

Radiator

CONFIGURATION GUIDE

Okta Application Setup

Date: 2026-11-16

This configuration allows authenticating RADIUS users from Radiator Policy Server against Okta using the OAuth2 Resource Owner Password Credentials (ROPC) flow. Radiator exchanges the user's credentials for an access token via Okta's Authorization server, then queries the `/userinfo` endpoint to retrieve the user's profile and group memberships.

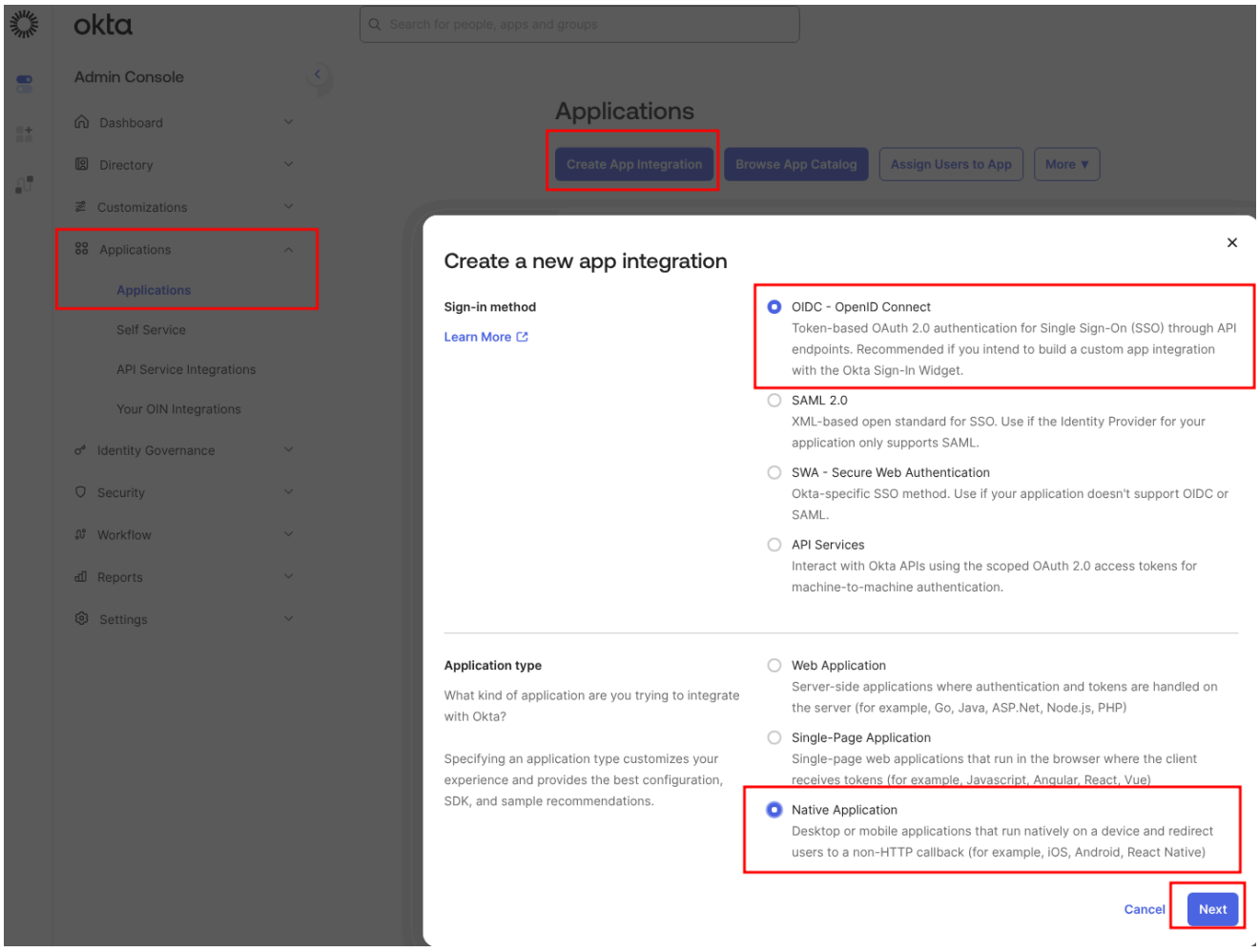
NOTE Using ROPC and adding a custom claim to Authorization server requires the Okta API Access Management add-on.

Prerequisites

- An Okta tenant
- User accounts, optionally belonging to groups
- Admin access to the Okta Admin Console

