**####################################################################################################################################################################################**

**Integrate Redmine with keycloak**

**####################################################################################################################################################################################**

* **Redmine**: Redmine serves as the issue management platform, enabling efficient tracking and handling of transportation-related issue-tickets. It centralizes issue reporting, assignment, and resolution.
* **Superset:** Superset functions as a powerful data visualization tool, providing interactive and insightful dashboards and charts. It aids in transforming raw data into comprehensible visual representations for informed decision-making.
* **Keycloak:** Keycloak plays a pivotal role in identity and access management. It facilitates a single sign-on (SSO) solution, ensuring that users can access both Redmine and Superset with a single set of credentials, enhancing user convenience and security.
* **PostgreSQL:**PostgreSQL serves as the underlying database management system, storing and organizing the data used by Redmine and Superset. It provides a reliable foundation for data storage and retrieval.

*By seamlessly integrating Redmine, Superset, Keycloak, and PostgreSQL, the toolset offers an efficient and comprehensive solution for managing transportation-related issues, visualizing data, streamlining user access, and maintaining a robust database environment.*

# **Why did we set up the keycloak?**

## **Problem:**

In the absence of a Single Sign-On (SSO) solution, users accessing the Redmine issue management tool and the Apache Superset data visualization tool were required to log in separately. This dual authentication process was time-consuming and potentially burdensome, especially for users who frequently accessed both tools.

## **Solution:**

Keycloak: Simplifying User Access with Single Sign-On (SSO)

To alleviate the inconvenience of separate logins and streamline the user experience, a Single Sign-On solution called Keycloak was introduced. Keycloak acts as an intermediary between Redmine and Superset, offering users a unified authentication process.

## **Benefits:**

* Unified User Authentication: With Keycloak, users need to authenticate only once. Once logged in, they gain seamless access to both the Redmine and Superset platforms, eliminating the need for redundant logins.
* Enhanced User Experience: The convenience of SSO significantly enhances user experience, reducing friction in the authentication process and improving user satisfaction.
* Efficiency and Productivity: Users can seamlessly transition between Redmine and Superset without interruption. This efficiency boost leads to enhanced productivity and smoother workflow.

**Environment :-**

Redmine version 4.0.5.stable

Keycloak 22.0.3

PostgreSQL 16.0

Superset

**OS Details :-**

Ubuntu 20.04

**If you have not podman installed , then follow the below command**

| **# echo "deb https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu\_20.04/ /" | sudo tee /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list   # curl -L"https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu\_20.04/Release.key" | sudo apt-key add -  # sudo apt update # sudo apt install podman** |
| --- |

A Dockerfile is a script used to create a Docker image. Docker containers are instances of Docker images, and they run in isolated environments on a host system. We are going to create a Redmine Dockerfile, build it, and then run the containers

**First we have to create Dockerfile for redmine**

**Touch Dockerfile:** is generally used for creating a file

**Vim Dockerfile:** vim is generally used for editing inside the file

**Cat Dockerfile:** cat is used for seeing the content inside the file

| **manoj@crunchy-postgres:~$ cat Dockerfile # syntax=docker/dockerfile:1  FROM docker.io/library/redmine:4.0.5 # Switching to root to install the required packages WORKDIR /usr/src/redmine/plugins RUN git clone https://github.com/onozaty/redmine-view-customize.git view\_customize RUN git clone https://github.com/devopskube/redmine\_openid\_connect.git  WORKDIR /usr/src/redmine/ RUN bundle install --without development test** |
| --- |

**Now build the images here with below command**

| **manoj@crunchy-postgres:~$ podman build -t redmine:manoj . STEP 1/6: FROM docker.io/library/redmine:4.0.5 STEP 2/6: WORKDIR /usr/src/redmine/plugins --> Using cache 0f62ab8008e7798deb7dfe5934edfde686ea2b51fd2eccfa5a7b4301d49f5efe --> 0f62ab8008e STEP 3/6: RUN git clone https://github.com/onozaty/redmine-view-customize.git view\_customize --> Using cache 73c2576456d2def7ea16bd0b0352667edc7564f99301a0db97caaa6354d1bea9 --> 73c2576456d STEP 4/6: RUN git clone https://github.com/devopskube/redmine\_openid\_connect.git --> Using cache 43bbfa8dc026e93b96477ba51d2f963bbda7a9526e29e62184fd3d0cfc32f47c --> 43bbfa8dc02 STEP 5/6: WORKDIR /usr/src/redmine/ --> Using cache 018c6ed3a78b3a1446a9c682276be6d5b83d0db99e4e9fe3333c02a2883ea807 --> 018c6ed3a78 STEP 6/6: RUN bundle install --without development test --> Using cache 36f8102faa89c459c9621e2a8dbde6b7337426684a0fcdb855fde0cc54cff0f5 COMMIT redmine:manoj --> 36f8102faa8 Successfully tagged localhost/redmine:manoj Successfully tagged localhost/redmine\_custom:1.0 36f8102faa89c459c9621e2a8dbde6b7337426684a0fcdb855fde0cc54cff0f5** |
| --- |

**Check the build images are present or not**

| **manoj@crunchy-postgres:~$ podman images REPOSITORY TAG IMAGE ID CREATED SIZE localhost/superset\_apache mj 37735fa096a4 4 days ago 1.33 GB localhost/redmine\_custom 1.0 36f8102faa89 5 days ago 553 MB localhost/redmine manoj 36f8102faa89 5 days ago 553 MB** |
| --- |

