



CEPH DAYS SEATTLE

IDENTITY FEDERATION WITH CEPH OBJECT STORAGE

Seth Cagampang
Sr. Engineer, OSNEXUS



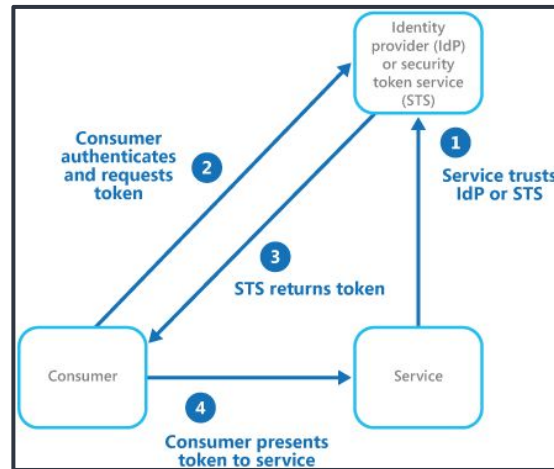
IDENTITY FEDERATION

WHAT IS USER IDENTITY FEDERATION?

- Allows users to authenticate with one identity provider (IdP), but access services and resources hosted by another system without creating separate accounts in each system.

WHY IS IT USEFUL?

- Centralized ID management
- Improved security
- Better user experience
- Easier compliance



<https://learn.microsoft.com/en-us/azure/architecture/patterns/federated-identity>

PROBLEM CASE

- User wants to use Azure directly as an Identity Provider (IdP) for Ceph RGW
- Issue: <https://tracker.ceph.com/issues/54562>
Ceph 18.2.4 Reef 'x5c' certificate parsing bug that causes Ceph RGW STS AssumeRoleWithWebIdentity to fail (backport fix in progress latest activity 04/28/2025)

PROPOSED SOLUTION:

- Keycloak can serve as an intermediary IdP to issue access tokens to Ceph RGW instead

IDENTITY FEDERATION

OPENID CONNECT (OIDC)

- Protocol to authenticate users and issue access tokens and ID tokens
- ID token - who the user is
- Access token - what the user can access

SECURE TOKEN SERVICE (STS)

- Mechanism that allows temporary access credentials to be issued to trusted identities
- Session Token - temporary credentials for making S3 API calls.

KEY COMPONENTS

- Relying Party (e.g. Ceph RGW)
- Identity Provider (e.g., Keycloak or Azure AD)
- End User
- OIDC Discovery

CEPH RGW JSON WEB KEY SET (JWKS) VALIDATION

- Verify token signature **
- Check issuer (iss) and audience (aud)
- Validate any claims (e.g. app_id)
- Enforce IAM (identity and access management) role trust policy

IDENTITY FEDERATION

AZURE AD

- Microsoft's Cloud-based identity and access management (IAM) service



WHAT CAN AZURE AD DO?

- Authentication
- Authorization
- Single Sign-On (SSO)
- Multi-Factor Authentication (MFA)
- Supports Protocols OAuth2, OIDC, and SAML

IDENTITY FEDERATION

KEYCLOAK

- Open-source Identity and Access Management (IAM) solution



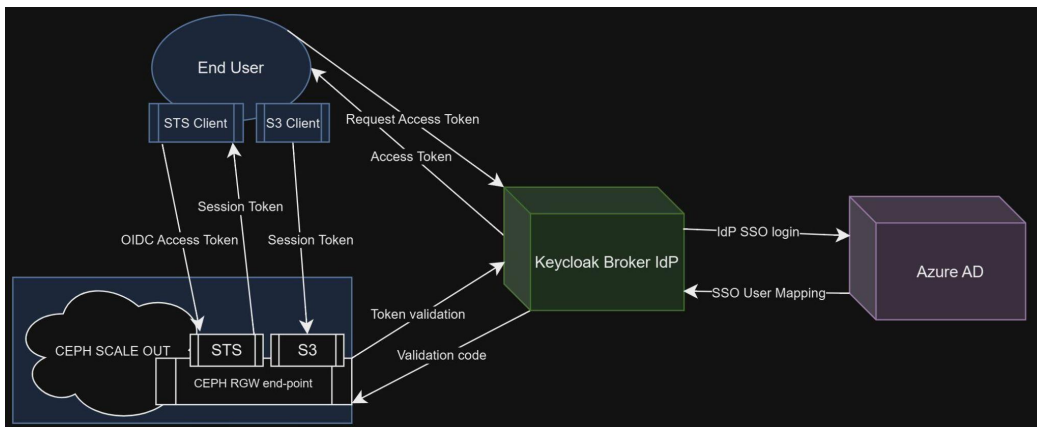
WHAT CAN KEYCLOAK DO?

