Microsoft Entra ID SSO for Gitlab

> **Prerequisites:** Ability to create an app registration with delegated standard rights and access to the Gitlab Docker volume. Gitlab should be installed and administrator access to the web interface should be available.

This guide describes how a Gitlab Docker instance can be equipped with all the benefits of Single Sign On (SSO) using Microsoft Entra ID. Gitlab offers options to make granular settings for the login behaviour.

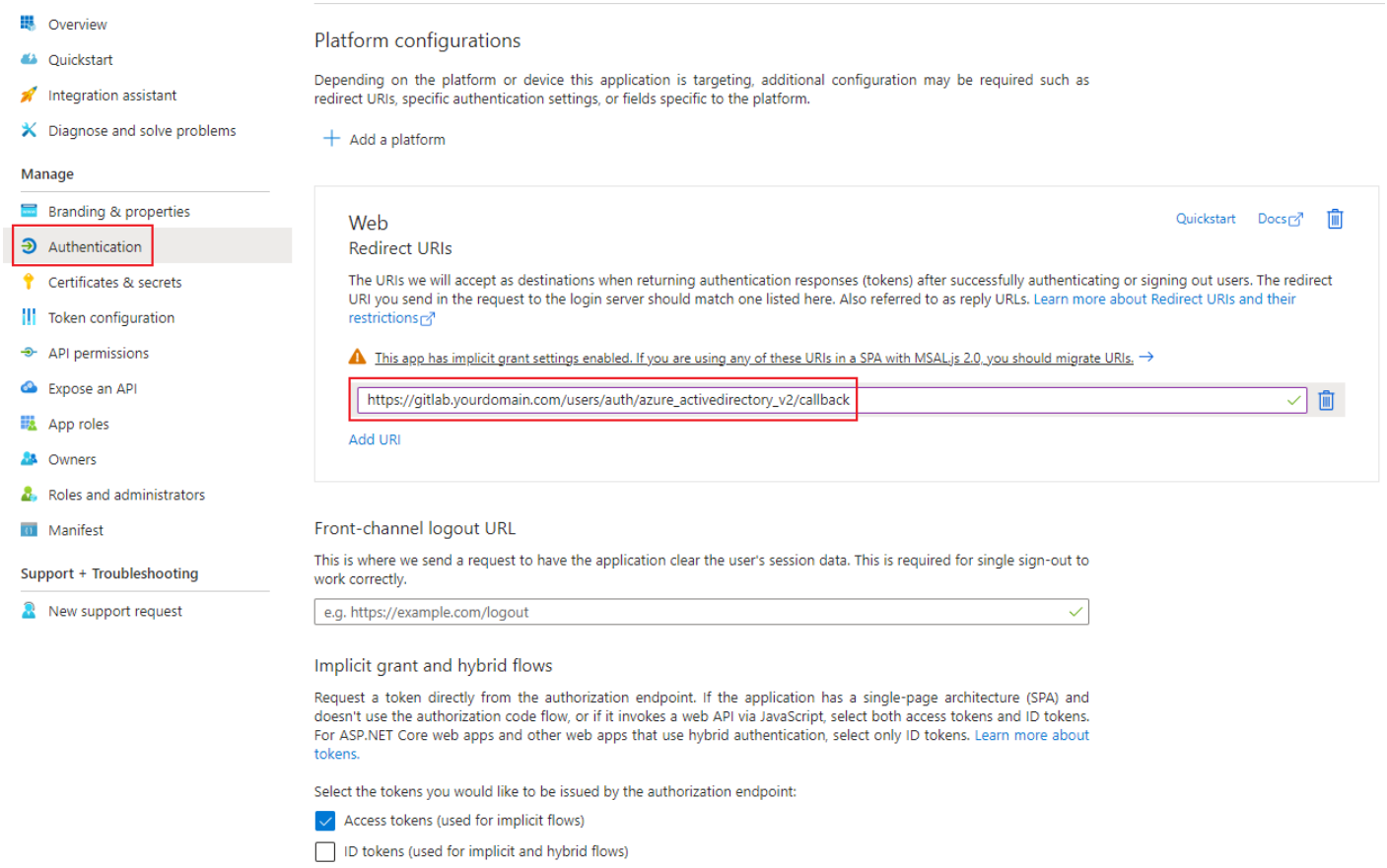
Things to consider & limitations

In order to be able to log in to the GIT CLI using OAuth, it is essential to work through the following instructions. Since your own Gitlab instance does not yet have an application with which the CLI can authenticate itself. After you have completed this setup, you must note the following: [OAuth SSO in CLI using... | LNC DOCS (lucanoahcaprez.ch)]