**I have run a Redmine container, built from the created image, and connected it to a PostgreSQL database.**

**Now we are creating a file of keycloak.sh**

**manoj@crunchy-postgres:~$ touch keycloak.sh**

**Copy the below bash script inside the keycloak.sh file**

**For checking the inside content we can use the command of cat and file name**

**manoj@crunchy-postgres:~$ cat keycloak.sh**

| **#!/bin/bash**  **# Create a folder for PostgreSQL data echo "Creating folder for PostgreSQL data" mkdir -p /home/manoj/data/redmine/postgres  # Change ownership of the data folder echo "Changing ownership of the data folder" podman unshare chown 999:999 /home/manoj/data/redmine/postgres  # Create a Podman pod for Redmine, Keycloak, and Superset echo "Creating the Redmine pod" podman pod create --name redmine \  --publish 3000:3000 \  --publish 5432:5432 \  # Set up PostgreSQL container echo "Running the PostgreSQL container" podman run -dt \  --pod redmine \  --name redmine-postgres \  -e POSTGRES\_DB=redmine \  -e POSTGRES\_USER=postgres \  -e POSTGRES\_PASSWORD=Password \  -e POSTGRES\_HOST\_AUTH\_METHOD=trust \  -e PGDATA=/var/lib/postgresql/data/pgdata \  -v /home/manoj/data/redmine/postgres:/var/lib/postgresql/data \  docker.io/postgres:latest  # Set up Redmine container echo "Running the Redmine container" podman run -dt \  --pod redmine \  --name redmine-app \  -e REDMINE\_DB\_POSTGRES=127.0.0.1 \  -e REDMINE\_DB\_PORT=5432 \  -e REDMINE\_DB\_DATABASE=redmine \  -e REDMINE\_DB\_USERNAME=postgres \  -e REDMINE\_DB\_PASSWORD=password \  redmine:manoj** |
| --- |

**Give execute permission to above script by +x filename**

**#chmod +x keycloak.sh**

**After giving execute permission Run below command for execute the script**

**#sh -x keycloak.sh or bash keycloak.sh ( choose anyway to execute this script )**

| **manoj@crunchy-postgres:~$ podman exec -it redmine-app bash root@b17c47e5f17a:/usr/src/redmine# bundle install The dependency tzinfo-data (>= 0) will be unused by any of the platforms Bundler is installing for. Bundler is installing for ruby but the dependency is only for x86-mingw32, x64-mingw32, x86-mswin32. To add those platforms to the bundle, run `bundle lock --add-platform x86-mingw32 x64-mingw32 x86-mswin32`. Using rake 13.0.1 Using concurrent-ruby 1.1.5 Using i18n 0.7.0 Using sqlite3 1.3.13 Bundle complete! 29 Gemfile dependencies, 61 gems now installed. Gems in the groups development and test were not installed. Bundled gems are installed into `/usr/local/bundle` root@b17c47e5f17a:/usr/src/redmine#  root@b17c47e5f17a:/usr/src/redmine# cd plugins/ root@b17c47e5f17a:/usr/src/redmine/plugins# ls README redmine\_openid\_connect view\_customize root@b17c47e5f17a:/usr/src/redmine/plugins# bundle install The dependency tzinfo-data (>= 0) will be unused by any of the platforms Bundler is installing for. Bundler is installing for ruby but the dependency is only for x86-mingw32, x64-mingw32, x86-mswin32. To add those platforms to the bundle, run `bundle lock --add-platform x86-mingw32 x64-mingw32 x86-mswin32`. Using rake 13.0.1 Using concurrent-ruby 1.1.5 Using i18n 0.7.0 Using minitest 5.13.0   root@b17c47e5f17a:/usr/src/redmine/plugins# cd redmine\_openid\_connect/**  **root@b17c47e5f17a:/usr/src/redmine/plugins/redmine\_openid\_connect# bundle install Fetching gem metadata from https://rubygems.org/........... Resolving dependencies... Using bundler 1.17.2 Using multi\_xml 0.6.0 Using httparty 0.14.0 Bundle complete! 1 Gemfile dependency, 3 gems now installed. Gems in the groups development and test were not installed. Bundled gems are installed into `/usr/local/bundle` root@b17c47e5f17a:/usr/src/redmine/plugins/redmine\_openid\_connect#  root@b17c47e5f17a:/usr/src/redmine/plugins/view\_customize# bundle install Resolving dependencies... Using i18n 0.7.0 Using minitest 5.13.0 Using activerecord-compatible\_legacy\_migration 0.1.2 Bundle complete! 1 Gemfile dependency, 11 gems now installed. Gems in the groups development and test were not installed. Bundled gems are installed into `/usr/local/bundle`**  **root@b17c47e5f17a:/usr/src/redmine/plugins/view\_customize# #run this command in last then restart root@b17c47e5f17a:/usr/src/redmine/plugins/view\_customize#**  **####### AFter restart the container you have to run below command #########**  **manoj@crunchy-postgres:~$ podman restart redmine-app**  **manoj@crunchy-postgres:~$ podman exec -it redmine-app bash root@b17c47e5f17a:/usr/src/redmine# bundle exec rake redmine:plugins:migrate RAILS\_ENV=production  manoj@crunchy-postgres:~$ podman restart redmine-app b17c47e5f17a94bdbaab3ced7dea0c9c5c3368893c5463957af75e94a7b8b8c7** |
| --- |

**Creating Keycloak container**

| **manoj@crunchy-postgres:~$ podman run -d --name keycloak -p 8080:8080 -e KEYCLOAK\_ADMIN=admin -e KEYCLOAK\_ADMIN\_PASSWORD=admin quay.io/keycloak/keycloak:22.0.3 start-dev** |
| --- |