- Authorization Services
- Single Sign-On (SSO)
- Supports Protocols OIDC, and SAML
- User Federation
- Robust Token Management

IDENTITY FEDERATION

HOW DOES IT WORK?

- Service requires an access token (Ceph RGW)
- User requests token from trusted IdP (Azure) via Single Sign On (SSO)
- IdP issues a token
- A broker (Keycloak) can process and translate that identity and issues its own OIDC token
- Service (Ceph RGW) verifies the token against broker (Keycloak) and issues a temporary session token



SETUP AZURE AS SSO IDP FOR KEYCLOAK

Home > osnexus.com | App registrations

+ New registration Endpoints Troubleshoot Refresh Download Preview features Got feedback?

Overview Preview features Diagnose and solve problems Manage Users Groups External identities Roles and administrators Administrative units Delegated admin partners Enterprise applications Devices App registrations Identity Governance Application proxy Custom security attributes Licenses Cross-tenant synchronization Microsoft Entra Connect Custom domain names Mobility (MDM and WIP)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications Owned applications Deleted applications

Start typing a display name or application (client) ID to filter these results Add filters

2 applications found

Display name ↑	Application (client) ID	Created on ↑	Certificates & secrets
keycloakbroker	[REDACTED]	4/4/2025	Current
qstor-ceph-oidc	[REDACTED]	3/28/2025	Current

Add or remove filters by pressing Ctrl+Shift+F

SETUP AZURE AS SSO IDP FOR KEYCLOAK

Microsoft Azure

Search resources, services, and docs (G+/I)

Copilot

Home > osnexus.com | App registrations >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

keyCloakBroker ✓

Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (osnexus.com only - Single tenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web https://keycloak-host/realm-name/broker/azuread/endpoint ✓

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

SETUP AZURE AS SSO IDP FOR KEYCLOAK

The screenshot shows the Azure portal interface for managing an application named 'keyCloakBroker'. The left sidebar contains a navigation menu with options like Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage, Branding & properties, Authentication, Certificates & secrets (highlighted), Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest, and Support + Troubleshooting. The main content area is titled 'keyCloakBroker | Certificates & secrets' and includes a search bar, a 'Got feedback?' link, and a list of tabs: Certificates (0), Client secrets (1), and Federated credentials (0). The 'Client secrets' tab is active, showing a table with columns for Description, Expires, Value, and Secret ID. A single entry is listed with Description 'id-broker-secret', Expires '4/4/2026', and a redacted Value. An 'Add' button is visible at the bottom right of the table. On the right side of the screen, a modal dialog titled 'Add a client secret' is open, showing a form with a 'Description' field containing 'OIDC app secret' and an 'Expires' dropdown set to '365 days (12 months)'. At the bottom of the dialog are 'Add' and 'Cancel' buttons.

Home > osnexus.com | App registrations > keyCloakBroker

keyCloakBroker | Certificates & secrets

Search

Got feedback?

Overview

Quickstart

Integration assistant

Diagnose and solve problems

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
id-broker-secret	4/4/2026	[REDACTED]	

