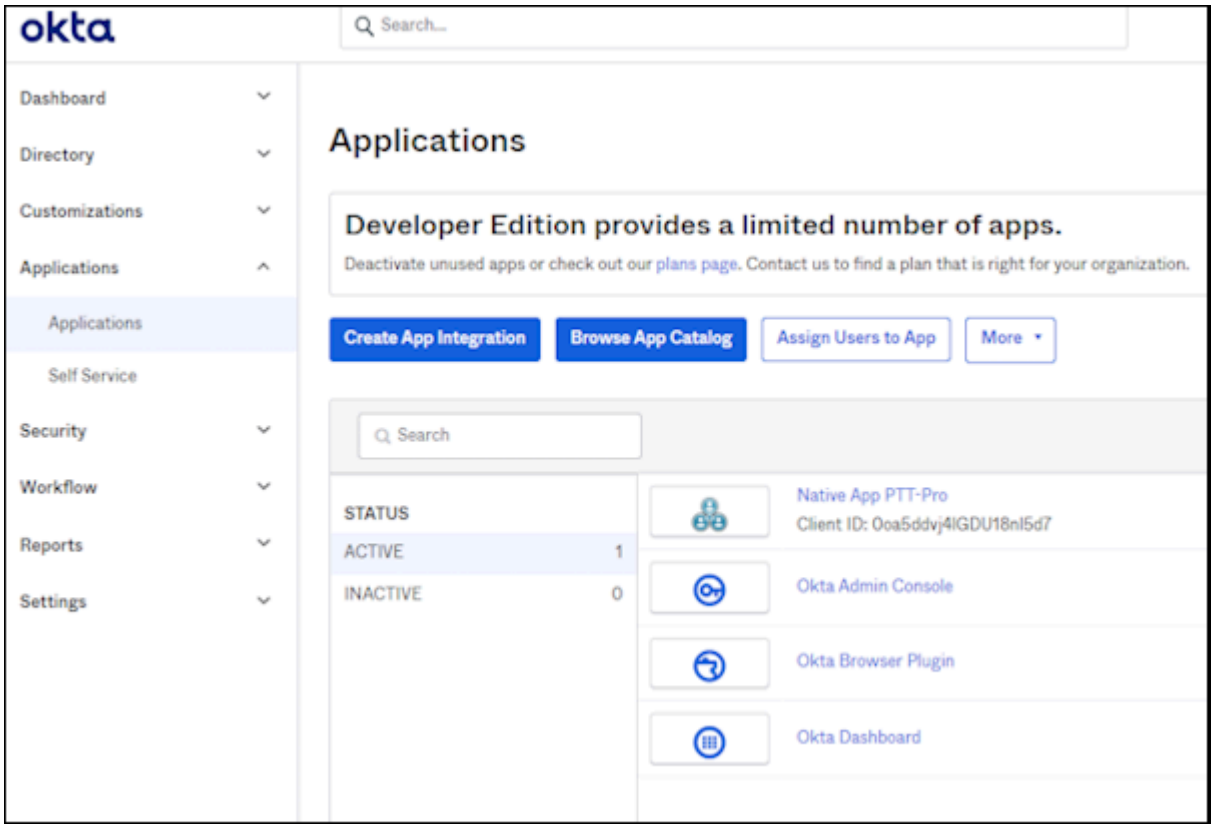
Phase 2 – Configuring Okta

In this phase, you add and configure a new Workcloud Communication native application using configuration elements from the ACS. The users should already exist in the Okta system.

Creating an App

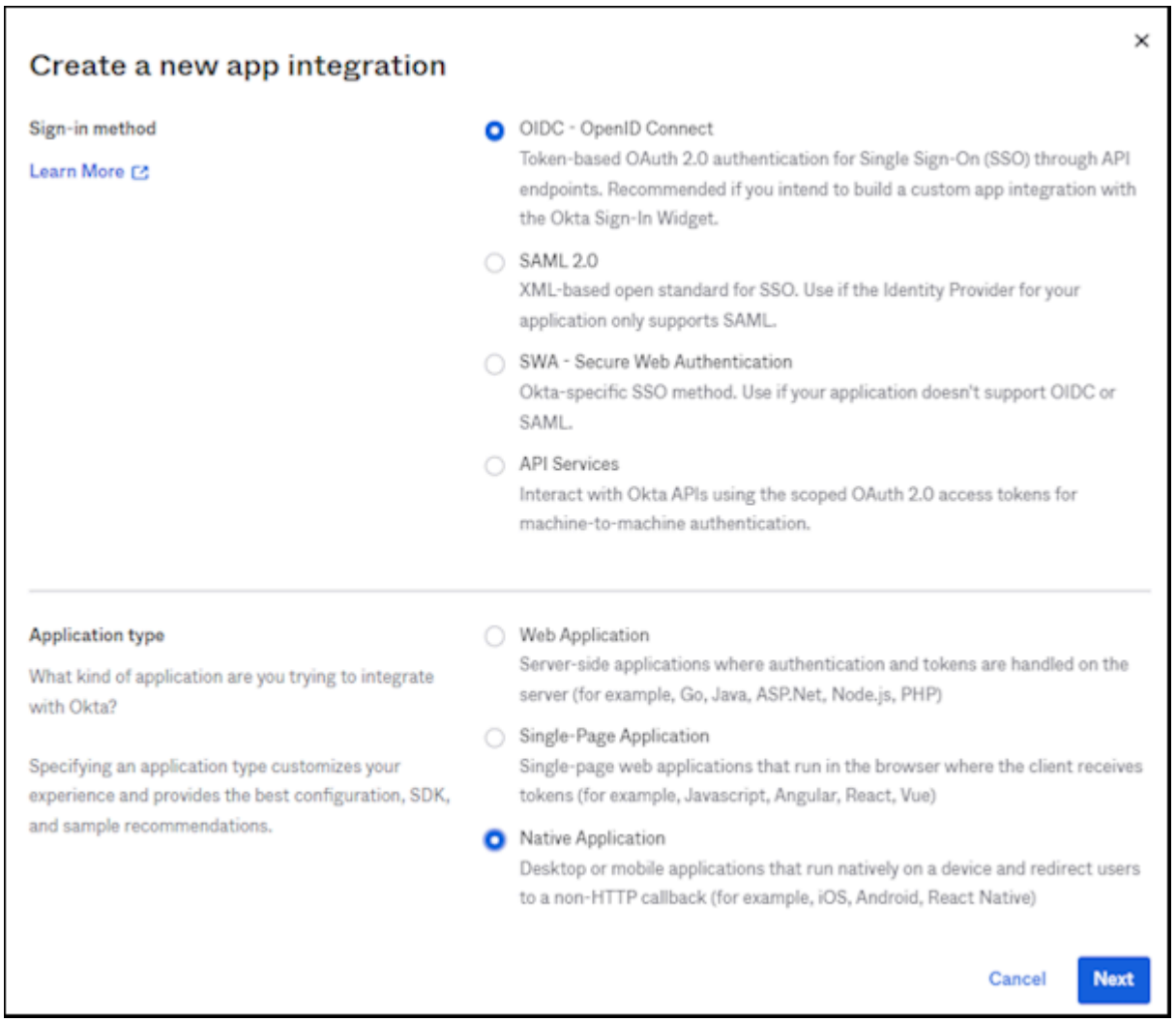
Use Okta to create a new native application for PTT Pro for Android.