**##########################################################################################**

**Create a REALM role as needed !!**

**6 - Keycloak Configuration for Redmine and Superset -**

**Note for SS0 to work properly create the same client for redmine and superset.**

**Create a client scope named openid in Clinet\_Scopes -**

**Add the following mappers -**

**username -- of type user property -- token claim name should be - user\_name**

**member\_of -- of type user realm role -- token claim name should be member\_of**

**roles --- of type user realm role -- token claim name should be roles**

**add all these mappers to id\_token , userinfo and access\_token.**

**5 - Open keycloak Admin ui and create a REALM role Admin for Admin user in superset and redmine . User for Normal user in redmine .**

**User Role Supported By Redmine -**

**Admin**

**User**

**User Role Supported By Superset -**

**Admin**

**Alpha**

**Gamma**

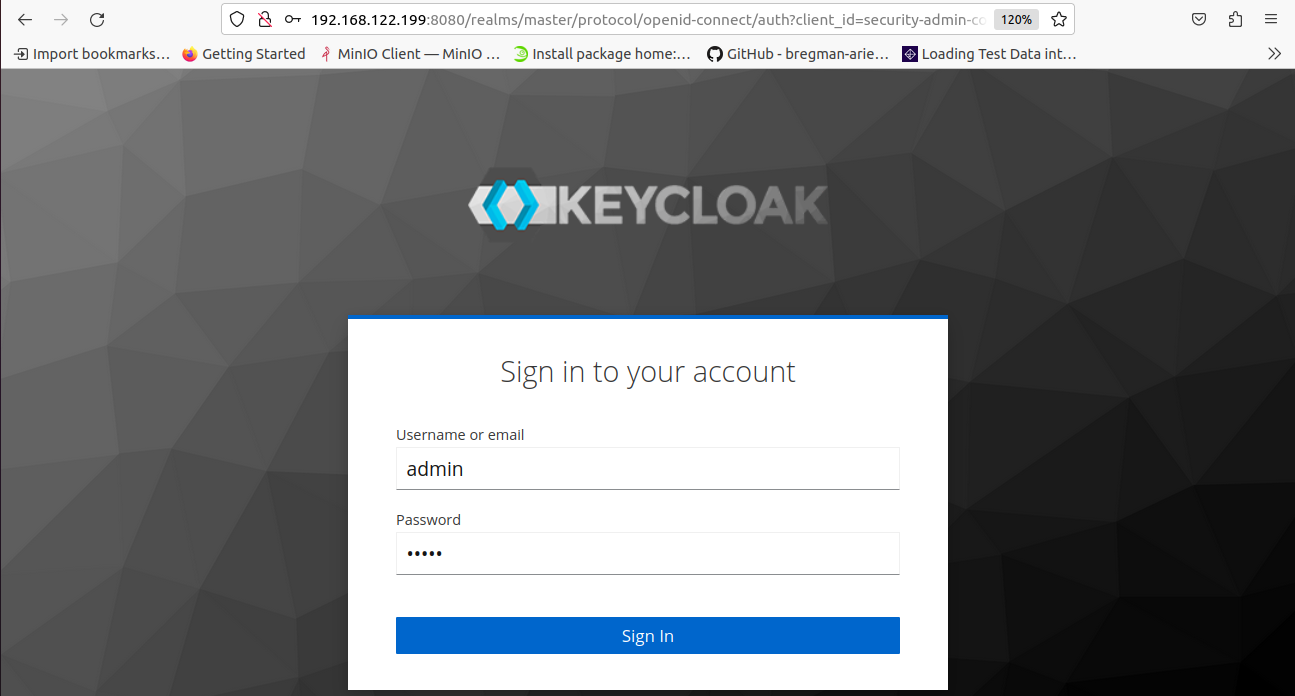
**Public**

**granter**

**sql\_lab**

**##########################################################################################**

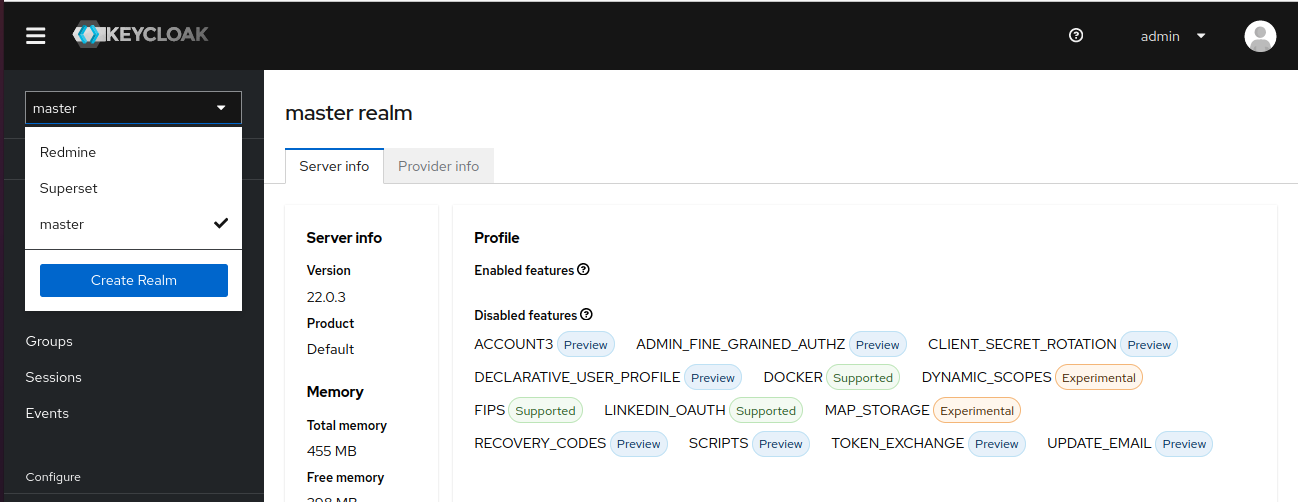
**Hit on browsers http://192.168.122.199:8080**