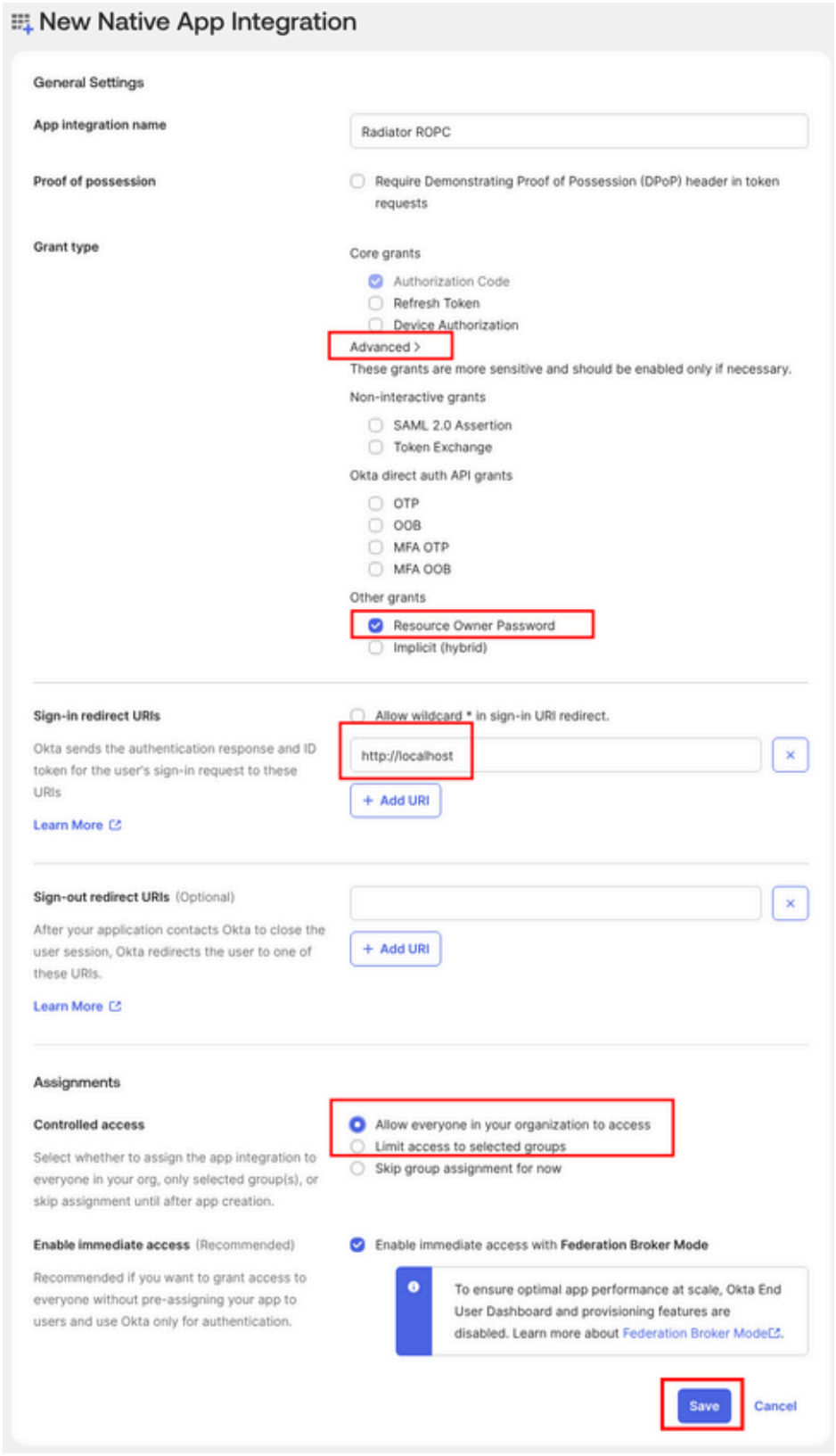
Create an OIDC Application

1. Go to **Applications > Applications > Create App Integration**
2. Select **OIDC - OpenID Connect**
3. Select **Native Application** as application type

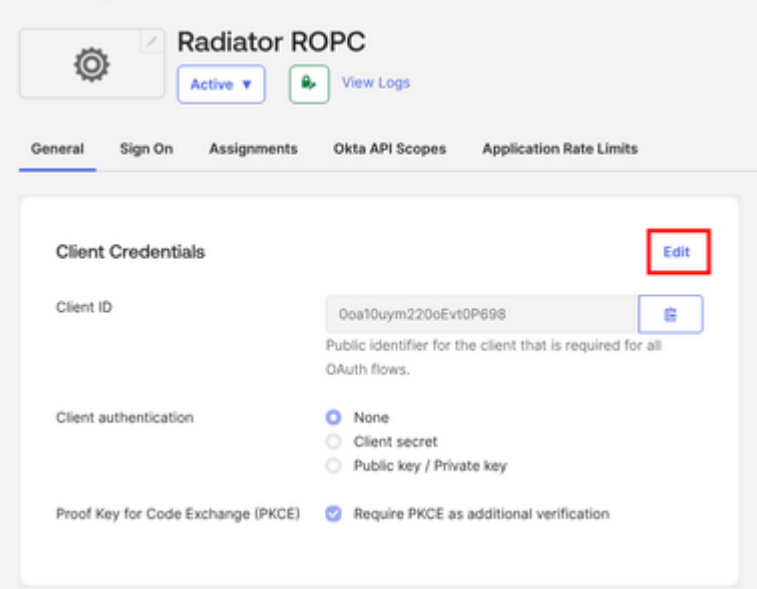


4. Configure the application:
    - App integration name: e.g., `Radiator ROPC`
    - Grant type: Check **Resource Owner Password** (under "Advanced" > "Other grants")
    - Sign-in redirect URIs: `http://localhost` (required, but not used by Radiator)
    - Assignments: Either **Allow everyone in your organization to access** or **Limit access to selected groups** depending on your needs

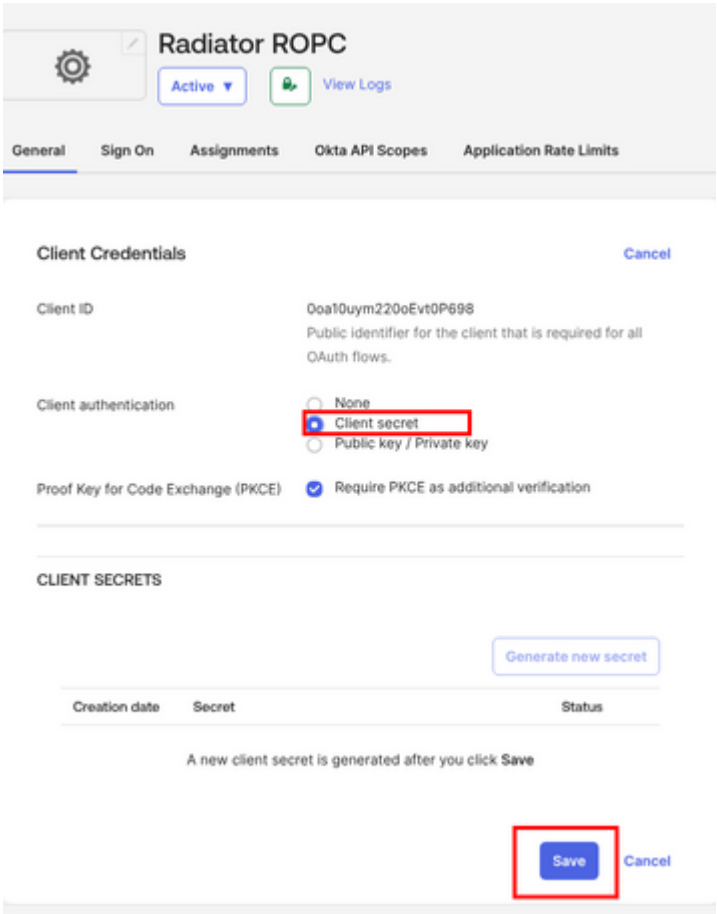
5. Click **Save**


New Native App Integration
General Settings
App integration name
Radiator ROPC
Proof of possession
Require Demonstrating Proof of Possession (DPoP) header in token requests
Grant type
Core grants
Authorization Code
Refresh Token
Device Authorization
Advanced >
These grants are more sensitive and should be enabled only if necessary.
Non-interactive grants
SAML 2.0 Assertion
Token Exchange
Okta direct auth API grants
OTP
OOB
MFA OTP
MFA OOB
Other grants
Resource Owner Password
Implicit (hybrid)
Sign-in redirect URIs
Allow wildcard * in sign-in URI redirect.
Okta sends the authentication response and ID token for the user's sign-in request to these URIs
http://localhost
+ Add URI
Learn More
Sign-out redirect URIs (Optional)
After your application contacts Okta to close the user session, Okta redirects the user to one of these URIs.
+ Add URI
Learn More
Assignments
Controlled access
Allow everyone in your organization to access
Limit access to selected groups
Skip group assignment for now
Select whether to assign the app integration to everyone in your org, only selected group(s), or skip assignment until after app creation.
Enable immediate access (Recommended)
Enable immediate access with Federation Broker Mode
Recommended if you want to grant access to everyone without pre-assigning your app to users and use Okta only for authentication.
To ensure optimal app performance at scale, Okta End User Dashboard and provisioning features are disabled. Learn more about Federation Broker Mode.
Save
Cancel