Add

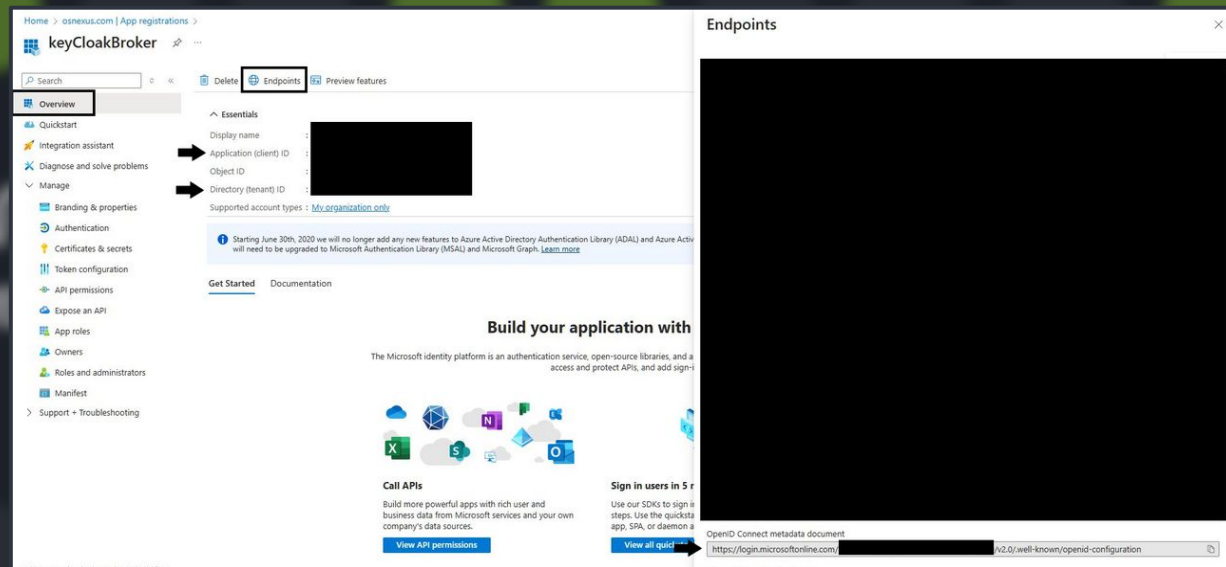
Cancel

Add a client secret

Description: OIDC app secret

Expires: 365 days (12 months)

SETUP AZURE AS SSO IDP FOR KEYCLOAK



SETUP AZURE AS SSO IDP FOR KEYCLOAK

Home > keycloakBroker

keycloakBroker | API permissions

Search Refresh Got feedback?

Overview

Quickstart

Integration assistant

Diagnose and solve problems

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

Granting tenant-wide consent may revoke permissions that have already been granted tenant-wide for that application. Permissions that use

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of co all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission Grant admin consent for osnexus.com

API / Permissions name	Type	Description	Admin consent requ.
Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	No

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise application](#)

Request API permissions

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

Start typing a permission to filter these results

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

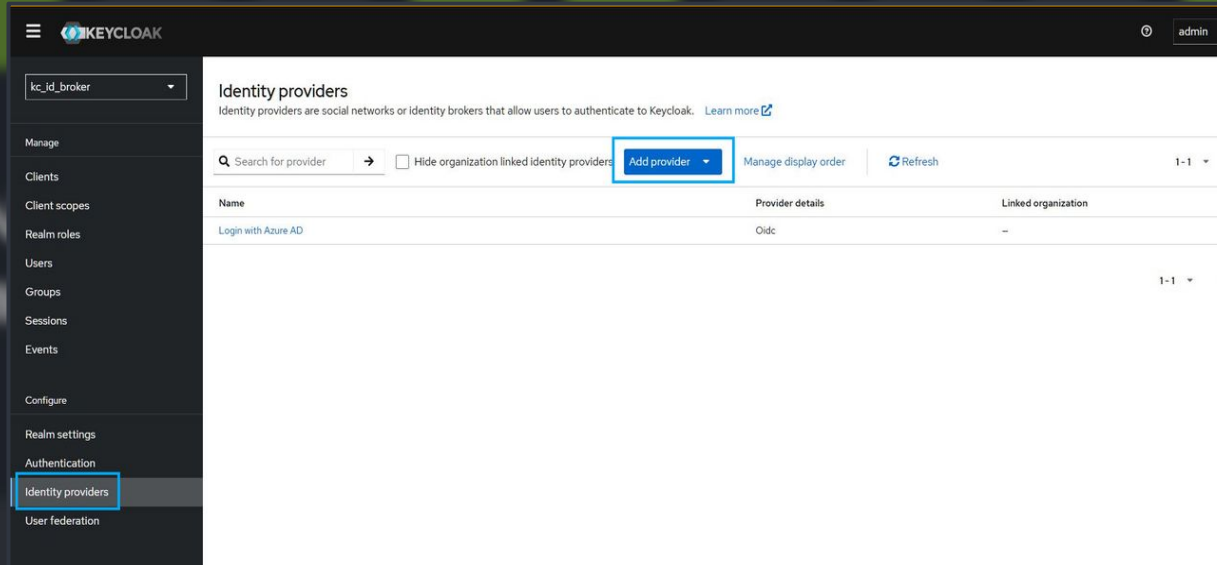
Permission	Admin consent required
Openid permissions (3)	
<input checked="" type="checkbox"/> email	No
<input type="checkbox"/> View users' email address	No
<input type="checkbox"/> offline_access	No
<input type="checkbox"/> Maintain access to data you have given it access to	No
<input checked="" type="checkbox"/> openid	No
<input checked="" type="checkbox"/> Sign users in	No
<input checked="" type="checkbox"/> profile	No
<input checked="" type="checkbox"/> View users' basic profile	No