1. Open Okta and navigate to **Applications** > **Applications** and click **Create App Integration**.



2. Select **OIDC - OpenID Connect** as the **Sign-in method**.

3. Select **Native Application** as the **Application type**.

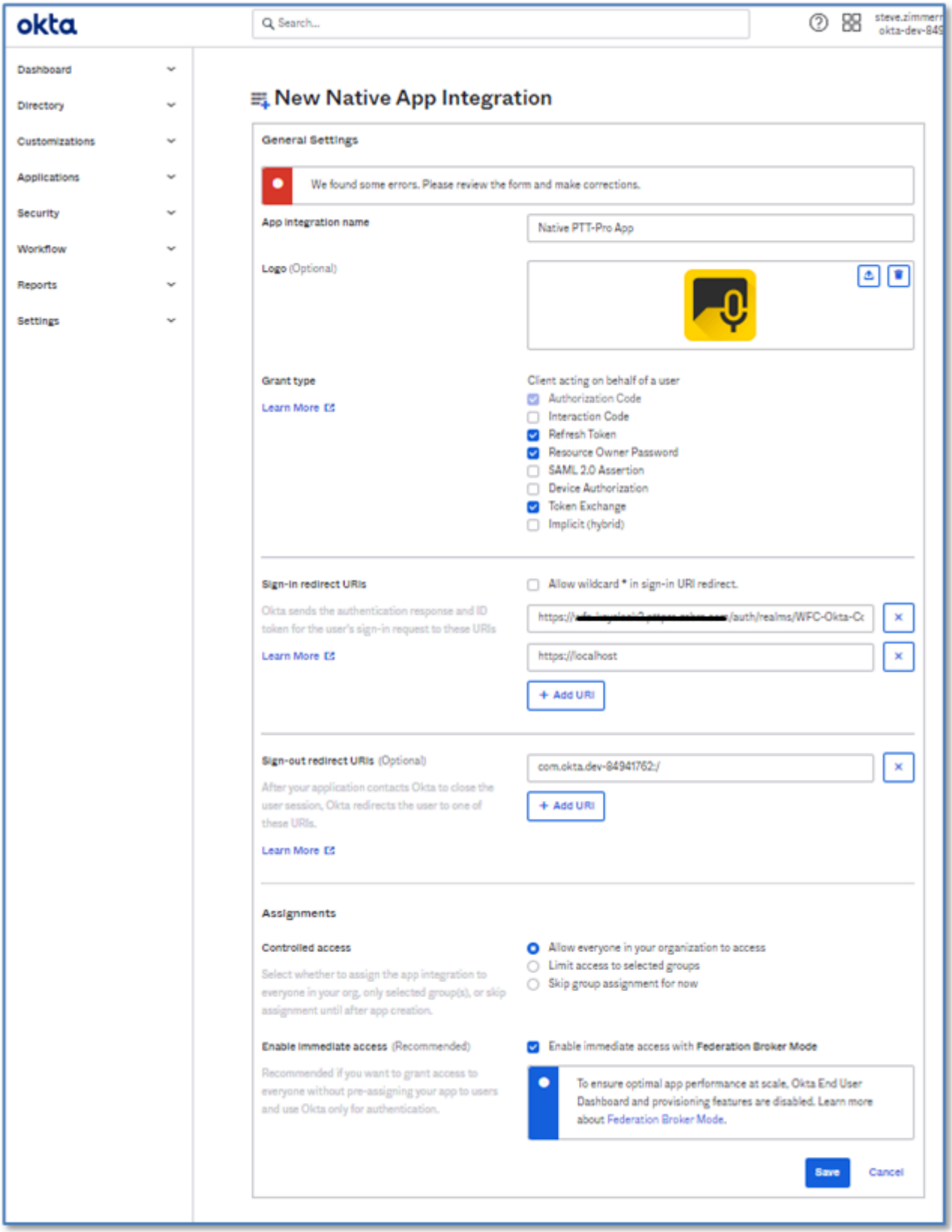


4. Click **Next**.

Configuring the New Native App

Configure the app with the described settings. Settings include naming the app and adding the redirect URI that you previously created.

1. Enter a name in the **App integration name** field.



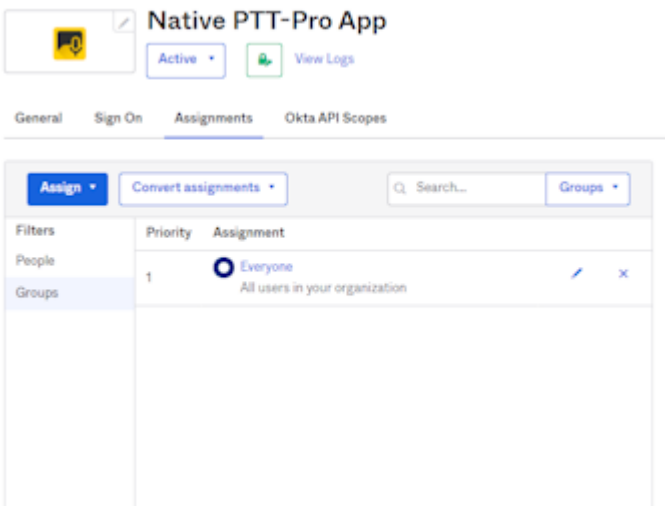
2. Optionally, add an application logo.

3. Select the **Grant type**.
   - Authorization Code is automatically selected.
   - Refresh Token sets the refresh from persistent to a time interval.
   - Additional types can be selected if required.
4. Modify the **Sign-in redirect URIs**.
   a) Delete the default redirect URI.
   b) Enter the Redirect URI from the OIDC Identity Provider. Go to Copying the Identity Provider Redirect URI on page 11.
   c) Click **Add URI** and enter `https://localhost`
5. Do not modify the **Sign-out redirect URIs**.
6. Under **Assignments**, determine how users are granted access to the application.
   - Set Controlled Access to Allow everyone in your organization to access.
   - Choose whether to grant immediate access with the Federation Broker Mode.
     - When enabled, all users are eligible to access the application. Custom access token claims are applied to these users.
     - When disabled, users are selectively granted access and customized claims need to be created and applied to the users. When the broker is not enabled, additional configuration is required. See Configuring Access to the Native App on page 16.
7. Click **Save** to preserve the **General Settings**.

Configuring Access to the Native App

When the Federation Broker is not used to grant users access to the application, complete the following task to determine who can access the app.

1. Open the application and select the **Assignments** tab.



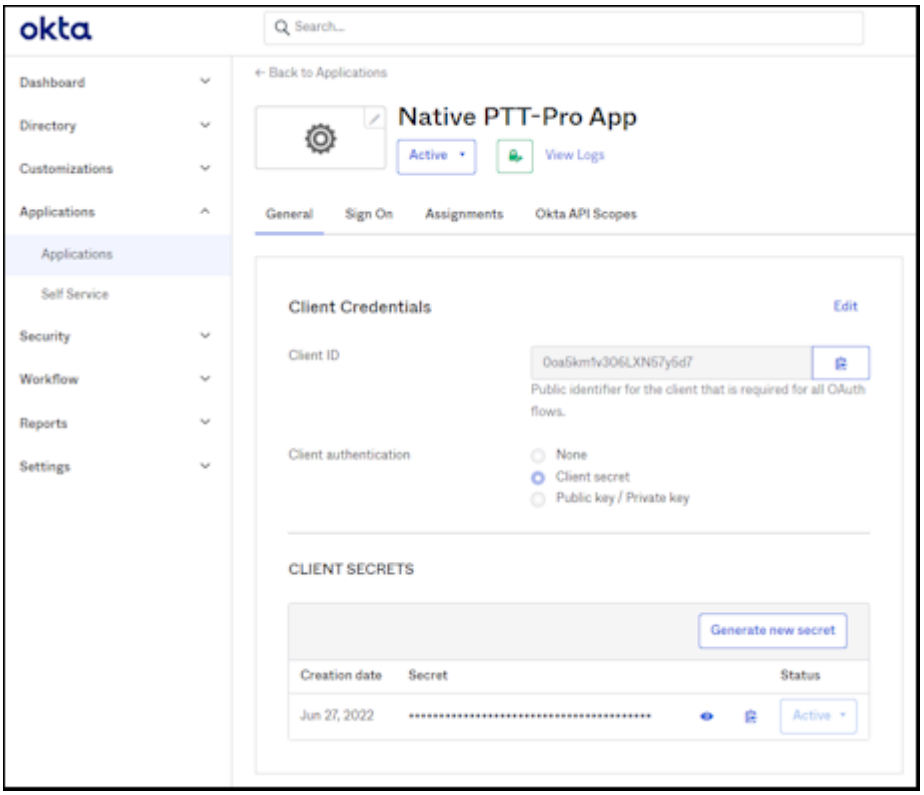
2. Select the access appropriate for your environment.

   In this example, the PTT-Pro App is made available to the group Everyone.

Completing the Native App Configuration

Complete the configuration of the app to finish the ACS Identity Provider.