6. Create **Client secret** from the application's **General** tab
    - Select **Edit**



    - Select Client secret (under client authentication)
    - Click **Save** to have client secret created

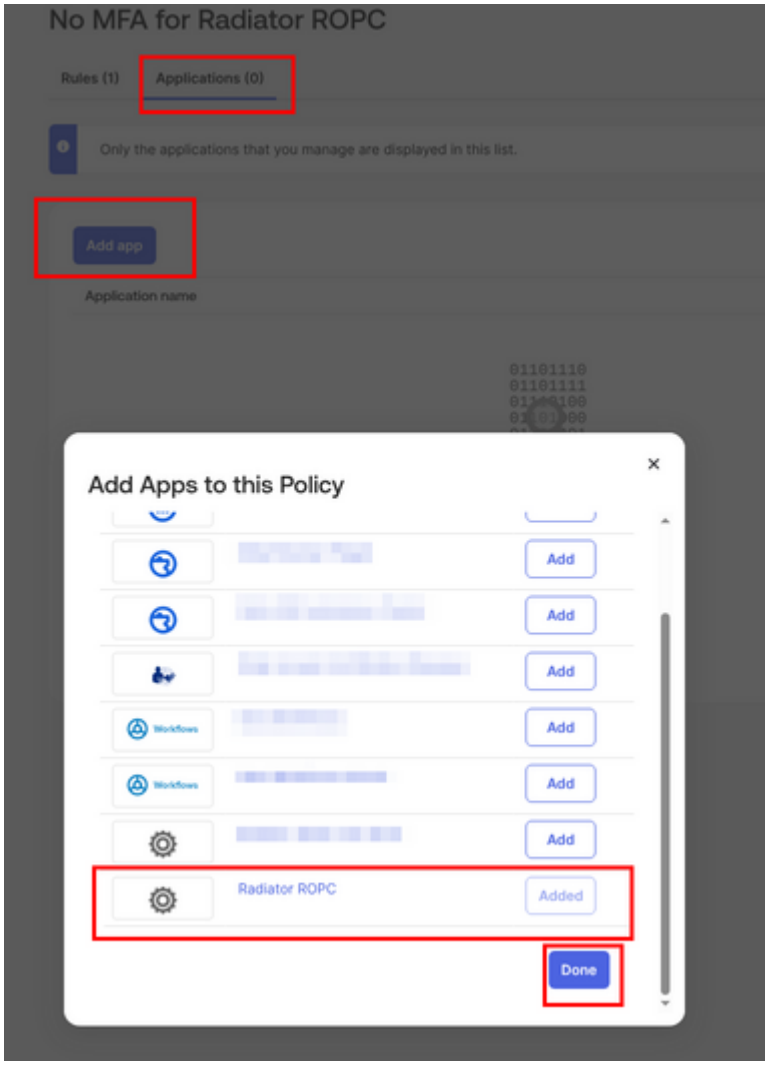


7. Note **Client ID** and **Client secret** down, these are needed on the Radiator configuration side (see Radiator Okta configuration guide)

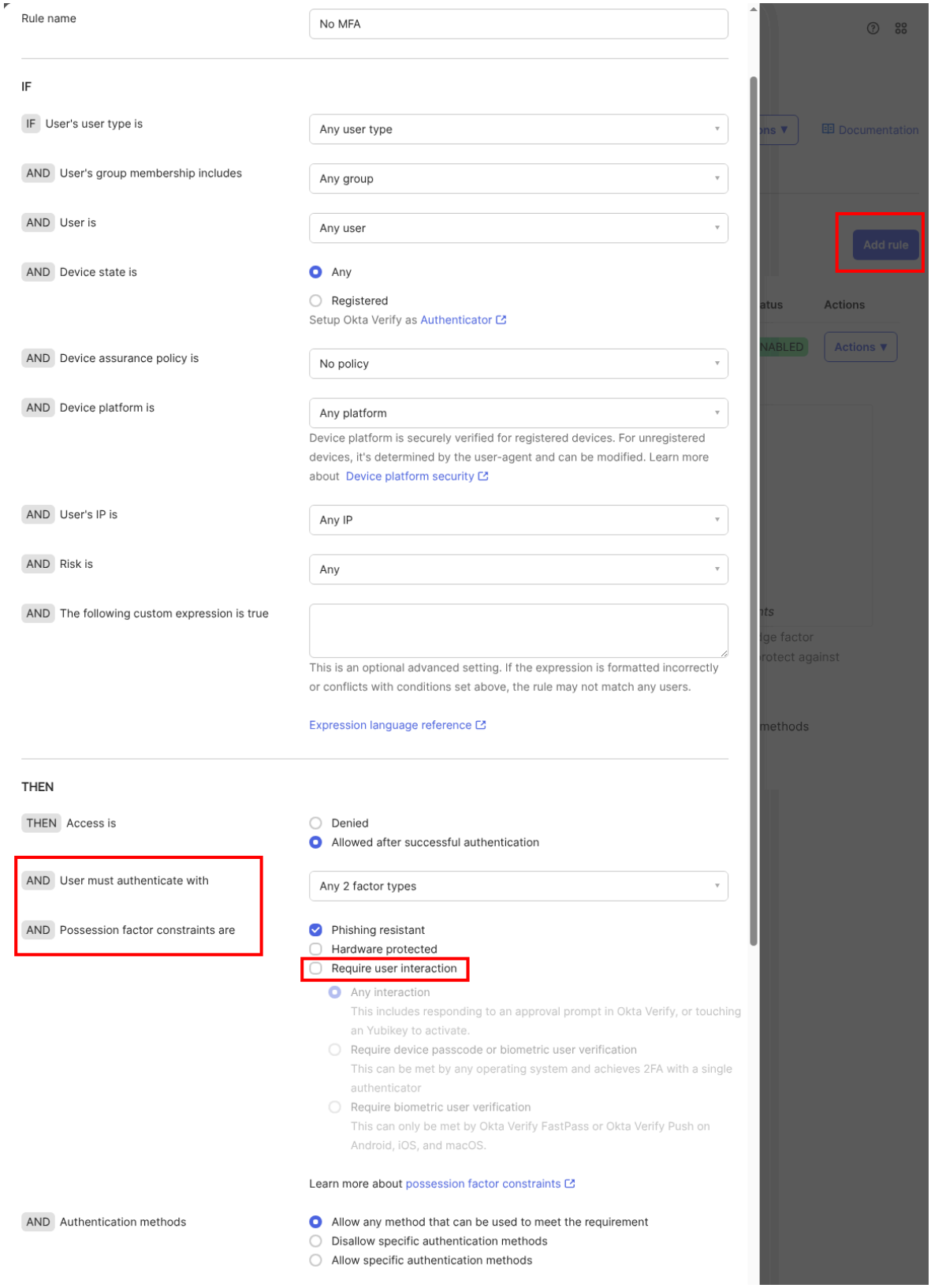
Disable MFA

The ROPC flow does not support MFA. If MFA is enabled, the token request will fail. Ensure MFA is not enforced for the users or groups:

1. Go to **Security > Authenticators**
2. Review active authenticators: if any MFA factors (e.g., Okta Verify, SMS) are set to **Required**, the ROPC flow will be blocked
3. Go to **Security > Authentication Policies** (or **Security > Authentication > App Sign-in Policies** in classic UI)
4. Create a new policy for the ROPC app:
    - Select **Create Policy** and name the new policy e.g. `No MFA for Radiator ROPC`
    - Add Radiator ROPC application to the policy from **Applications** tab. **Add app**, select your app and select **Done**



    - Add a rule that removes the requirement for MFA
    - Select **Add rule** and set **AND Possession factor constraints are** to **Not required**
    - Alternatively: set the authentication requirements in **AND User must authenticate with** to **Password only** (1FA)



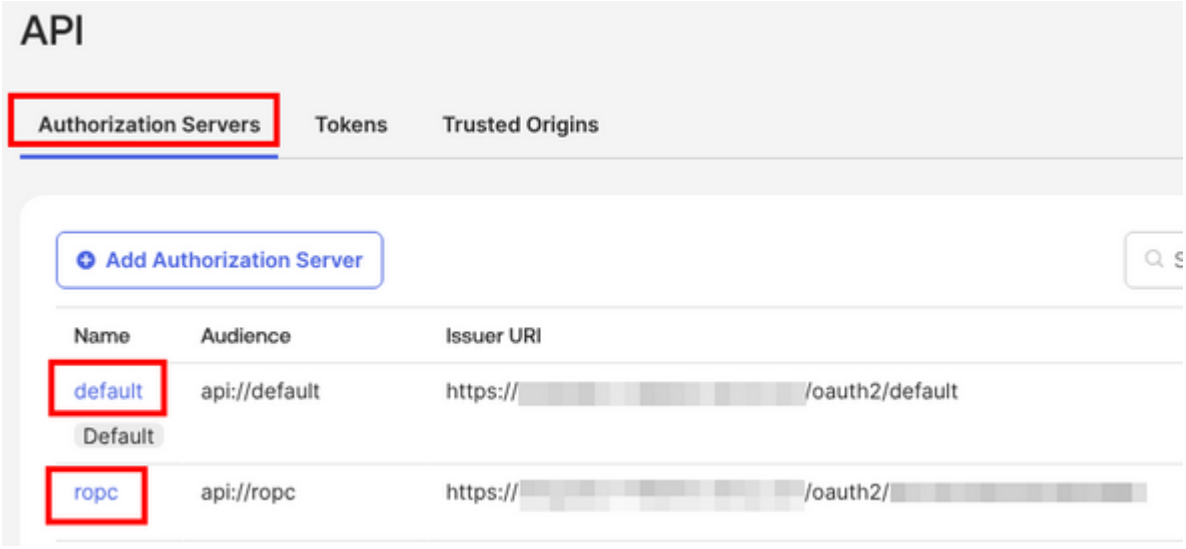
Configure the Authorization Server

The ROPC flow uses Okta's Authorization server. This can be either the Default Authorization server (`/oauth2/default`) or custom Authorization Server. The `groups` claim must be added to the used Authorization Server so that group memberships appear in the `/userinfo` response.

NOTE Using ROPC and adding a custom claim to Authorization server requires the Okta API Access Management add-on.