> AccessReview

Add permissions Discard

SETUP AZURE AS SSO IDP FOR KEYCLOAK

Keycloak Version: 26.1.4



The screenshot shows the Keycloak administration console. The left sidebar contains a menu with the following items: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers (highlighted with a blue box), and User federation. The main content area is titled 'Identity providers' and includes a description: 'Identity providers are social networks or identity brokers that allow users to authenticate to Keycloak. [Learn more](#)'. Below the description, there is a search bar, a checkbox for 'Hide organization linked identity providers', and an 'Add provider' button (highlighted with a blue box). To the right of the button are links for 'Manage display order' and 'Refresh', and a dropdown menu showing '1-1'. A table below lists the existing identity providers:

Name	Provider details	Linked organization
Login with Azure AD	OIDC	-

At the bottom right of the table, there is a dropdown menu showing '1-1' and a refresh icon.

SETUP AZURE AS SSO IDP FOR KEYCLOAK

KC_ID_BROKER


Sign in to your account

Username or email

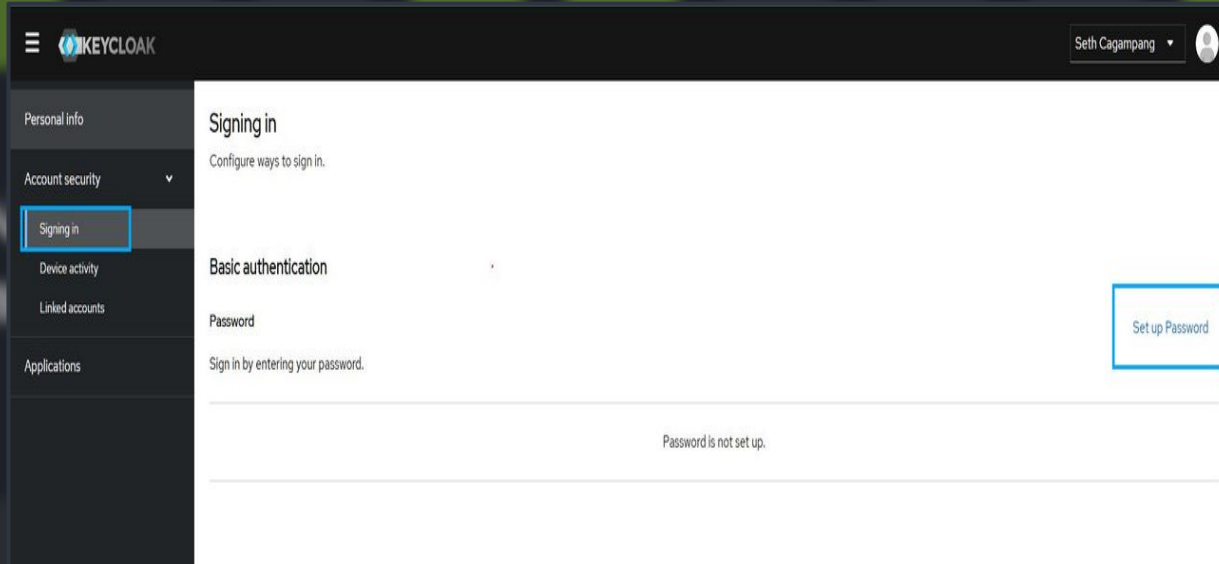
Password

Sign In

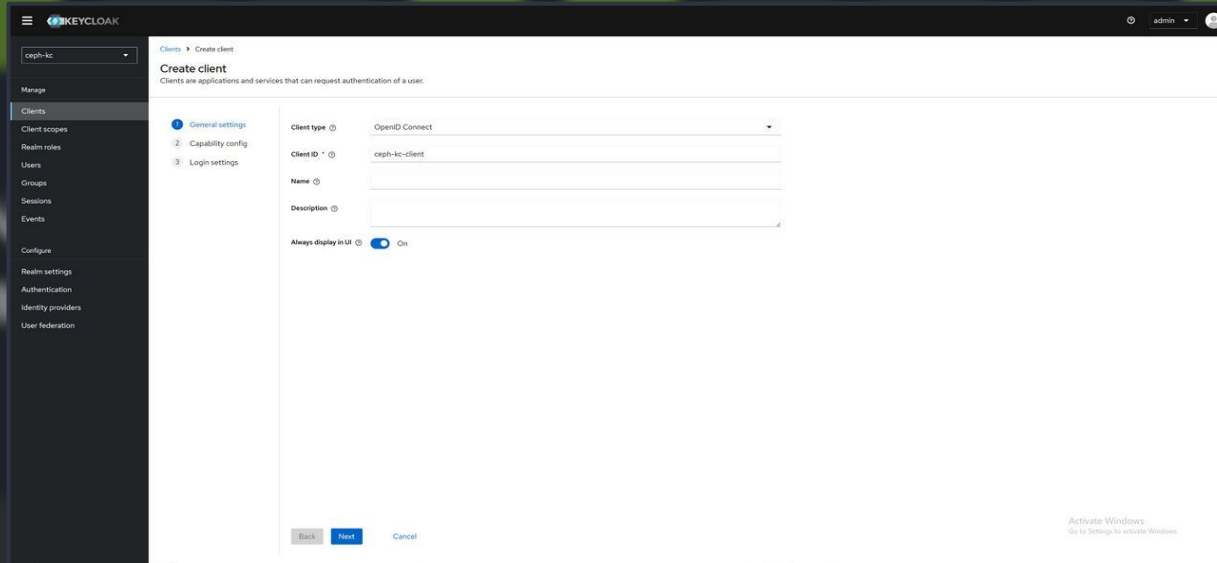
Or sign in with

 Login with Azure AD

SETUP AZURE AS SSO IDP FOR KEYCLOAK



SETTING UP KEYCLOAK AS IDP FOR CEPH RGW