Create App Registration

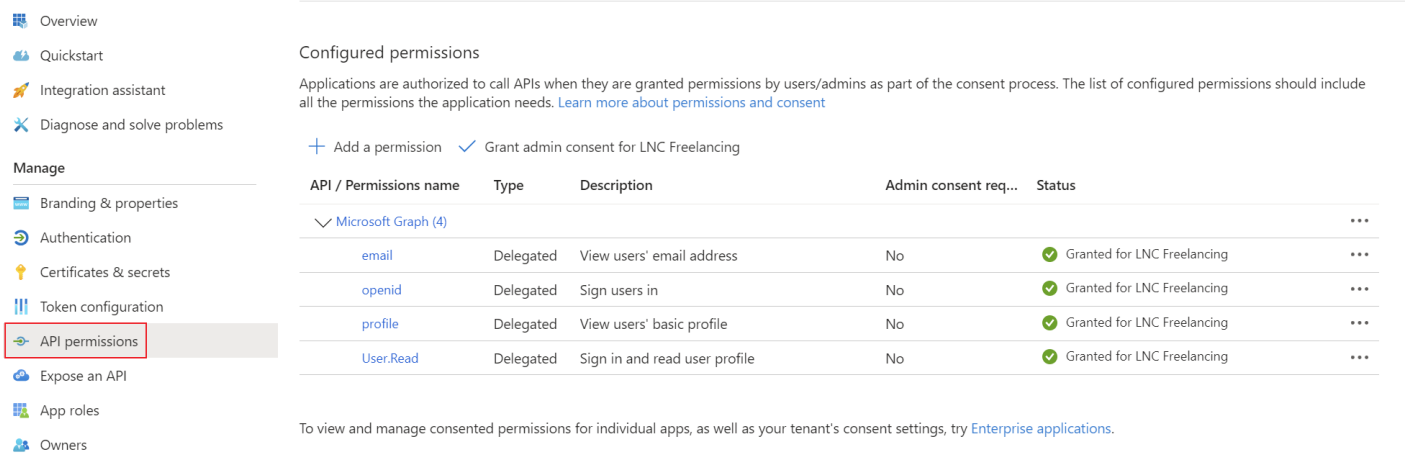
First, an app registration including client secret must be created in Microsoft Entra ID. All settings can be left at the default values. Important settings are the Redirect URIs under the Authentication tab. Set these URIs to your external or internal domain on which Gitlab is available. These URIs will be used for Microsoft Entra ID to know where to redirect the user in case of successful logins.

- **Authentication Type:** Web
- **Redirect URIs:**
  https://gitlab.yourdomain.com/users/auth/azure_activedirectory_v2/callback



Add the corresponding permissions for OpenID Connect as delegated permissions and grant admin consent for your tenant.

- **Permissions:** Delegated OpenId permissions (email, offline_access, openid, profile)



Create a client secret for the application and save the tenant ID, application ID and client secret in your password manager. You can find instructions for this information here: [Get app details and grant permissions to app registration]

Setup Microsoft Entra ID login provider

This step requires the authentication details TenantID, ClientID, Client Secret from the first step.

With Gitlab, the Microsoft Entra ID configurations can be set up using the GITLAB_OMNIBUS_CONFIG environment variable. The following part of the variable enables the configuration of the OpenID integration.

The relevant environment variables for OpenID are as follows:

- **gitlab_rails['omniauth_enabled']:** This boolean represents wheter OAuth is enabled or not.
- **gitlab_rails['omniauth_allow_single_sign_on']:** The providers are specified here. All providers that are to be used in parallel can be specified here.
- **gitlab_rails['omniauth_block_auto_created_users']:** If this setting is set to true, auto-created user must be admin approved.
- **gitlab_rails['omniauth_providers']:** This is the main part of the configuration. Because it is an object that is assigned here, these sub-levels are defined:
  - **name:** This is the display name for the button on the sign in screen.
  - **label:** This is the display name for the button on the sign in screen.
  - **args:** Here you will define the App Registration credentials from the previous step.
    - **client_id:** This is the client id of your App Registration in Microsoft Entra ID.
    - **client_secret:** Here you have to provide the Client Secret from the App Registration.
    - **tenant_id:** Here you have to enter the Tenant ID of your Microsoft Entra ID Tenant.

Side note: As this type of configuration involves the omnibus variable, these contents can also be transferred via the volume as the gitlab.rb file, specified with a simple startup command or specified in another declarative context (Kubernetes manifest, Terraform, etc.).

Docker compose example

This Docker Compose file shows a possible configuration for Gitlab that authenticates using Microsoft Entra ID. In addition mail settings are also specified.

Customize this content with your specifications and save the content in a normal docker-compose.yaml file. As this is Docker Compose, the application can be started easily with the following command (in detach mode -> -d):

```
docker compose up -d
```

```
version: '3.6'
services:
  web:
    container_name: <yourcontainername>
    image: 'gitlab/gitlab-ee:latest'
    restart: unless-stopped
    hostname: '<yourgitlabdomain>'
    environment:
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'https://<yourgitlabdomain>'
        gitlab_rails['gitlab_shell_ssh_port'] = <yoursshshellport>
        gitlab_rails['smtp_enable'] = true
        gitlab_rails['smtp_address'] = "<yourmailserver>"
        gitlab_rails['smtp_port'] = 587
        gitlab_rails['smtp_user_name'] = "<yoursmtpmailuser>"
        gitlab_rails['smtp_password'] = "<yoursmtpmailpassword>/"
        gitlab_rails['smtp_domain'] = "<yourgitlabdomain>"
        gitlab_rails['smtp_authentication'] = "login"
        gitlab_rails['smtp_enable_starttls_auto'] = true
        gitlab_rails['omniauth_enabled'] = true
        gitlab_rails['omniauth_allow_single_sign_on'] = ['azure_activedirectory_v2']
        gitlab_rails['omniauth_block_auto_created_users'] = false
        gitlab_rails['omniauth_providers'] = [
          {
            "name" => "azure_activedirectory_v2",
            "label" => "Microsoft Entra ID",
            "args" => {
              "client_id" => "<yourclientid>",
              "client_secret" => "<yourclientsecret>",
              "tenant_id" => "<yourtenantid>",
            }
          }
        ]
    ports:
      - '<yourhttpport>:80'
      - '<yourhttpsport>:443'
      - '<yoursshport>:22'
    volumes:
      - <yourpersistentpath>/config:/etc/gitlab
      - <yourpersistentpath>/logs:/var/log/gitlab
```

```
      - <yourpersistentpath>/data:/var/opt/gitlab
    shm_size: '256m'
```

Revision #2
Created 2024-07-21 12:20:48 UTC
Updated 2025-08-11 15:53:34 UTC by Luca Noah Caprez