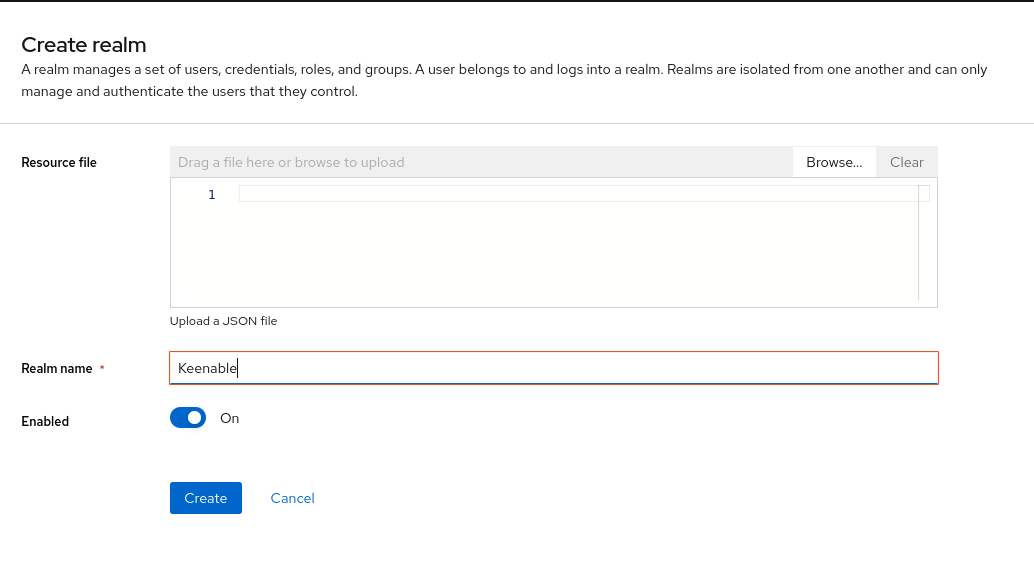
****

**Go to left side after sign in**

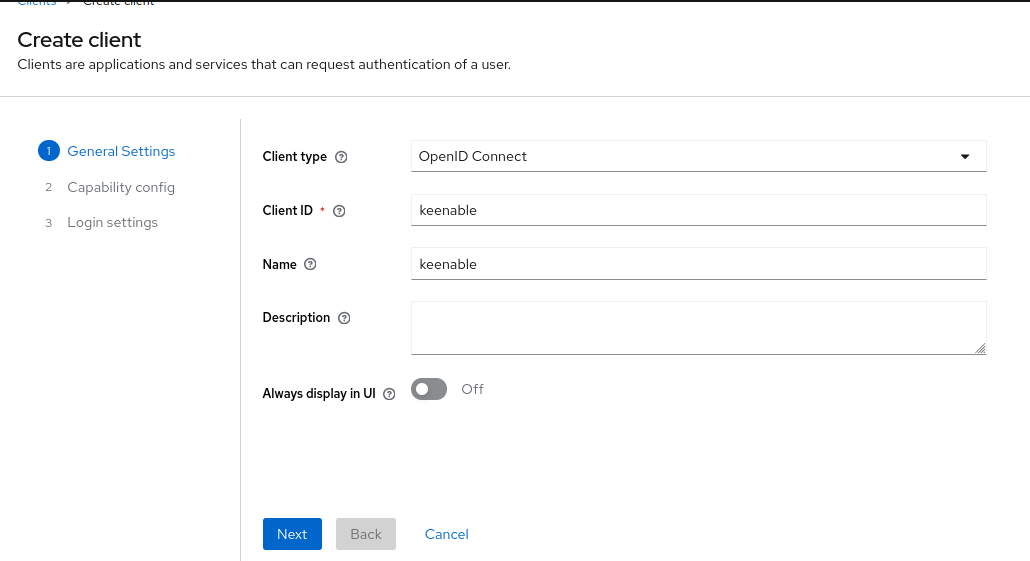
**Create realm >Keenable**

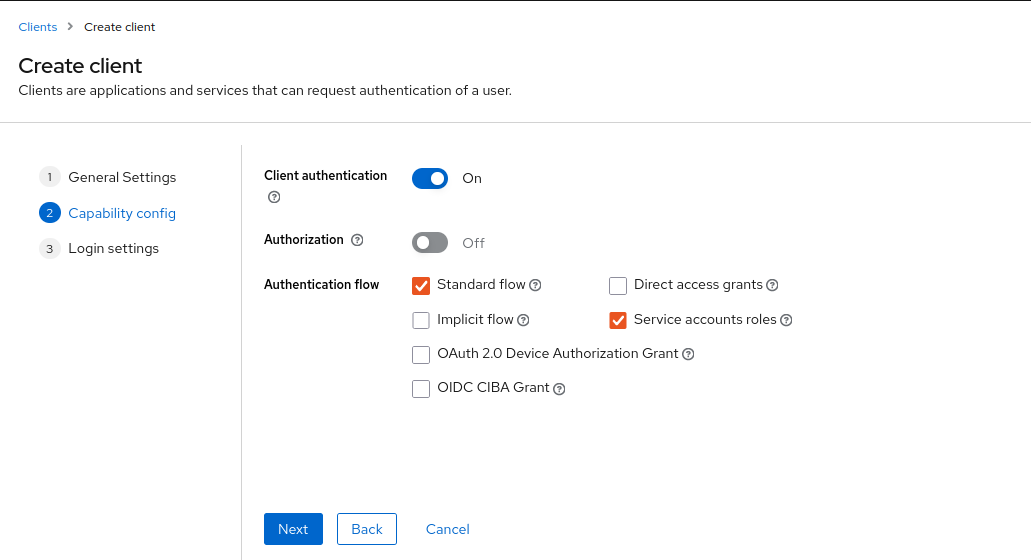