The screenshot shows the Keycloak administration interface. On the left is a dark sidebar with a menu containing: Manage, Clients (highlighted), Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Clients' and 'Create client'. Below this is a sub-header 'Create client' with a description: 'Clients are applications and services that can request authentication of a user.' There are three numbered steps: 1. General settings (active), 2. Capability config, and 3. Login settings. The 'General settings' section contains the following fields: 'Client type' (a dropdown menu showing 'OpenID Connect'), 'Client ID' (a text input field containing 'ceph-kc-client'), 'Name' (an empty text input field), and 'Description' (an empty text input field). At the bottom of this section is a toggle switch for 'Always display in UI', which is currently turned 'On'. At the very bottom of the form are three buttons: 'Back', 'Next' (highlighted in blue), and 'Cancel'. In the bottom right corner of the interface, there is a small 'Activate Windows' watermark.

SETTING UP KEYCLOAK AS IDP FOR CEPH RGW

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Capability config

Client authentication ☒ On

Authorization ☒ On

Authentication flow

- ☒ Standard flow
- ☐ Implicit flow
- ☐ OAuth 2.0 Device Authorization Grant
- ☐ OIDC CIBA Grant
- ☒ Direct access grants
- ☒ Service accounts roles

SETTING UP KEYCLOAK AS IDP FOR CEPH RGW

KEYCLOAK

ceph-kc

Manage

- Clients
- Client scopes
- Realm roles
- Users
- Groups
- Sessions
- Events
- Configure
- Realm settings
- Authentication
- Identity providers
- User federation

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

- General settings
- Capability config
- Login settings**

Root URL:

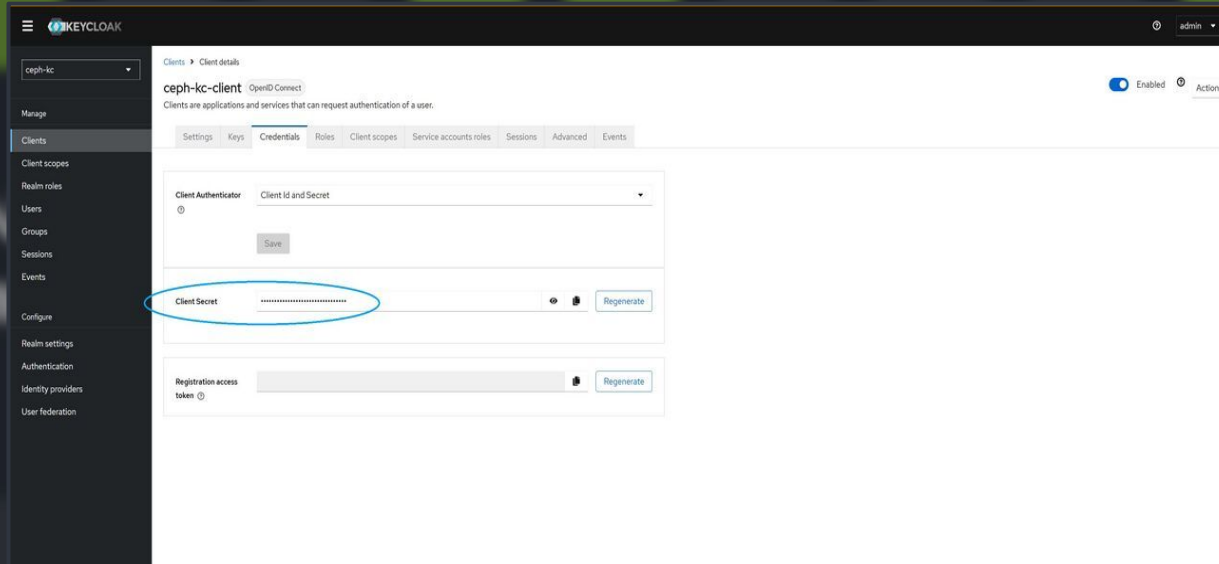
Home URL:

Valid redirect URIs: [Add valid redirect URIs](#)

Valid post logout redirect URIs: [Add valid post logout redirect URIs](#)

Web origins: [Add web origins](#)

SETTING UP KEYCLOAK AS IDP FOR CEPH RGW



GITHUB

<https://github.com/>

GET ACCESS TOKEN

Access tokens can

```
curl -k -v -X POST
-H "Content-Type:
-d "scope=openid"
-d "grant type=password"
-d "client id=$KID"
-d "client secret=$SECRET"
-d "username=$KID"
-d "password=$SECRET"
"http://$KID:$SECRET@localhost:8080/realms/master/protocol/openid-connect/token"
```

[illegible]

USING ACCESS TOKEN WITH CEPH RGW STS

CREATE CEPH RGW USERS FOR OIDC MANAGEMENT AND STS CLIENT

IAM User - Creates OIDC Provider object in CEPH RGW and defines IAM roles.

```
radosgw-admin --uid TESTUID1 --display-name "iam_user" --access_key TESTUID1 --secret test123 user create
radosgw-admin caps add --uid="TESTUID1" --caps="oidc-provider=*"
radosgw-admin caps add --uid="TESTUID1" --caps="roles=*
```

STS Client - Consumes access token, to receive a session token from OIDC Provider

```
radosgw-admin --uid TESTUID2 --display-name "sts_client_user" --access_key TESTUID2 --secret test321 user create
radosgw-admin caps add --uid="TESTUID2" --caps="roles=*
```