1. Open the application and perform the following steps under the **General** tab.

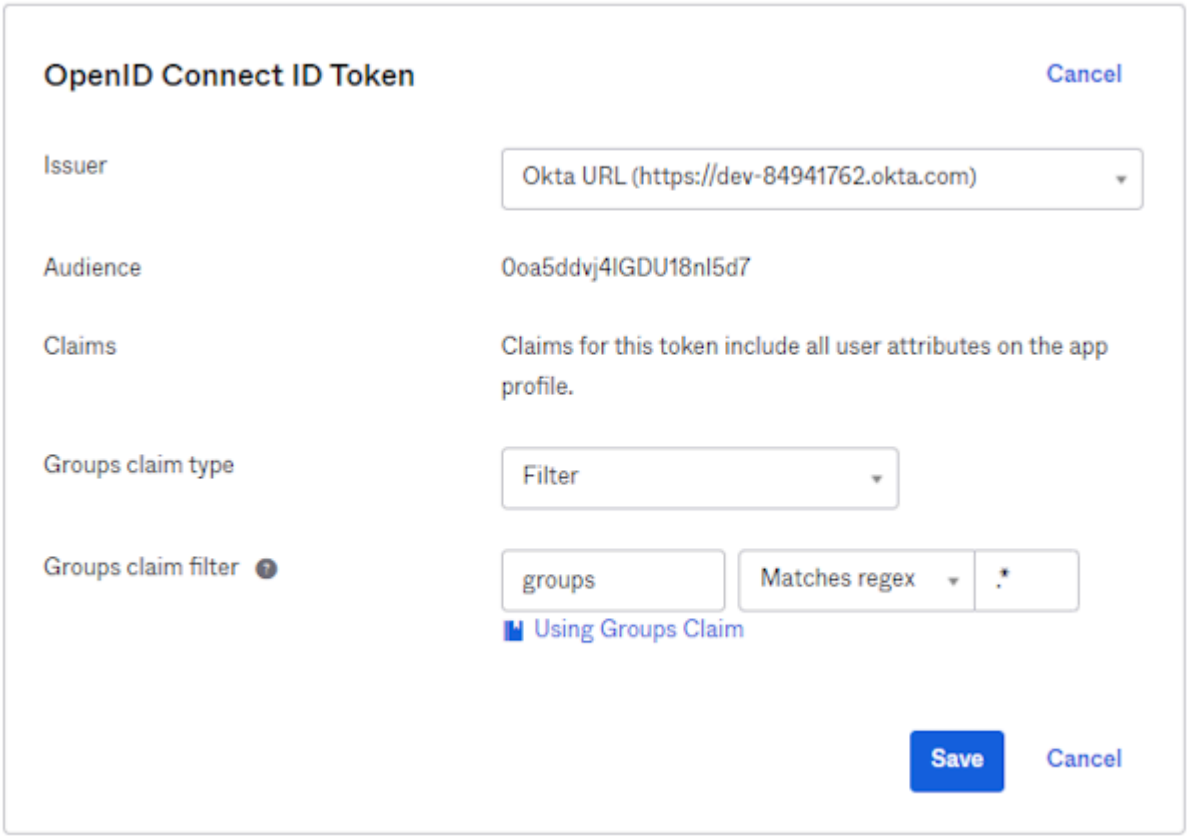


2. Click **Edit**.
3. Copy the client credentials from the **Client ID** field.

   In this example, the client ID is: 0oa5km1v306LXN57y5d7
4. Under **CLIENT SECRETS**, reveal the secret, and copy the secret to your computer for later in the configuration.

   In this example, the client secret is: QUndeR7dkcARwPFjVyGVh6NeFjzbs00Md2xWYFLS

5. Click the **Sign On** tab to edit the **OpenID Connect ID Token**screen.



6. Select Okta URL from the **Issuer** menu.
7. **Audience** is automatically populated with the Client ID.
8. Select **Filter** for the **Group claims type**.
9. Enter the **Group Claims Filter**.

   A typical filter is `.*`.
10. Click **Save**.

Review the remaining configuration options of the native application and make any modifications necessary for your environment.

Phase 3 – Completing the ACS Configuration

You completed the application configuration with the Client ID and Client Secret. In this section, you will complete the configuration of the Identity Provider with information from the Okta configuration.

Creating A Simple Login Authentication Flow

The authentication flow enables the mobile client to seamlessly pass credentials through ACS to the Okta IdP.

Complete the Authentication Flow before completing the IdP definition. The Authentication Flow is not required, and other configurations are possible, but it improves the sign-in experience for the user.

For example, if the ACS IdP configuration specifies:

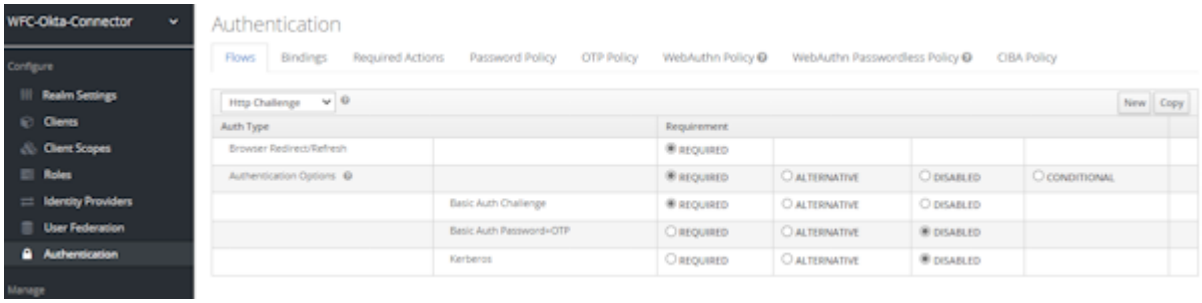
- Browser for the First Login Flow, the user cannot progress past the login screen.
- First Broker Login, the user is prompted to enter their first name, last name, and email address. While this configuration works, it is a poor user experience.
- Direct Grant generates a missing parameter error.
- Registration flow requires that the user enter their user profile information before continuing.

The ACS adds users to the user table as they access the system. The Simple Login Flow automatically populates the user table with the user name when the user signs in. No passwords are examined or tracked. The user authentication occurs on the Okta system

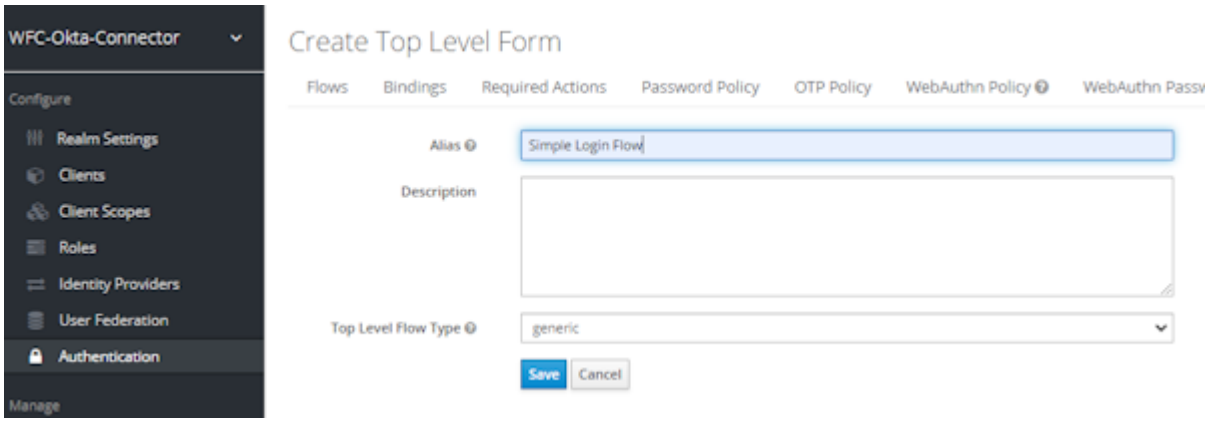
Creating the Authentication Flow

Create an authentication flow to enable the device client to pass credentials through the ACS to the Okta IdP.