1. Go to **Security > API > Authorization Servers**

2. Click on **default** or your custom Authorization Server

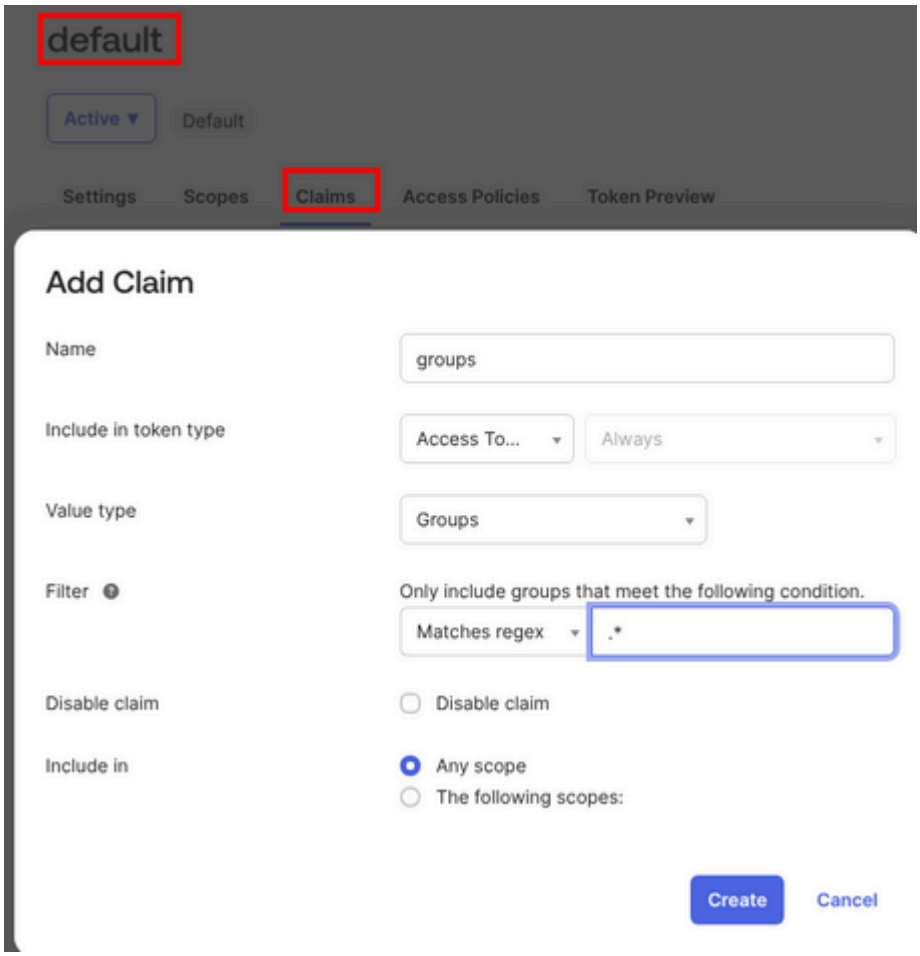


3. Go to the **Claims** tab

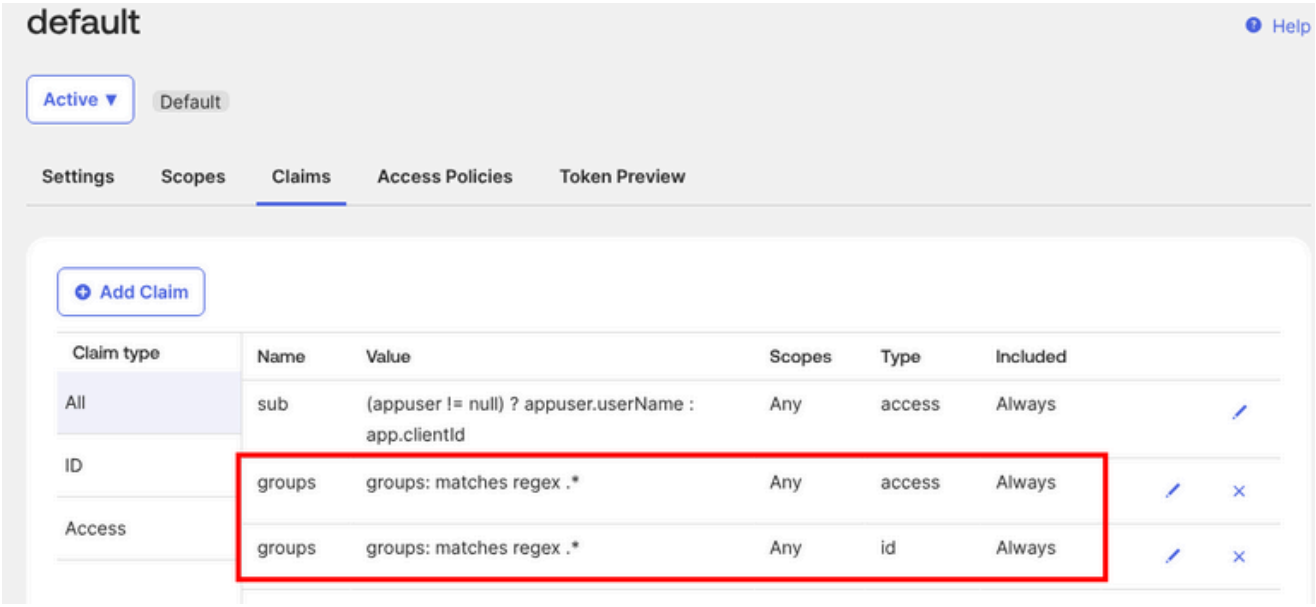
4. Click **Add Claim**:

   - Name: `groups`
   - Include in token type: **Access Token**
   - Value type: **Groups**
   - Filter: **Matches regex** `.*` (this means include all groups, more restrictive filter is possible)
   - Include in: **Any scope**