USING ACCESS TOKEN WITH CEPH RGW STS

GENERATE IDP CERT THUMBPRINTS

```
root@ ~ # ./scripts/get_thumbprints.sh https://auth.keycloaktest.com:8443/re
alms/kc_id_broker/.well-known/openid-configuration
Processing URI: https://auth.keycloaktest.com:8443/realms/kc_id_broker/protocol/openid-connect/certs
Assembling Certificates....
-----BEGIN CERTIFICATE-----
MIICPzCCAY8CBGw0JZAGDANBgkqhkiG9w0BAQsFADAXMRUwEwYDQ0QDAAxY19pZF9icm9rZXIwHhcNMjUwNTE0MjA1NTI5MWhcNMzUwNTE0MjA1NzA5WjAXMRUwEwYDQ0QDAAxY19pZF9icm9rZXIwggEiMA0GCSqGSIb3D
QEBAQUAA4IBDwAwggEKAoIBAQDAX+aCKETKnyfjcs1gGLnVZ42tVPKw15IyeIGdwzSTW08Vum16w+RSyYBCZ1qu4xwdVhAu6qb7vnEoa+zdszr8hYq44fXWZRLn4j5hQx01n8Kc13mRCZ/5pPwCJzAwBvS1pO9Q9sdbRwLk
FLEXMTRjXMcTfX3MfE0hpzKnK3jeXcc2TeTy87kZBbYw5/KAB++yCd/6BGI7L8gA0P7r+RnHP0g2B0zc330i8hDnKL8/8jvZVMPmg3vdjKk1I8CCvMfXI7qefppye3nFAfAfeZr0moBN1LjZiQ09ZACXWmyvADbriC9w
bgjx48/nbWmbz8fLOS7VQm0FfmVTAgiWBAAEwDQYJKoZIhvcNAQELBQADggEBAIKaZXiF0KjMWRVSILNnZNUHezbcFXpyXT/AN+6Z3zw6nliYkh8qpabZIpTq1pesW/vWvEbct3YkpWQAQOQ5tpPuhvA8y0JRibsJ29F/yqfF
4FoMKEBANwB2zStyH5dbR77KZYksLdydI10dQV+yQ03gdseCMyK1BUCi1Ycb9WdA+y/a03HB6FyC50ZEaWkD5v19sVAECuMesahEHGoVE0X6f6S1rmrW0Y2jBGORefdP7s9+ANSHx+AHV/kMX2eDu6kggGc7rxBMRT1l
in2LeP/q9qR+UMFRp14BXvOCAySuV108H6IKo8fg9PEpargmL4b8140c0j0n9EnIGU=
-----END CERTIFICATE-----
Generating thumbprint for certificate 1....
-----BEGIN CERTIFICATE-----
MIICPzCCAY8CBGw0JZBfzANBgkqhkiG9w0BAQsFADAXMRUwEwYDQ0QDAAxY19pZF9icm9rZXIwHhcNMjUwNTE0MjA1NTMwHhcNMzUwNTE0MjA1NzEwWjAXMRUwEwYDQ0QDAAxY19pZF9icm9rZXIwggEiMA0GCSqGSIb3D
QEBAQUAA4IBDwAwggEKAoIBAQzS9HhXXcEyTidAc1znNUz1kyKsja2TKTQPjvSXIcFbgde9w8jr+tuU85xqmHkJKU4jnfStVdYQ0Pa25jcs1sZwKxLteIUeg3t8yWda5pFe0zdA1R5j6Tk6FvnyMepCGRw0BD0w5y2a0
F19eiMtJCRN61f61ZJeTsXNWUPRcpG4Fef640C5tINH0zRS9w0fpr+F0fvFvDcLKHknVmmFIw5jQkmYmp/0CwMdm8+a0SmlWA2vBL+RStzh10N/oxVfWNL78LTLBTLXBS81IDB640QP4kddisEPYFbGf0t0t2te3pt
zt+1SIFYOpS8p9NTKd8BsAgf4Ey+XAgWBAAEwDQYJKoZIhvcNAQELBQADggEBAKL6sJwOG4Wsu35bh+PCx6v8I8gEDBHOxfNjy5ySQ1RPjfwrdUjfq8b1RjxHrgNJWvvyMQ0qfku24FgLHjGYbtr9NPLsQ0U2MwLq2t1gVHk
UKJMF3i1Cv36WjRcw4jKdgvZct600HontJYNeKwNaOvX1XDALT9nvqESu4FnVBiiUhr5Y5a0PjXvqgBZ88+wp/8Swe1QFHh3WnLfwcsjcfFIAPjKTI0b17LWSkw3NAJ6K00a03VsAs1v5LDX76HZ4iFTgDh5sTVr0s1rnv+o
s2m3873GHZvoq7gwVDHXI1gCHFRXyX72TyqJ3j/gKyfKe4iC1A7mIdvKezaW2DZEY=
-----END CERTIFICATE-----
Generating thumbprint for certificate 2....
root@ ~ # cat thumbprints.txt
119FF33B5256857A92954375292C40CA08C9B6EA8
343D140B4F074071207487793737B5D52910A7D9
```