1. Select **Authentication** > **Flows** > **New**.

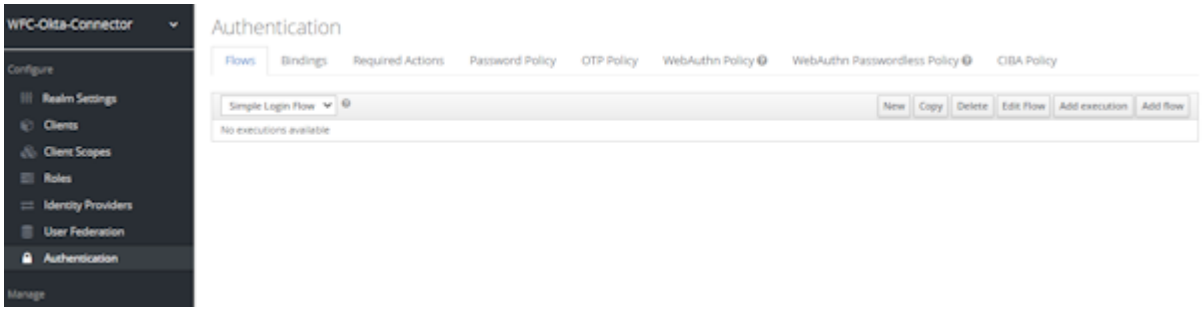


2. Enter a name in the **Alias** field and select generic from the **Top Level Flow Type** menu.

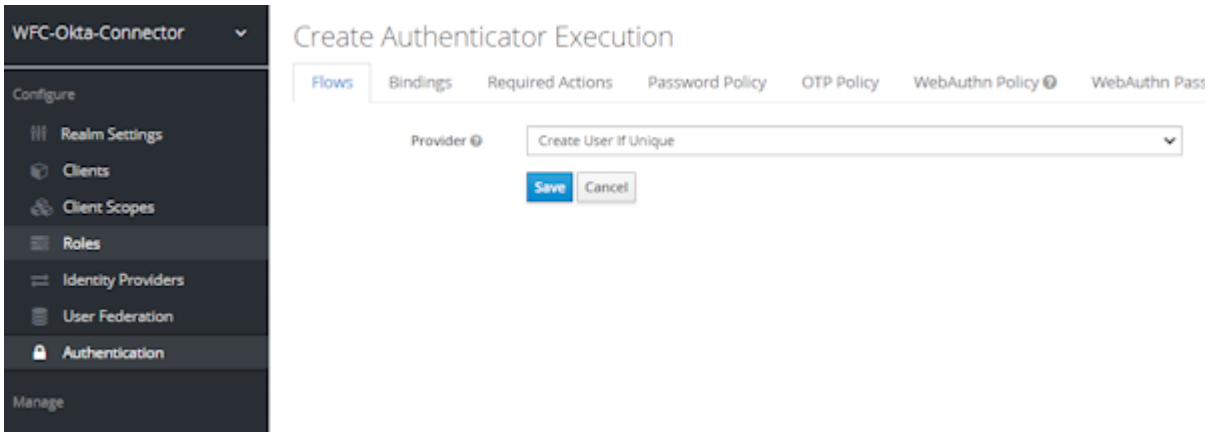


3. Click **Save** to create the flow.

4. Select the new flow from the **Authentication** view and click **Add Execution**.

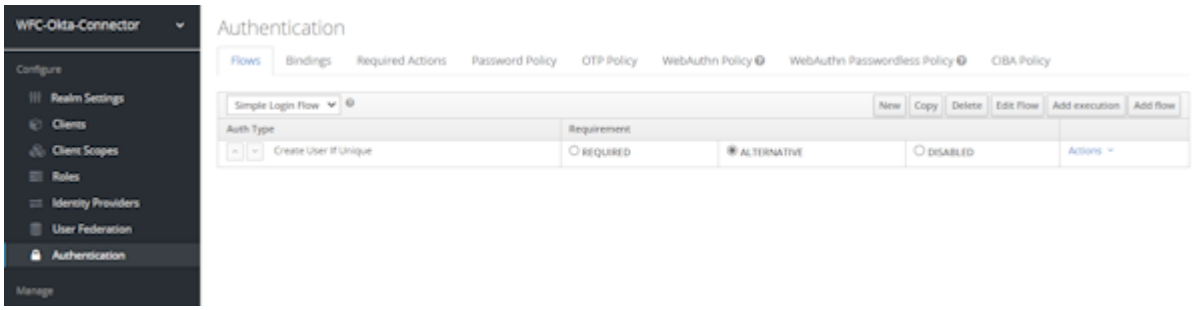


5. Select **Create User If Unique** from the **Provider** menu.



6. Click **Save**.

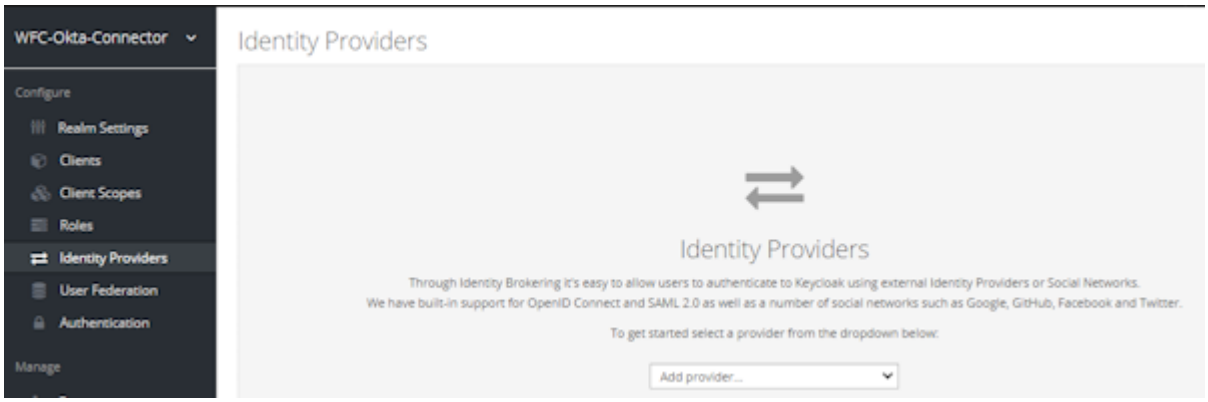
7. In the **Authentication** view, select **Alternative** to activate the flow. The configuration is automatically saved.



Creating the Identity Provider

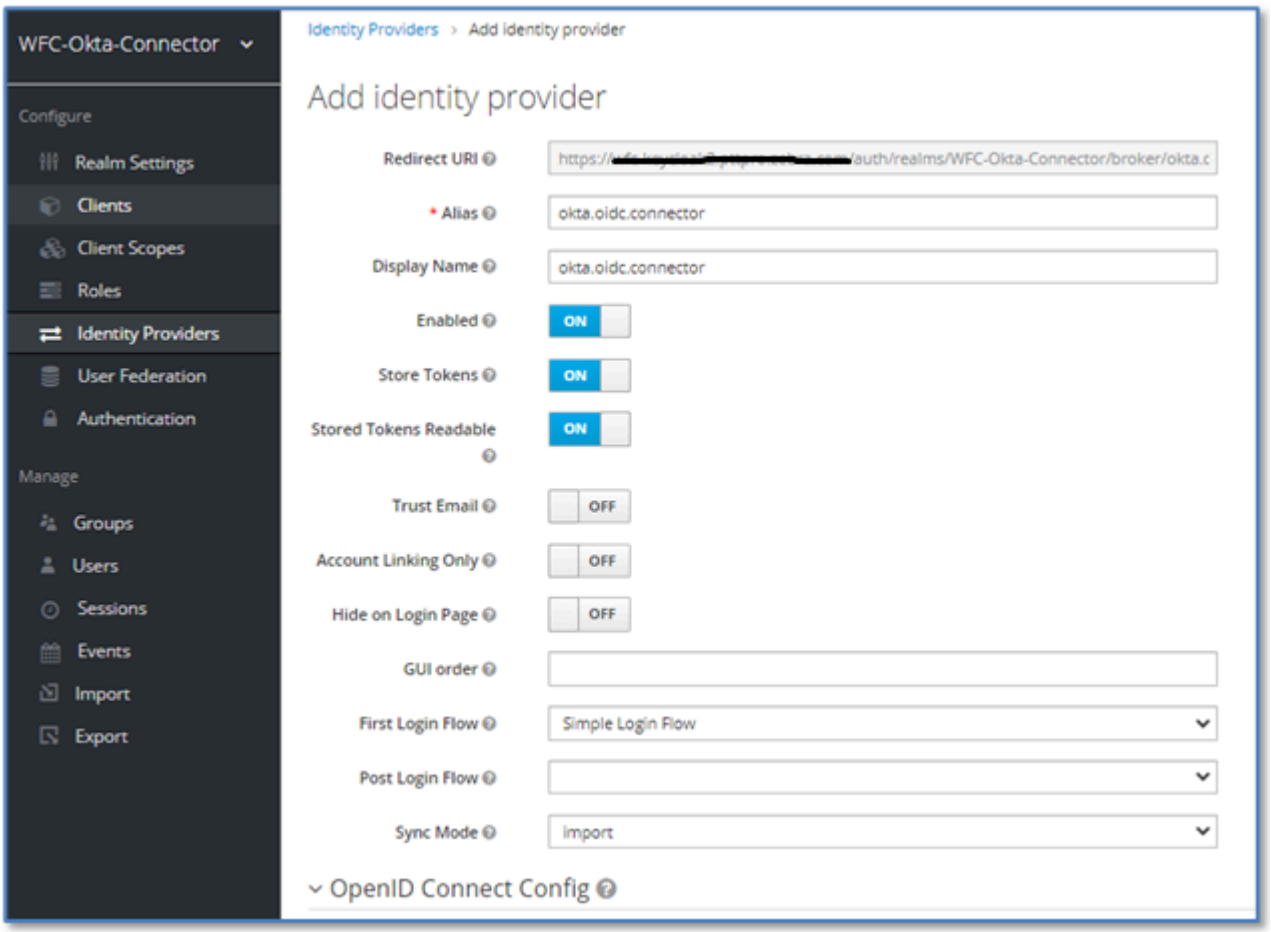
Create the identity provider using information from Okta to complete the configuration.

1. Select **Identity Providers** from the ACS interface.



2. Select **Open ID Connect 1.0** from the menu.

3. Configure the Identity Provider.



- **Alias**, okta.oidc.connector in this example.
- Set **Enabled** to On.
- (Optional) Set **Store Tokens** to On.
- (Optional) Set **Store Tokens Readable** to On.
- Select Simple Login Flow from the **First Login Flow** menu.

The redirect URI is displayed in the **Redirect URI** field and is used as the Okta Redirect for the native app. In this example, the redirect URI is:

```
https://<acs-server-name>/auth/realms/WFC-Okta-Connector/broker/
okta.oidc.connector/endpoint
```

Collecting Information from the Well-Known URL

To complete the configuration of the Identity provider, collect additional information from the well-known URL.

To complete the Identity Provider configuration the following elements are required:

- Authentication URL from the well-known URL
- Token URL from the well-know URL

- Client Authentication
- Client ID
- Client Secret
- Client Assertion Signature Algorithm

You can obtain this information by browsing into the Okta IdP using the well-known URL. The Okta domain and client ID are required to create the well-known URL.

Accessing the Well-Known URL

Use the Okta domain and the client ID to create the well-known URL.