5. Click **Create**



6. Repeat steps 4 and 5 to token type **ID Token**

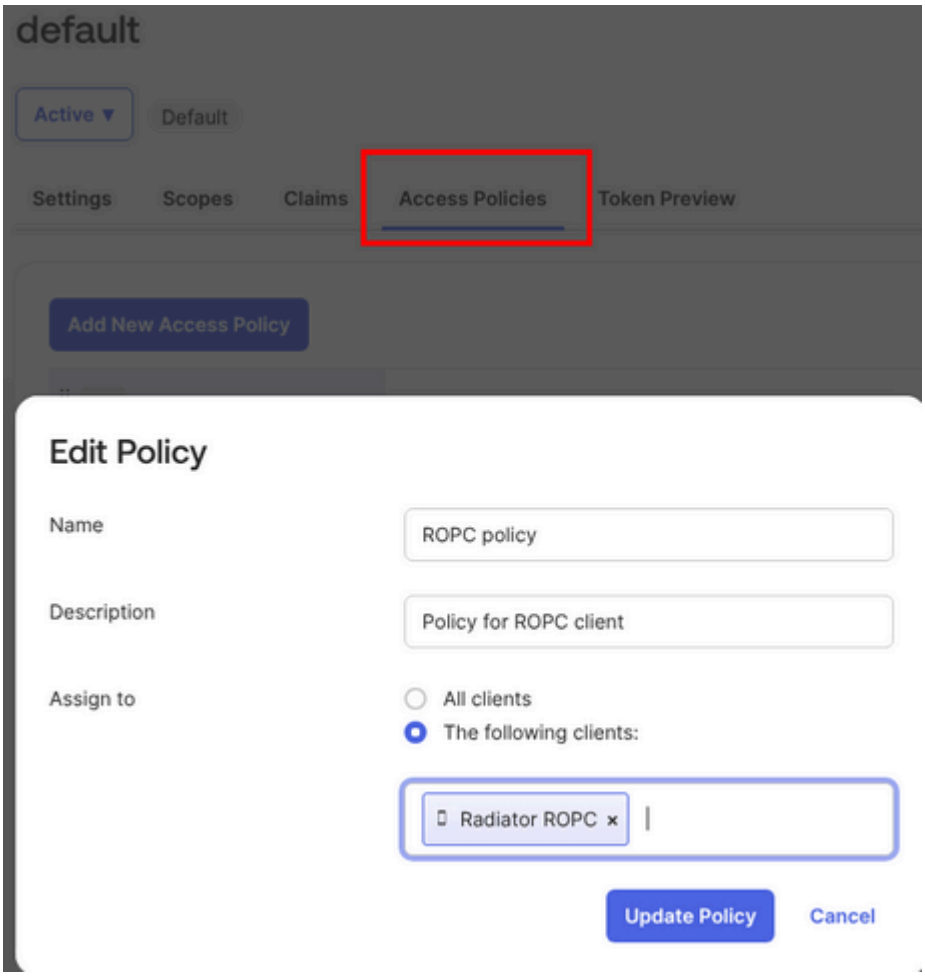


Add Access Policy for ROPC

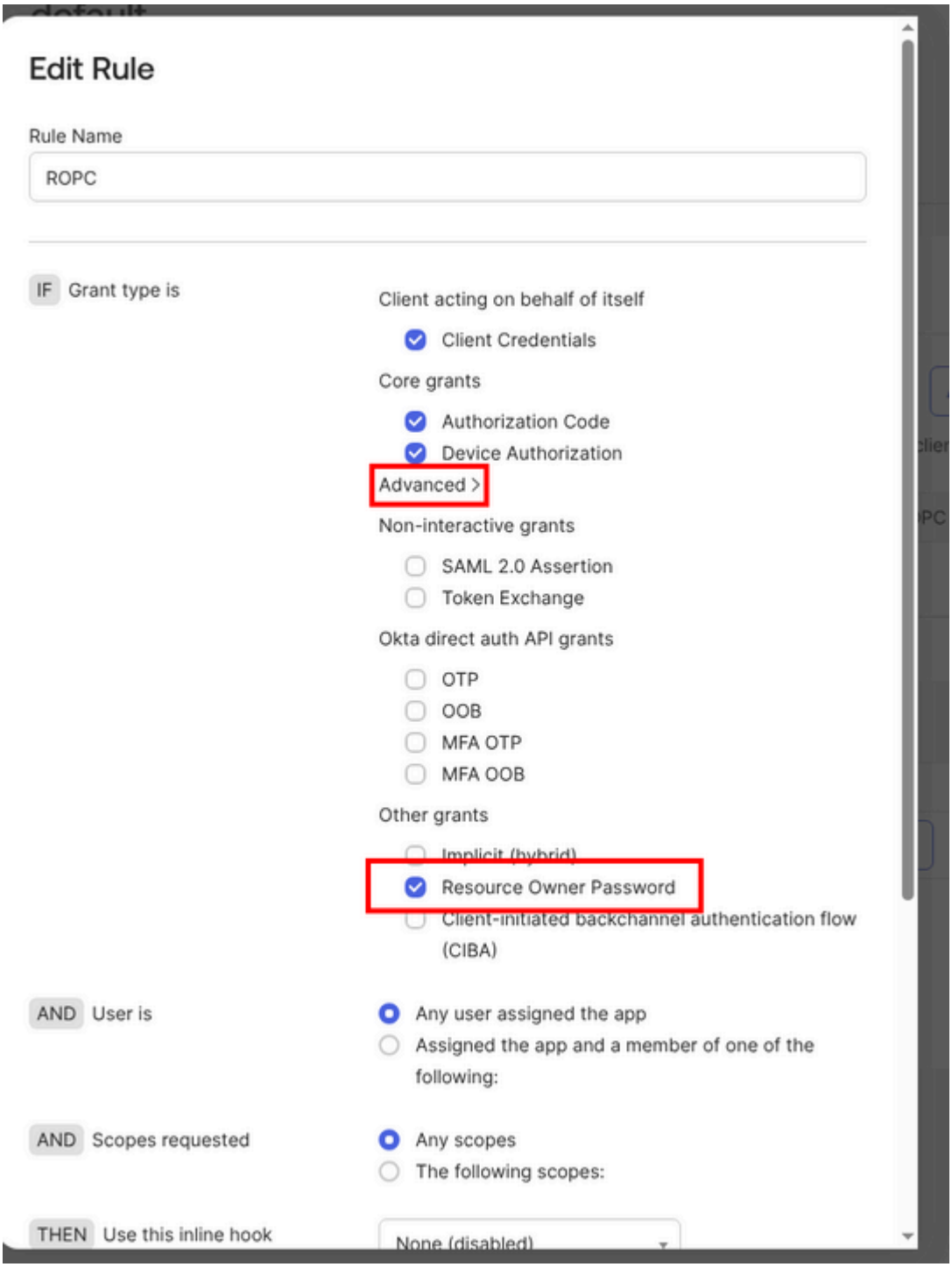
The Authorization Server needs an access policy that allows the ROPC grant type for your application:

1. Go to **Security > API > Authorization Servers**
2. Click on **default** or your custom Authorization Server
3. Go to the **Access Policies** tab

4. Either edit the existing policy or create a new policy:
    - If editing: assign it to the **Radiator ROPC** application
    - If creating: give it a name (e.g., `ROPC Policy`) and assign it to the **Radiator ROPC** application



5. Add or edit a rule within the policy:
    - Grant type: ensure **Resource Owner Password** is checked (under Advanced)



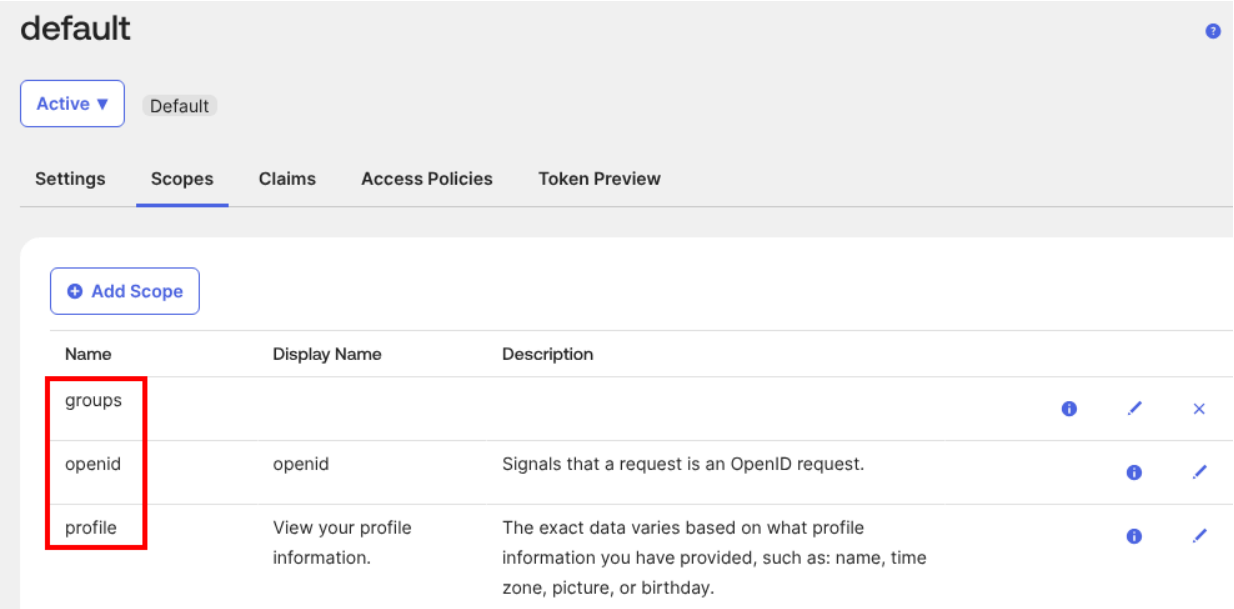
6. Click **Create Rule** / **Update Rule**

Without this policy, token requests will fail even if the application itself has the ROPC grant type enabled.