USING ACCESS TOKEN WITH CEPH RGW STS

CREATE IAM CLIENT AND OIDC PROVIDER

- Create 'iam' client using 'iam_user' credentials for your Ceph RGW endpoint
- Register Keycloak as an OIDC Provider for Ceph using create_open_id_connect_provider()

```
try:
    oidc_response = iam_client.create_open_id_connect_provider(
        Url=oidc_app_endpoint,
        ClientIDList=[
            oidc_client_id
        ],
        ThumbprintList=ThumbprintListIn
    )
    print("Successfully created open id connect provider...")
except Exception as e:
    print(e)
```

USING ACC

ADD "S3ACCE

- Using the assume
- Set a role actions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": [
          "arn:aws:iam::oidc-pr"
        ]
      },
      "Action": [
        "sts:AssumeRoleWith"
      ],
      "Condition": {
        "StringEquals": {
          "<key-cloak-host>:8060/realms/<realm name>/app_id . account"
        }
      }
    }
  ]
}
```

federated realm to

USING ACCESS TOKEN WITH CEPH RGW STS

CREATE STS CLIENT AND USE ACCESS TOKEN TO ASSUMEROLEWITHWEBIDENTITY

- Using the 'boto3.client' create 'sts' client using 'sts_user' credentials for your Ceph RGW endpoint
- With the 'sts_client' created we can try to assume the "S3Access" role using the access token issued by Keycloak.

```
try:
    response = sts_client.assume_role_with_web_identity(
        RoleArn=roleResponse['Role']['Arn'],
        RoleSessionName='Bob',
        DurationSeconds=3600,
        WebIdentityToken=<Web Token>
    )
except Exception as e:
    print(e)
```

- "response" contains session token if access token validation is successful.

USING ACCESS TOKEN WITH CEPH RGW STS

CREATE CEPH RGW CLIENT

- Session token can be used by “s3” client, which can perform s3 actions.

```
try:
    s3client = boto3.client('s3',
        aws_access_key_id = response['Credentials']['AccessKeyId'],
        aws_secret_access_key = response['Credentials']['SecretAccessKey'],
        aws_session_token = response['Credentials']['SessionToken'],
        endpoint_url=<S3 URL>,
        region_name=<S3 region>)
    bucket_name = 'my-bucket'
    s3bucket = s3client.create_bucket(Bucket=bucket_name)
except Exception as e:
    print(e)
```

USER AUTHORIZATION FROM KEYCLOAK

USER ACCESS CONTROL

- Authentication is handled by Azure AD
- Authorization can be centralized in Keycloak, Azure, or split depending on your goals

RECOMMENDED: CENTRALIZED AUTHORIZATION IN KEYCLOAK

- “Mapper” feature inject roles, groups, custom claims, etc for federated users.
- Access tokens issued will be issued with embedded mappings
- Ceph RGW uses those claims to control access.

WHY?

- Control fine-grained authorization inside Keycloak
- Keeps Azure AD simpler
- Better for multi-IdP scenarios (e.g., adding Google, GitHub later)
- Ceph RGW doesn't need to know about Azure AD

THANK YOU FOR YOUR TIME!

SOURCES:

Azure OIDC:

<https://learn.microsoft.com/en-us/entra/identity-platform/v2-protocols-oidc>

Keycloak w/ RadosGW docs:

<https://docs.ceph.com/en/latest/radosgw/keycloak/>

Boto3 Docs:

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sts.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/iam.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html>

SSO for Keycloak:

<https://docs.getvisibility.com/enterprise-setup/authentication/single-sign-on-sso/using-azure-ad-as-keycloak-identity-provider>

Keycloak Getting Started:

<https://www.keycloak.org/getting-started/getting-started-docker>

ADDITIONAL GUIDES:

Github Scripts:

<https://github.com/OSNEXUS/KeyCloak-w-Ceph-RADOSGW>

OSNEXUS Wiki:

https://wiki.osnexus.com/index.php?title=KeyCloak_Integration

https://wiki.osnexus.com/index.php?title=KeyCloak_Azure_Federation