A well-known URL uses the following structure:

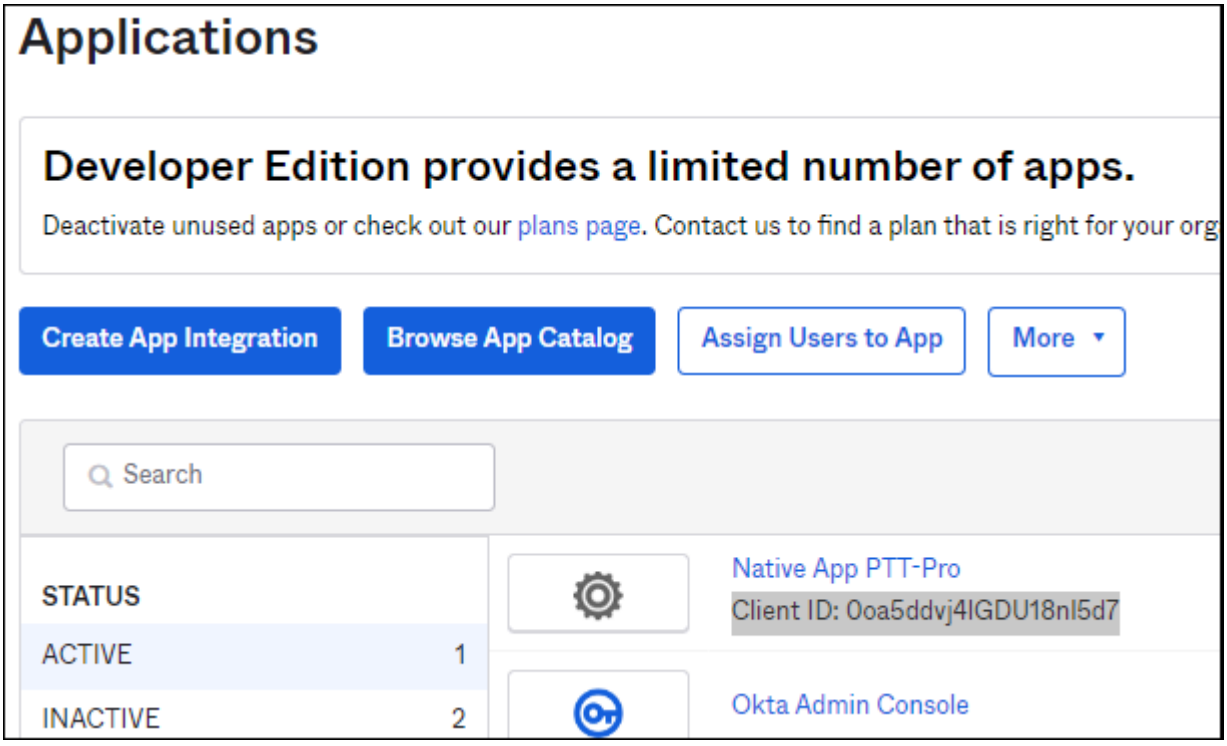
`https://<domain>/.well-known/openid-configuration?client_id=<client-id>`

In this example, the well-known URL becomes:

`https://dev-84941762.okta.com/.well-known/openid-configuration?client_id=0oa5km1v306LXN57y5d7`

- The domain is `dev-84941762.okta.com`.
- The Client ID, `0oa5km1v306LXN57y5d7`, was generated when you created the Native App.

Figure 1 Native App with Client ID



Entering the well-known URL in a web browser returns a response in a JSON file format. The JSON response includes information required for the ACS configuration:

- Authorization endpoint
- Token endpoint

- The supported encrypting format


```
{
"issuer": "https://dev-84941762.okta.com",
"authorization_endpoint": "https://dev-84941762.okta.com/oauth2/v1/
authorize",
"token_endpoint": "https://dev-84941762.okta.com/oauth2/v1/token",
"userinfo_endpoint": "https://dev-84941762.okta.com/oauth2/v1/userinfo",
"registration_endpoint": "https://dev-84941762.okta.com/oauth2/v1/
clients/0oa5km1v306LXN57y5d7",
"jwks_uri": "https://dev-84941762.okta.com/oauth2/v1/keys?
client_id=0oa5km1v306LXN57y5d7",
"response_types_supported": [
"code"
],
"response_modes_supported": [
"query",
"fragment",
"form_post",
"okta_post_message"
],
"grant_types_supported": [
"authorization_code",
"refresh_token",
"password",
"urn:ietf:params:oauth:grant-type:token-exchange"
],
"subject_types_supported": [
"public"
],
"id_token_signing_alg_values_supported": [
"RS256"
],
"scopes_supported": [
"openid",
"email",
"profile",
"address",
"phone",
"offline_access"
],
"token_endpoint_auth_methods_supported": [
"client_secret_basic"
],
"claims_supported": [
"iss",
"ver",
"sub",
"aud",
"iat",
"exp",
"jti",
"auth_time",
```

```
"amr",
"idp",
"nonce",
"name",
"nickname",
"preferred_username",
"given_name",
"middle_name",
"family_name",
"email",
"email_verified",
"profile",
"zoneinfo",
"locale",
"address",
"phone_number",
"picture",
"website",
"gender",
"birthdate",
"updated_at",
"at_hash",
"c_hash"
],
"code_challenge_methods_supported": [
"S256"
],
"introspection_endpoint": "https://dev-84941762.okta.com/oauth2/v1/
introspect",
"introspection_endpoint_auth_methods_supported": [
"client_secret_basic"
],
"revocation_endpoint": "https://dev-84941762.okta.com/oauth2/v1/revoke",
"revocation_endpoint_auth_methods_supported": [
"client_secret_basic"
],
"end_session_endpoint": "https://dev-84941762.okta.com/oauth2/v1/logout",
"request_parameter_supported": true,
"request_object_signing_alg_values_supported": [
"HS256",
"HS384",
"HS512"
],
"device_authorization_endpoint": "https://dev-84941762.okta.com/oauth2/v1/
device/authorize",
"pushed_authorization_request_endpoint": "https://dev-84941762.okta.com/
oauth2/v1/par"
}
```


NOTE: The authorization and token URLs are shown in the response from the well-known URL. If the Okta system uses a Federation Broker, modify the authorization and token URLs.

The original URLs from the JSON response:

- `"authorization_endpoint": "https://dev-84941762.okta.com/oauth2/v1/authorize"`
- `"token_endpoint": "https://dev-84941762.okta.com/oauth2/v1/token"`

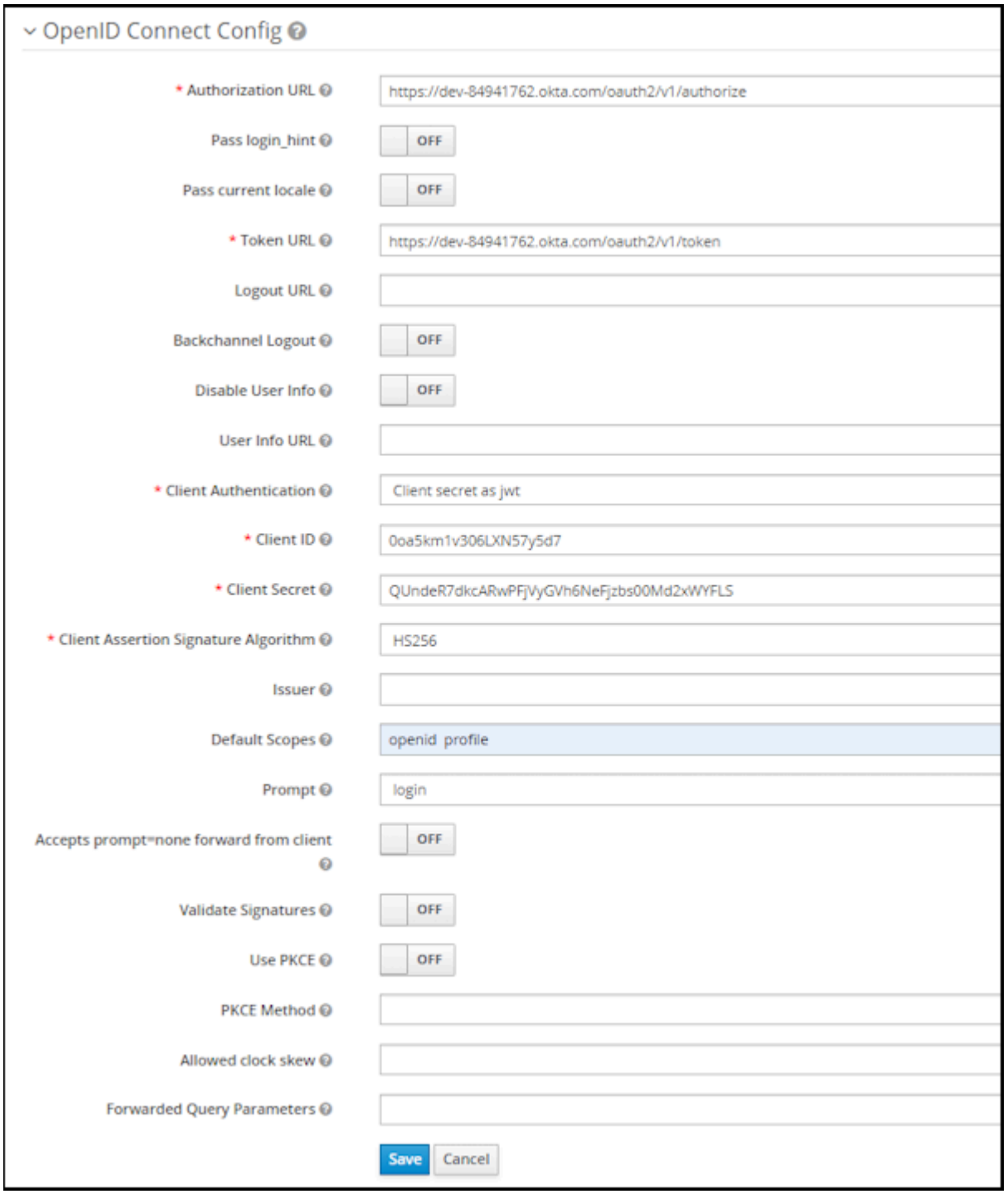
Modify the URLs to include the Authorization Server:

- `"authorization_endpoint": "https://dev-84941762.okta.com/oauth2/<server_name>/v1/authorize"`
- `"token_endpoint": "https://dev-84941762.okta.com/oauth2/<server_name>/v1/token"`

Completing the Identity Provider Configuration

Use the information from the JSON response to the well-known URL to complete the configuration of the Identity Provider.

1. Open the Identify Provider and complete the configuration.



2. The example configuration uses the following data.
   - **Authorization URL** is `https://dev-84941762.okta.com/oauth2/v1/authorize`
   - **Token URL** is `https://dev-84941762.okta.com/oauth2/v1/token`
   - Set **Client Authentication** to `Client secret as jwt`
     - Sent as Post also works
     - basic auth also works
     - JWT Signed with Private Key will fail. The device displays the error, `An Unexpected error when authenticating with the Identity Provider` after entering credentials
   - **Client ID** is `0oa5km1v306LXN57y5d7`

     This value is assigned by the Okta system for the Native App definition. Go to Completing the Native App Configuration on page 17.
   - **Client Secret** is `QUndeR7dkcARwPFjVyGVh6NeFjzbs00Md2xWYFLS`

     This value is assigned by the Okta system for the Native App definition. Go to Completing the Native App Configuration on page 17.
   - **Client Assertion Signature Algorithm** is HS256 (HS384 or HS512)

     The authentication algorithm is specified in the response from the well-known URL. If the algorithm is not supported, the device displays an authentication error after the credentials are entered.
   - Enter `openid`, `offline_access`, and `profile` as space-separated strings in the **Default Scopes** field. These scope values are returned in the response from the well-known URL.
     - If only `openid` is entered, authentication does not advance beyond the credentials screen.
     - If only `offline_access` is entered, authentication does not advance beyond the credentials screen.
     - If only `profile` is entered, the device displays a web page not available error.
   - Enter `login` in the **Prompt** field.
     - The `Consent` and `Select_Account` parameters also work.
     - Unspecified results in a blank screen on the device after the credentials are entered.
     - Non results in a blank screen.
3. Click **Save**.

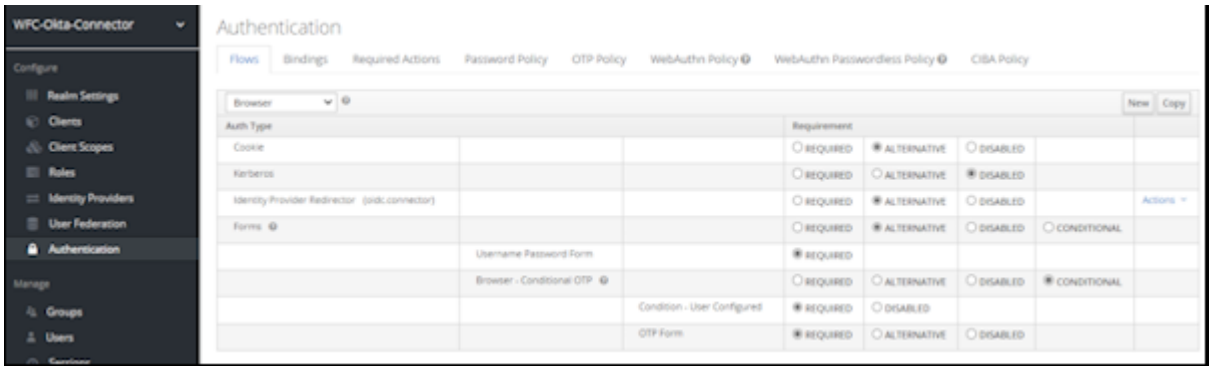
Setting the Login Browser Flow

After you configure the Identity Provider, you can complete the configuration of the ACS Identity Provider.

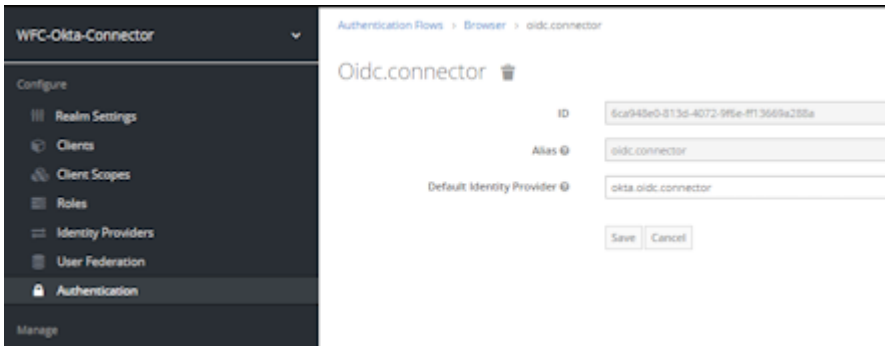
The Identity Provider Redirector should be set to the Identity Provider to enable the Okta login screen. If this step is not performed, the user is presented with the ACS login screen.

NOTE: A user could click on the **okta.oidc.connector** to log in, but this can be confusing in a runtime environment.

1. Select **Authentication** and click the **Flows** tab to list all of the authentication types.



2. Click the **Actions** menu on the right side of the screen and select **Configuration** for the Identity Provider Redirector.
3. Configure the oidc.connector.



- Enter an **Alias**. In this example, oidc.connector.
- Enter the **Default Identity Provider**. In this example, okta.oidc.connector.

4. Click **Save**.

Validate the Configuration

Validate the configuration before testing with a mobile device. Two tools are used to validate the configuration: the Postman application and the jwt.io token validator.

Validating with Postman

Testing access with Postman validates that the collected information is accurate and produces the expected results.

Create a profile with the following configuration parameters:

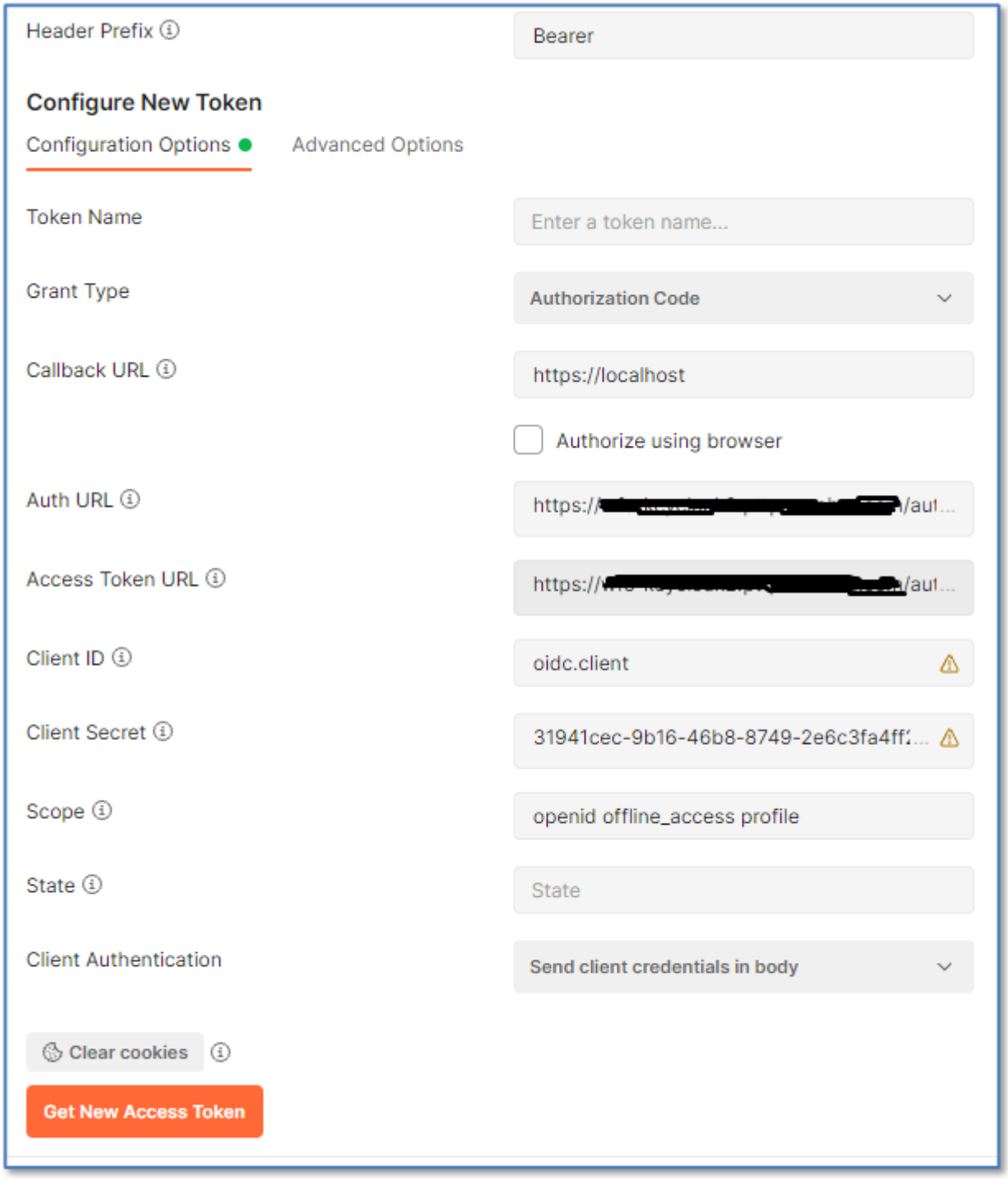
- Auth URL
- Access Token URL
- Client ID
- Client Secret
- Scope with the space-separated values of `openid`, `profile`, and `offline_access`