Verify Scopes

1. In the Authorization Server, go to the **Scopes** tab

2. Ensure these scopes exist:
    - `openid`
    - `profile`
    - `groups` Add any scope that does not exist: click **Add Scope**, name it e.g. `groups`, select **Create**



Radiator Policy Server configuration

After completing the previous steps you will need to update your Radiator Policy Server configuration. See Radiator Policy Server Okta configuration guide for more details.

Troubleshooting

Here is a collection of known errors shown on the Radiator Policy Server authentication log, and how they can be fixed either on Radiator Policy Server configuration or on Okta side. When Radiator Policy Server configuration is edited, remember to restart Radiator Policy Server after changes

Error: The credentials provided were invalid

{"error":"invalid_grant","error_description":"The credentials provided were invalid."

Solution

Check both password and username as either one or both are incorrect

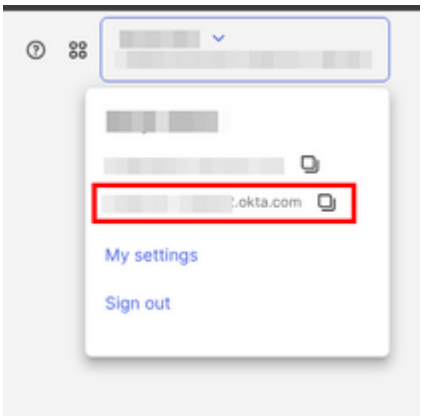
Error: Invalid value for 'client_id' parameter

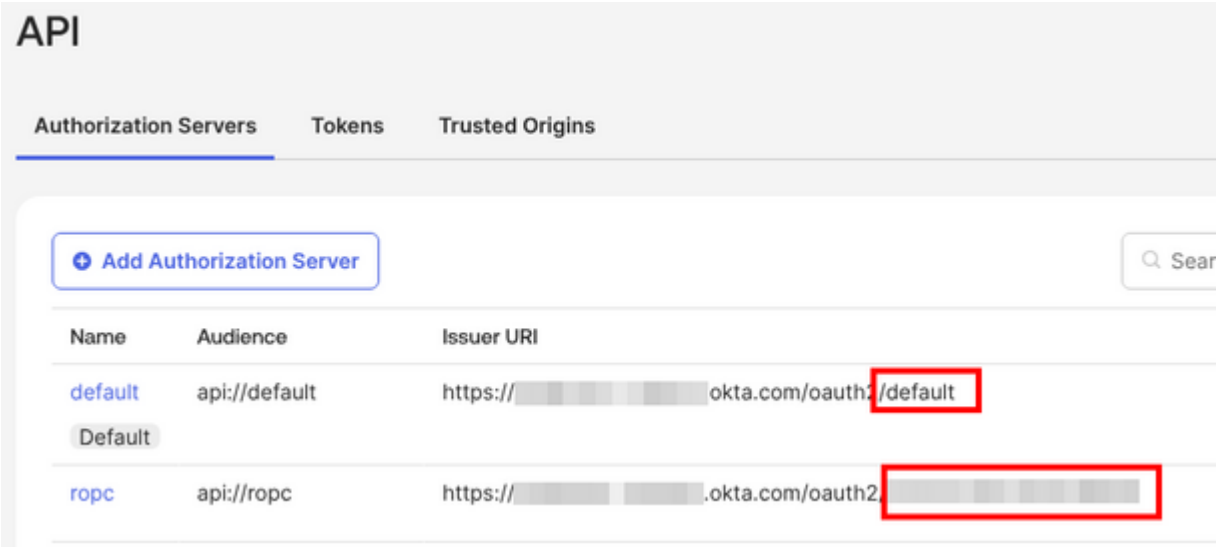
{"errorCode":"invalid_client","errorSummary":"Invalid value for 'client_id' parameter.

Solution

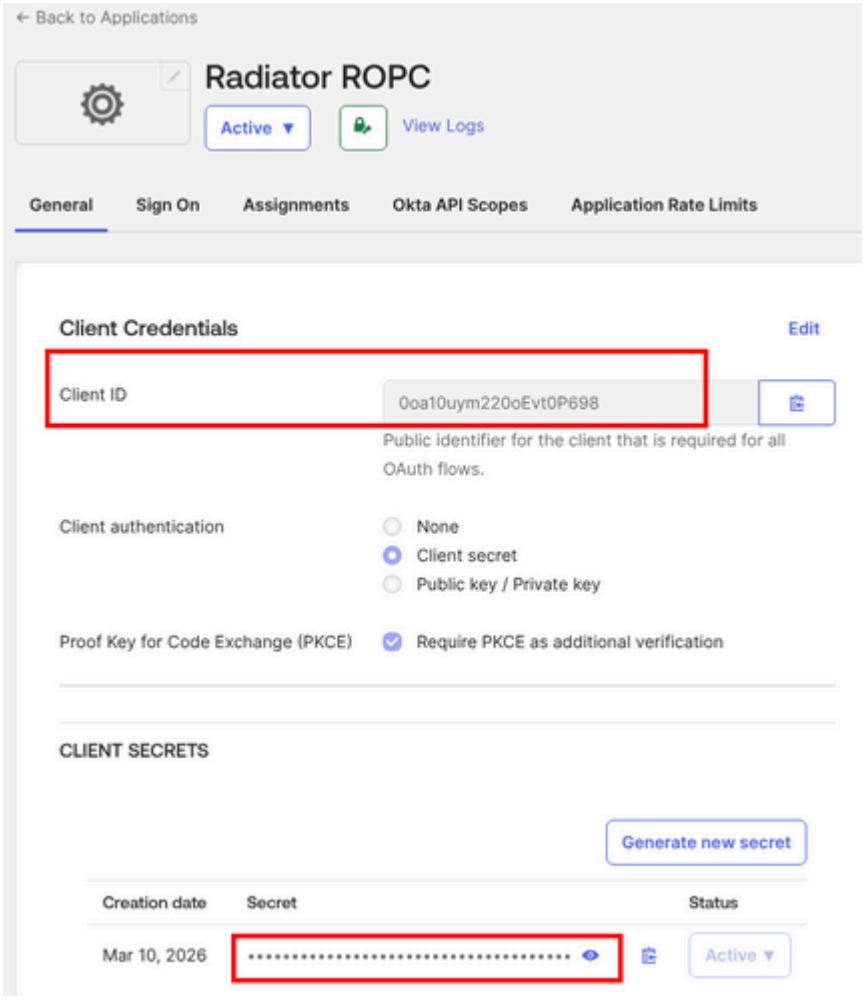
There are multiple solutions to this error:

- Check that Radiator Policy Server configuration and the *vars.aaa.oauth2.issuer-uri* and the *vars.aaa.oauth2.authz-server* have the correct Okta domain and Authorization Server identifier