**And client name>keenable**

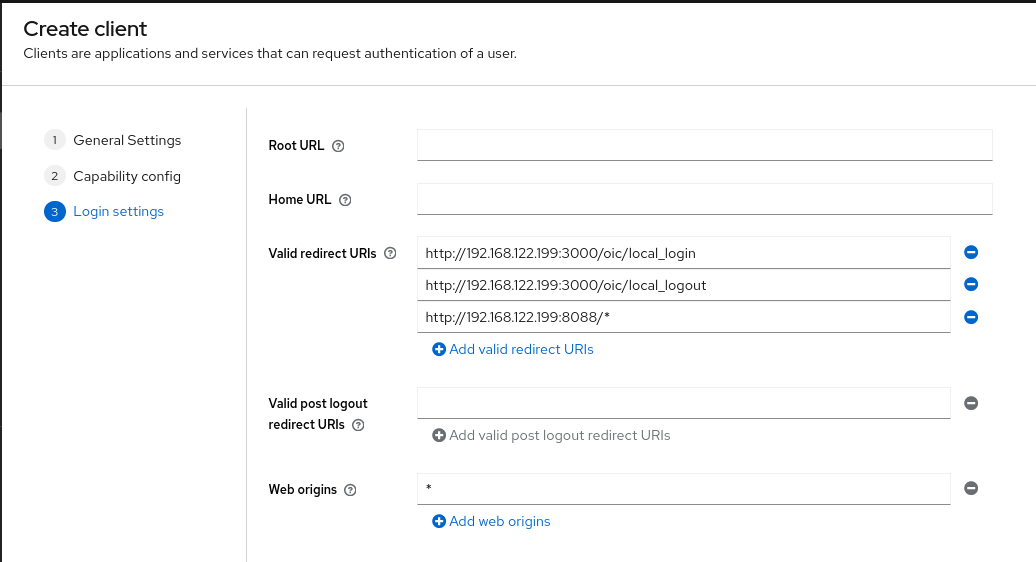
****

****

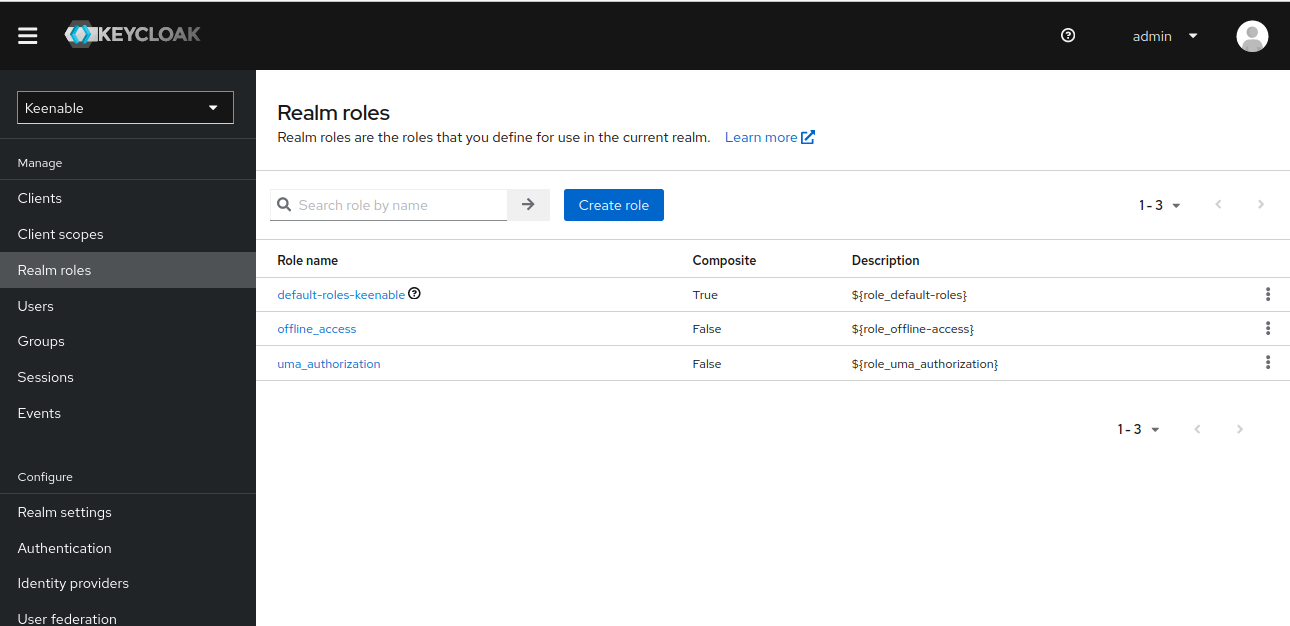
**Go to client section and create client name keenable**