  The profile scope is used to get a refresh token.

1. Insert the following URLs into the Postman profile along with the Client ID and Client Secret.

   The URLs can be found by entering the well-known URL into a browser. In this example, the well-known URL is `https://<acs-server-name>/auth/realms/WFC-Okta-Connector/.well-known/openid-configuration`

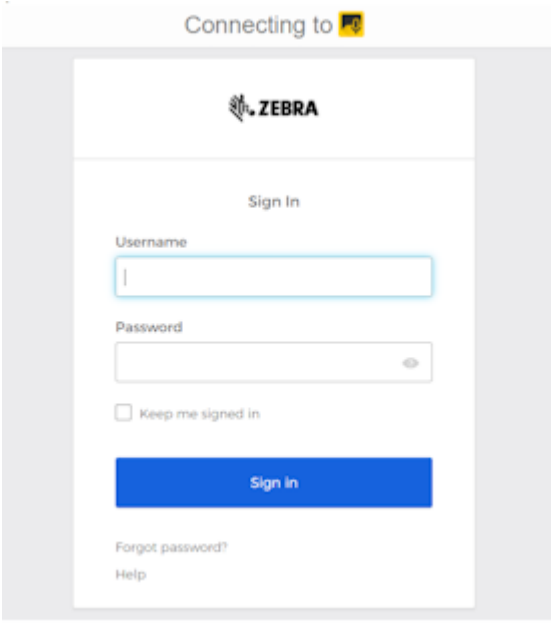
   - Authorization endpoint URL: `https://<acs-server-name>/auth/realms/WFC-Okta-Connector/protocol/openid-connect/auth`
   - Access Token URL: `https://<acs-server-name>/auth/realms/WFC-Okta-Connector/protocol/openid-connect/token`



Remember to clear the cookies before running the test again.

2. Click **Get New Access Token**.

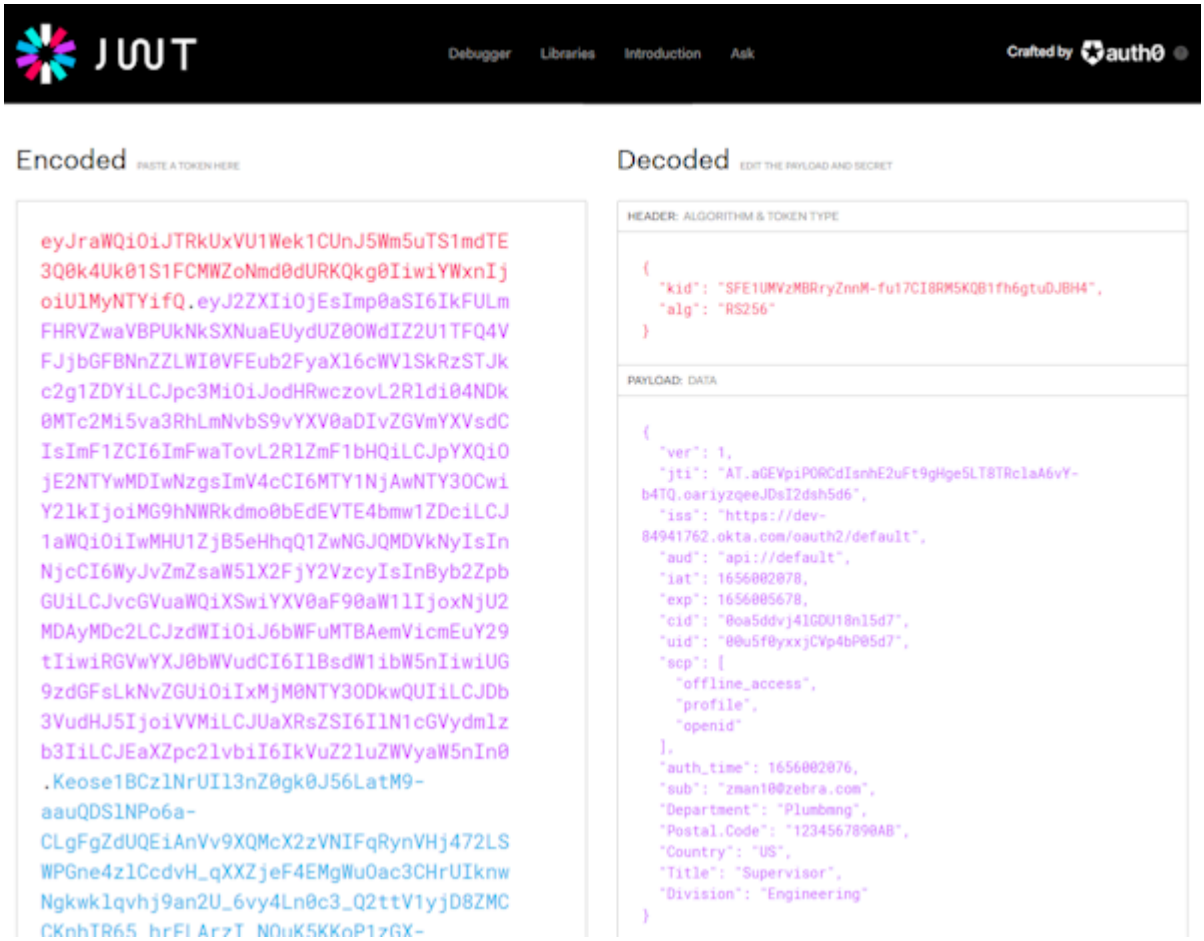
   A successful test displays the Okta login screen from the ACS.



NOTE: The branding icons in the example are configured separately and outside the scope of this document.

3. Enter user credentials.

4. Analyze the access token by entering it into the https://jwt.io/ token analysis site.



5. Review the decoded payload data and verify the user identification. In this example, the `sub` claim is the UserID used by Workcloud Communication.

NOTE: Additional claims are shown in this example. You can provide custom access token claims by configuring and mapping attributes in the Okta Directory / Profile Editor. This is not covered in this document.

Configuring the Workcloud Communication System

After validating the basic Okta and ACS configuration, configure the PTT Pro Server. PTT Pro on mobile devices first connects to the PTT Pro Server and uses the URLs to connect to the ACS server, which redirects the user to the Okta system.

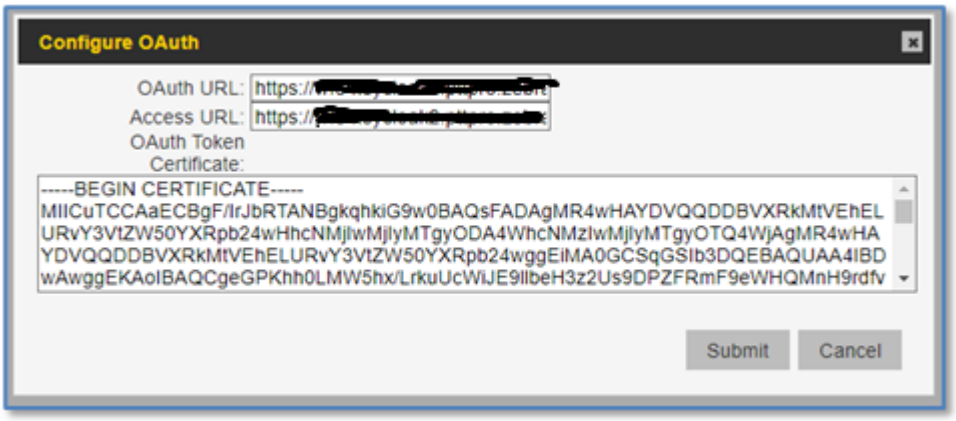
The PTT Pro Server requires three configuration parameters:

- Access URL
- Token URL
- Signing certificate

1. Log in to the Workcloud Communication PTT Pro Management Portal and navigate to **Customer** > **Profile** to configure OAuth.