- Check that Radiator Policy Server configuration and the *vars.aaa.oauth2.client* parameter has the correct Client ID

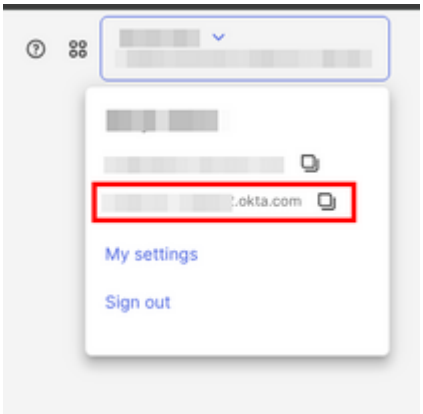


Error: 401 Unauthorized

"Reason":"HTTP backend 'OKTA_ROPC' query 'AUTH_USERS' failed with status "401 Unauthorized":

Solution

Check that Radiator Policy Server configuration and the *vars.aaa.oauth2.okta-domain* has the correct Okta domain

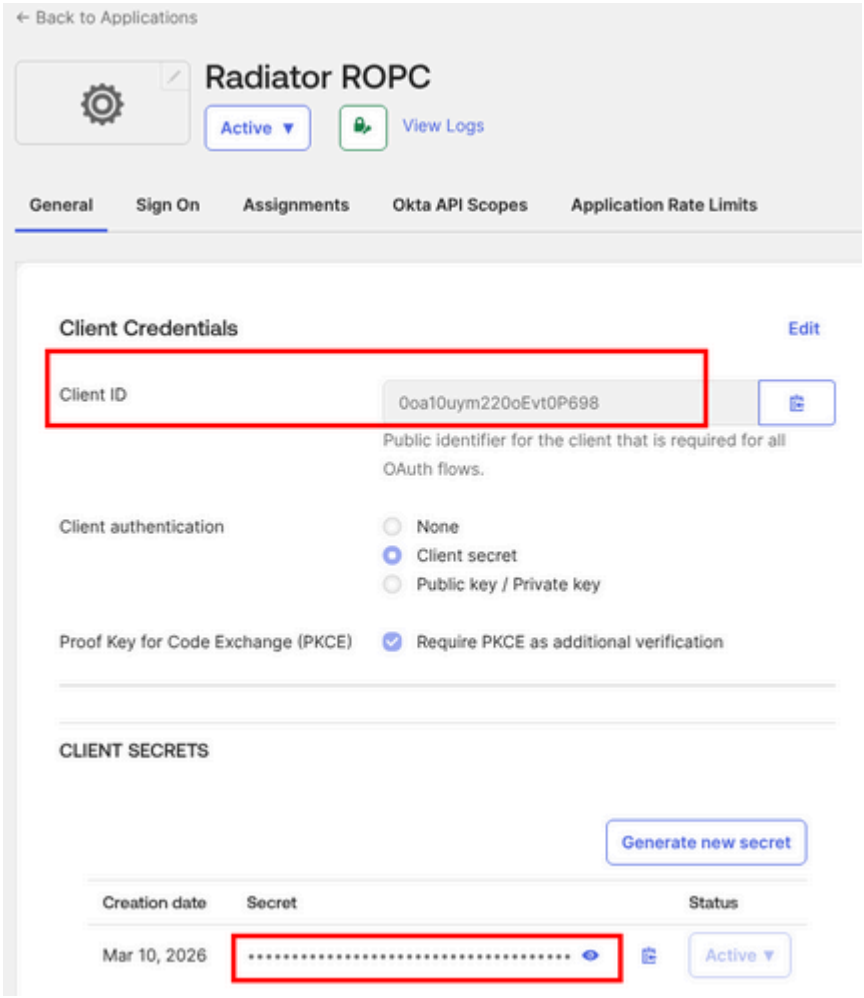


Error: invalid_client

{"error":"invalid_client","error_description":"The client secret supplied for a confidential client is invalid."}

Solution

Check the Radiator Policy Server configuration and the *vars.aaa.oauth2.secret* parameter has the secret value from the correct application.

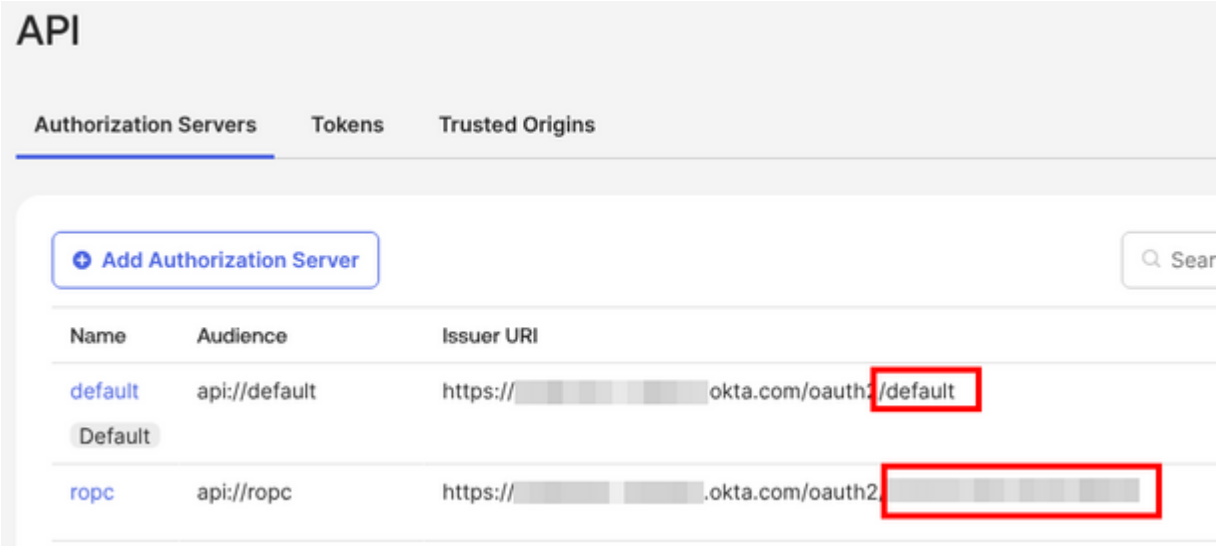


Error: 400 Bad Request

"Reason":"HTTP backend 'OKTA_ROPC' query 'AUTH_USERS' failed with status "400 Bad Request"

Solution

Check that Radiator Policy Server configuration and the *vars.aaa.oauth2.authz-server* has the correct Authorization Server identifier


API
Authorization Servers
Tokens
Trusted Origins
Add Authorization Server
Sear
Name
Audience
Issuer URI
default
api://default
https://okta.com/oauth2/default
Default
ropc
api://ropc
https://.okta.com/oauth2/