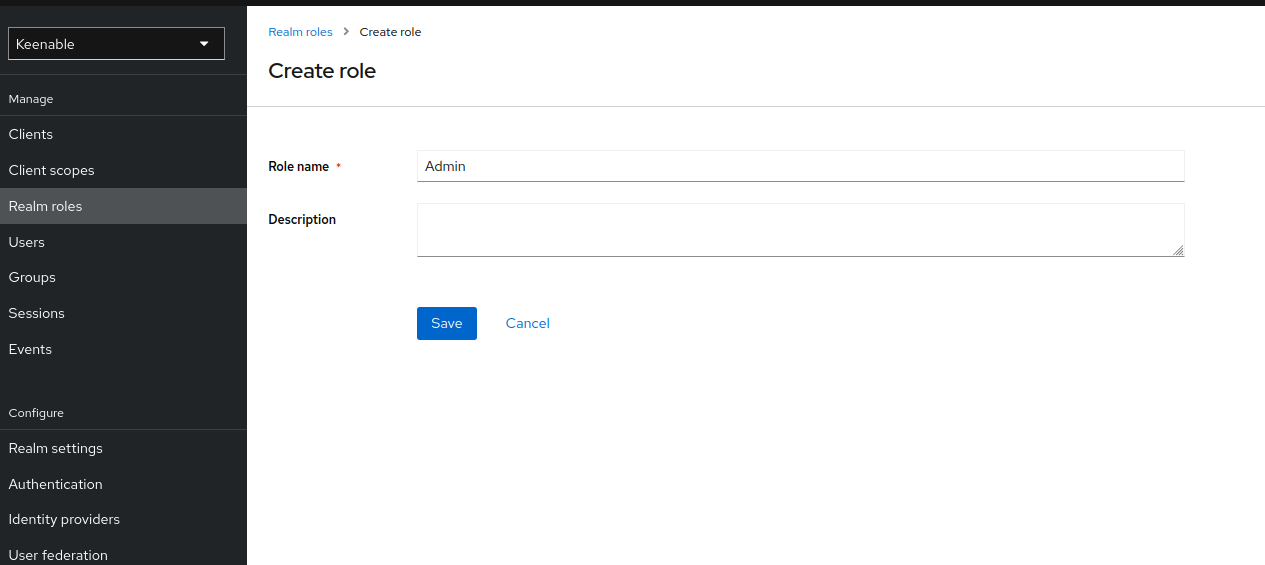
****

****

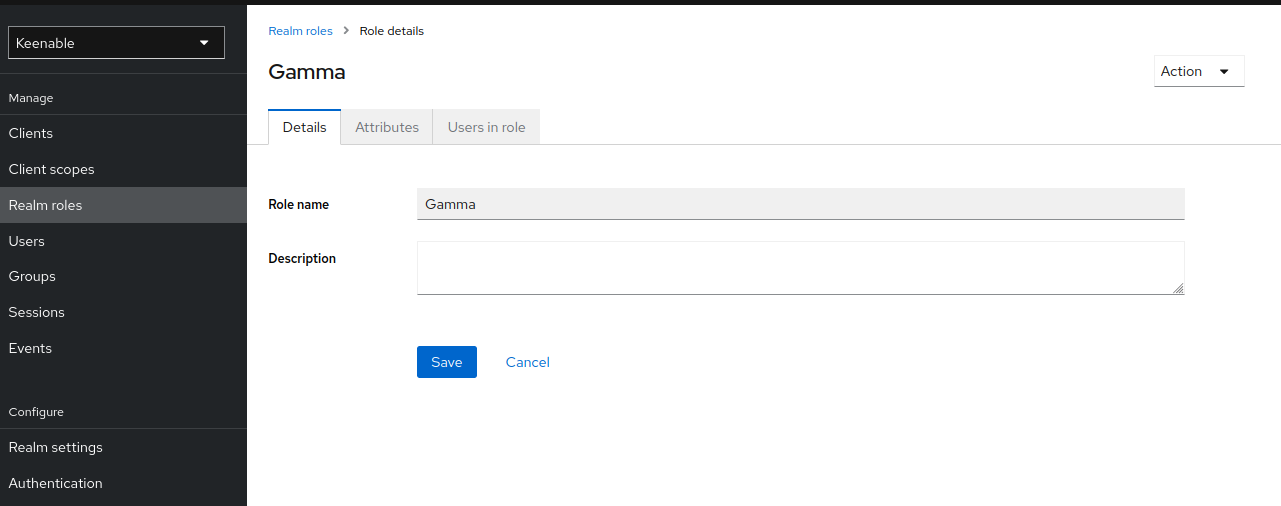
****

**Go to the realm roles section**

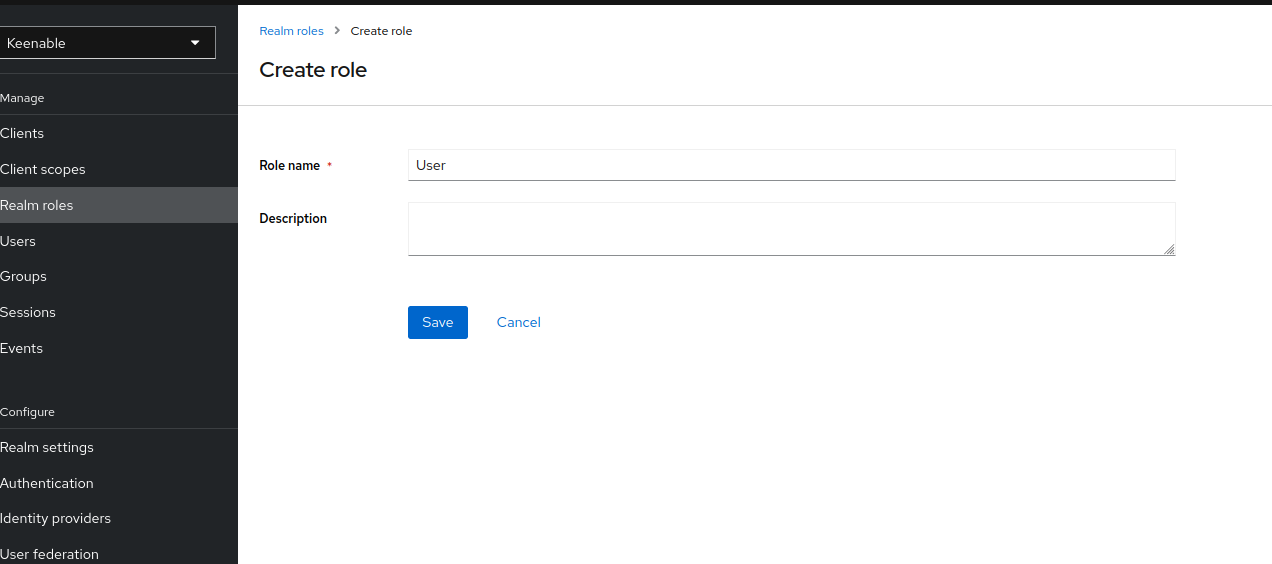