2. Enter the Access URL and Authorization Endpoint URL (OAuth URL) in the **Configure OAuth** dialog.
   - The Authorization Endpoint URL is the **Oauth URL** field: `https://[acs-server-name]/auth/realms/WFC-Okta-Connector/protocol/openid-connect/auth`
   - The Access URL is the Access URL field: `https://[acs-server-name]/auth/realms/WFC-Okta-Connector/protocol/openid-connect/token`

   These are the same URLs used to validate the configuration with Postman and obtained from the JSON output of the well-known URL.



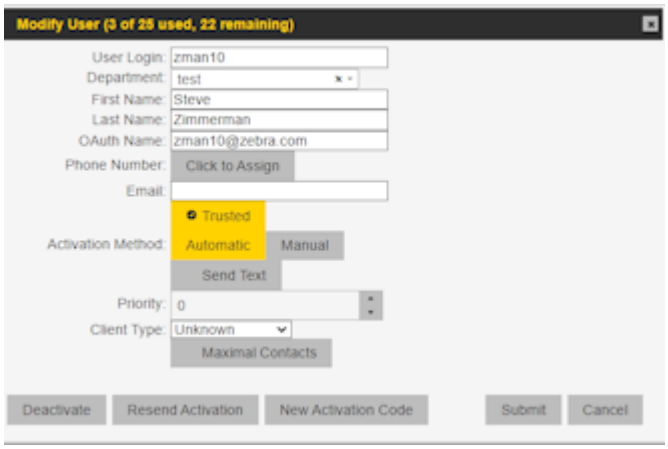
3. Enter the OAuth Token Certificate. You copied the certificate previously in Retrieving the Signing Certification of the Realm on page 5.

   Copy the certificate and paste it into a text editor such as Notepad++. Add Begin Certificate and End Certificate as shown in the example.

```
-----BEGIN CERTIFICATE-----
MIICszCCAZsCBgGBpcwvCzANBgkqhkiG9w0BAQsFADAdMRswGQYDVQQDDBJXRkMt
T2t0YS1Db25uZWN0b3IwHhcNMjIwNjI3MTUzMjEwWhcNMzIwNjI3MTUzMzUwWjAd
MRswGQYDVQQDDBJXRkMtT2t0YS1Db25uZWN0b3IwggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQCacby8lfRjEeDXn8VeKaxMBxxuQAkhiC/tnKC6q2MKCWIHES
QqK21HlZ/Pj1HbnDM7GGpBn7zvhQm+aCJ0XjyZiYxy3wkisBJTqdc6JEzdEXwWxkZ
58huenh/PTkpNQy0JLzwCOyzG+iQWWGTYb/xBuBtY9QJoS2yhSHKU53c8txyytLTq
urYNUPTfQC1bcZrQJpeonAmmH4+FjM5sfOIQ28O7xD1g69IQ1hiW3+y1CsMoLp4
F68SdQ+pmkwH8jmIdLGWmIn6R1FsccnrdLDiydZc84sZcMCKTMnC9PYu3FGQQzDH
sT1goN+QDDfMhCUzbm1zQ4kSXcOkzHoOaG44zAgMBAAEwDQYJKoZIhvcNAQELBQAD
ggEBAFvl6+35OlKPsbEWPUPttNJpWfWGTBpcVGohNmt8e2tfVj0GT7xh4zvNmQv
xh+eaewhzuwKhpT/JG8dyuQVhF4O2Os2W8YPZqvtLWS0cOY9kljqRl1A3z1o2wO
1IfDU+D5aaGSkylyBxL7HkuJsPoWtwUMyfBZNH14Xp4Scwb25BfddECPSBNCGJ+j
4s1rwfac5YVKTswjcePF+r4VsHzEfTgdMhjJha1wI7GKgzrBXOagZCA6ZfeQMINL
TkBSXW6m+xkkcU/owmMXsGJOTEQOTT0HefiBXq0Jt/0h/NReuc6Qk4AlJHh0Cj9
FhAT2OTPvPbn7Yj3vB7Tne+dMk+p1A=
-----END CERTIFICATE-----
```

NOTE: Do not add or remove any characters from the certificate because it will cause the authentication to fail.

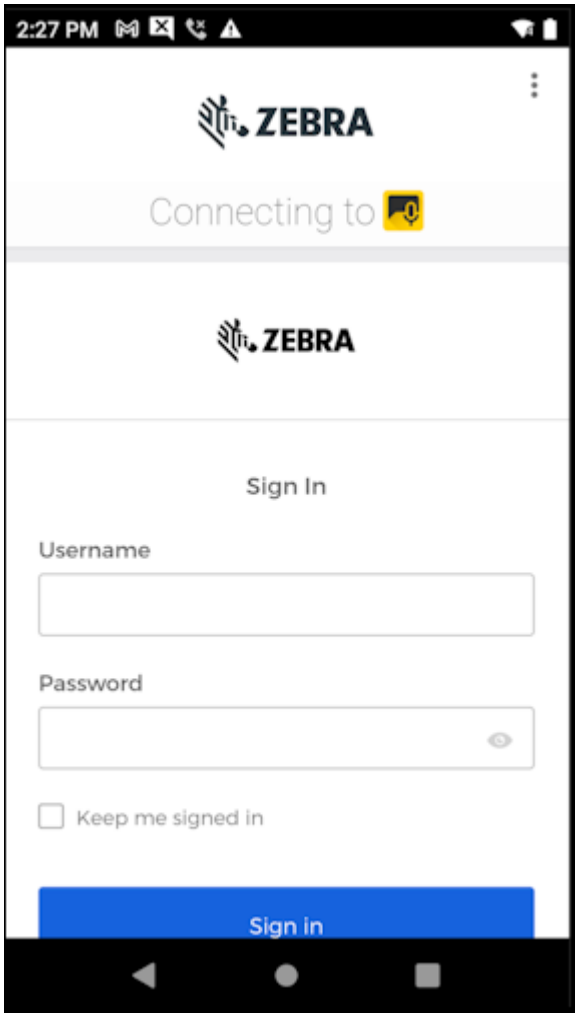
4. Enter the device serial number in the PTT Pro Server and create OAuth user accounts with an accurate OAuth name.



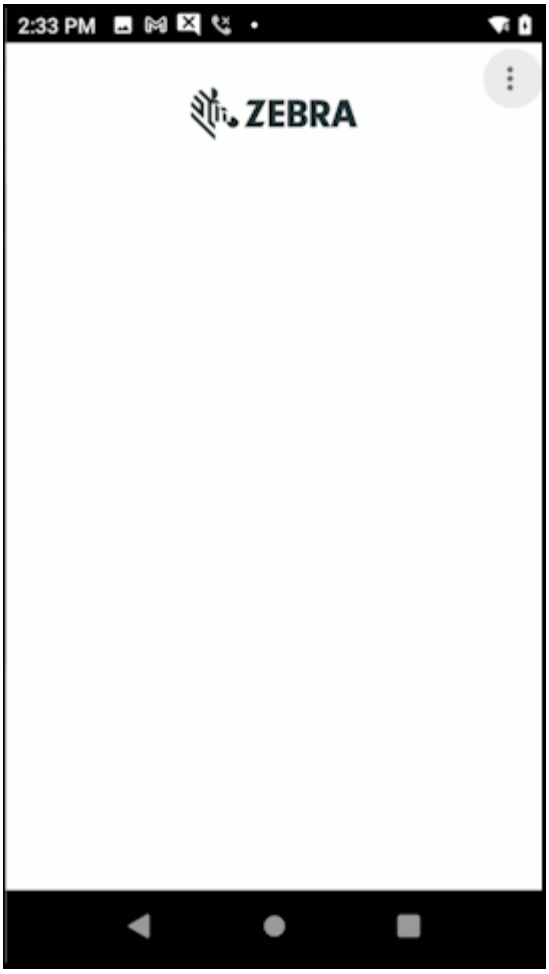
5. Add the JSON configuration to the mobile devices so that the PTT Pro client connects to the ACS server.

```
{
    "oAuthClientID" : "oidc.client",
    "oAuthClientSecret" : "31941cec-9b16-46b8-8749-2e6c3fa4ff23",
    "oAuthBasicHeader" : true
}
```

6. Use a mobile device to verify the configuration. The Okta sign-on screen should display.



If the device displays a blank screen after entering the credentials, potential causes include a certificate with unprintable characters or white space. Another potential cause is that the Authorization Endpoint URL or Access URL is not correct.



Revision History

Revision	Date	Description
MN-004831-01EN	January, 2024	First version.