****

****

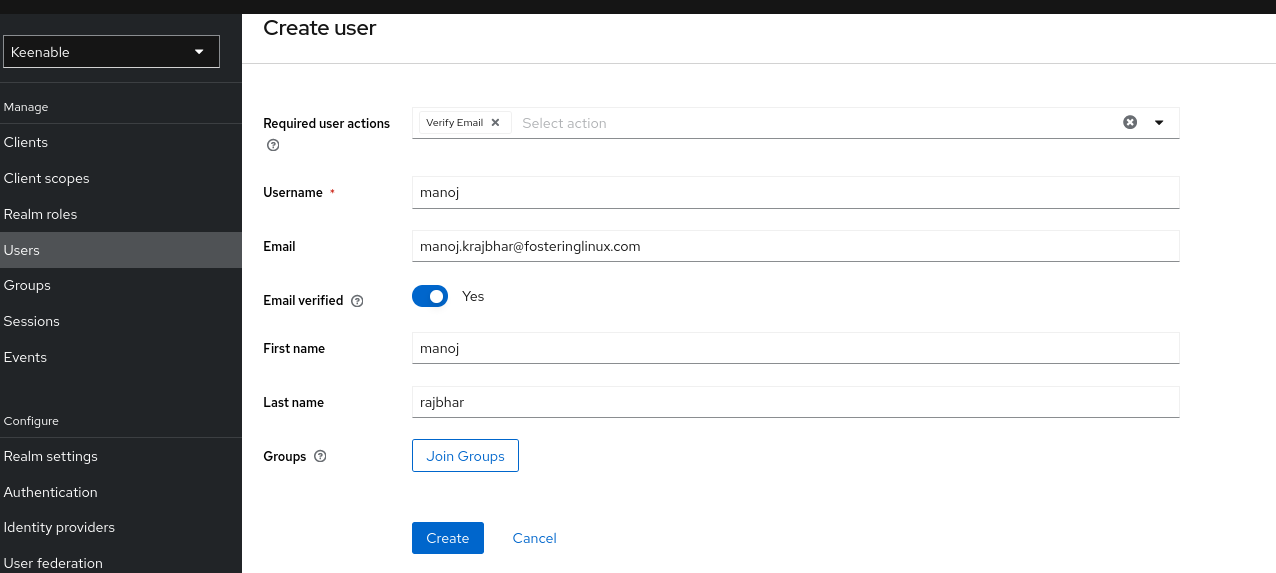
**Also Creating Gamma Role**

****

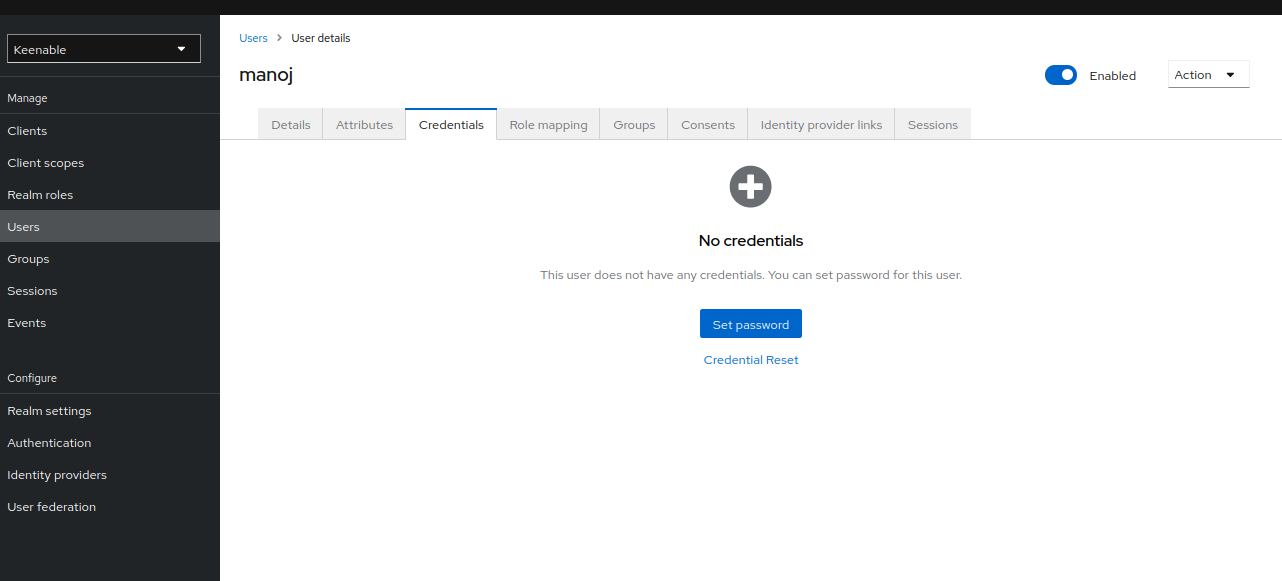
**Creating User Role**

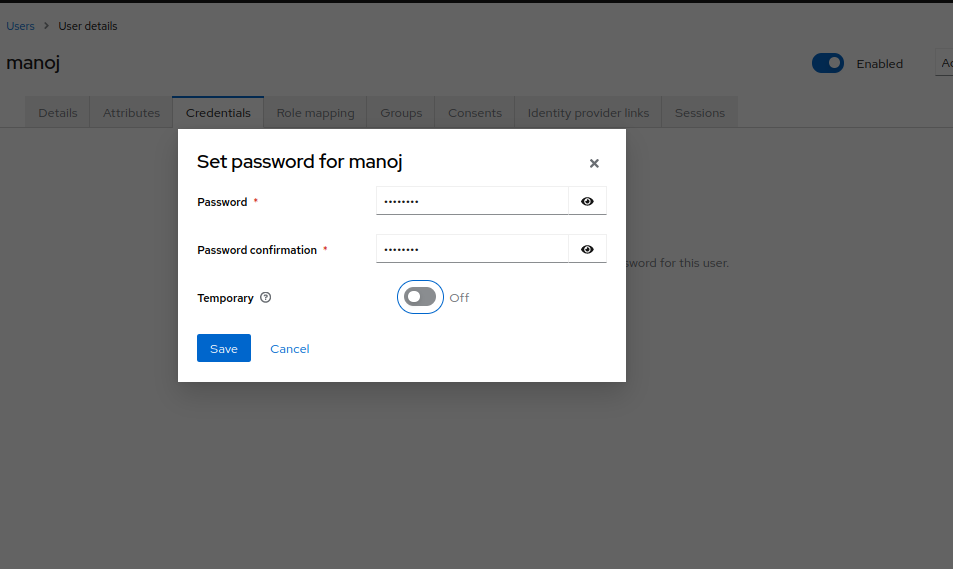
****

**User creating**

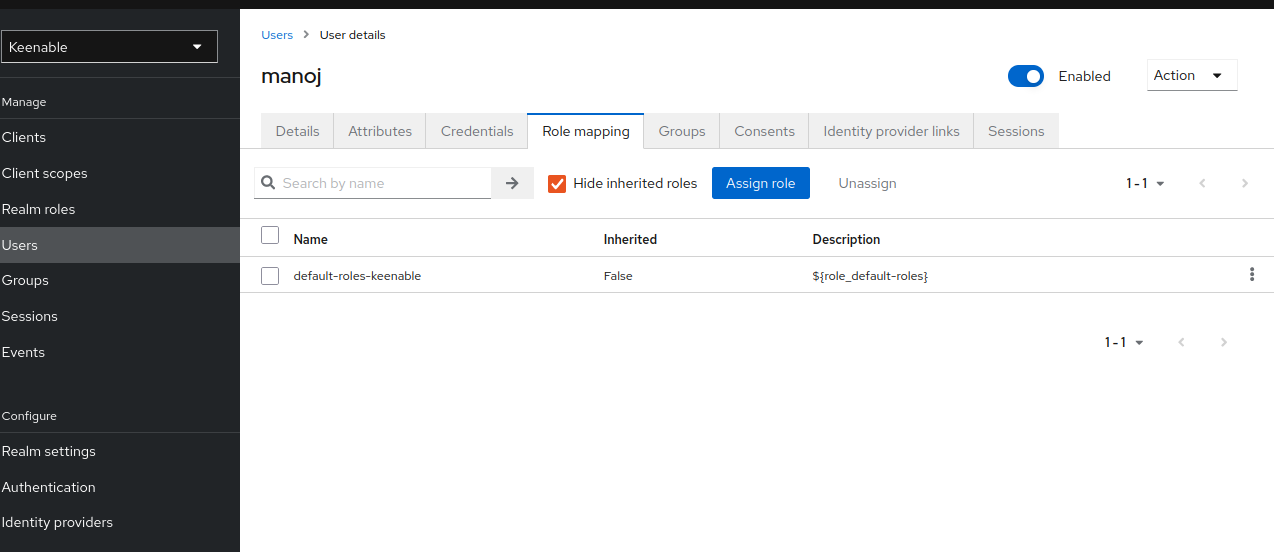
****

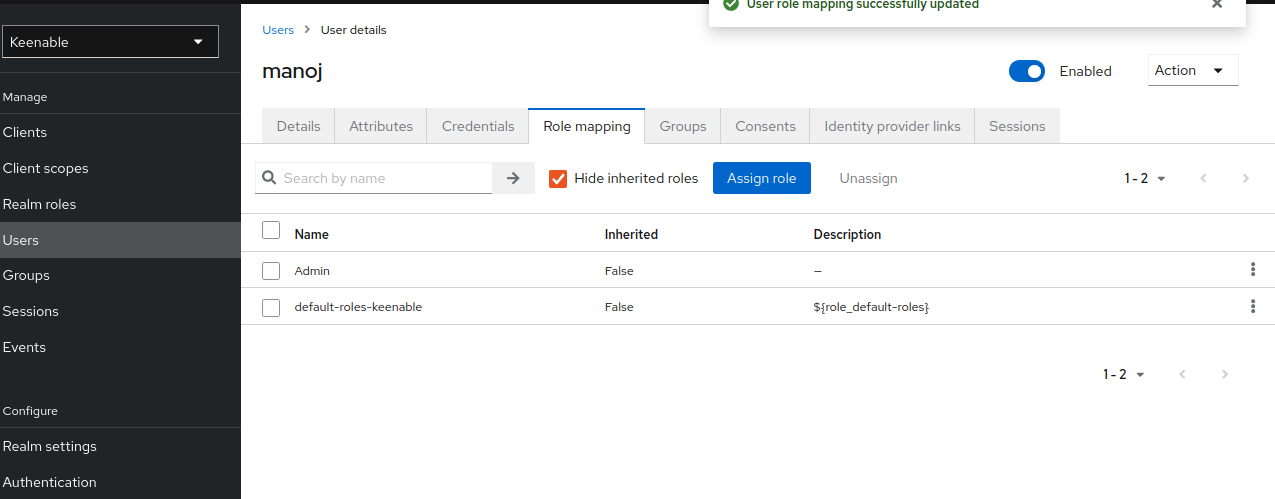
**Also set password for every user**

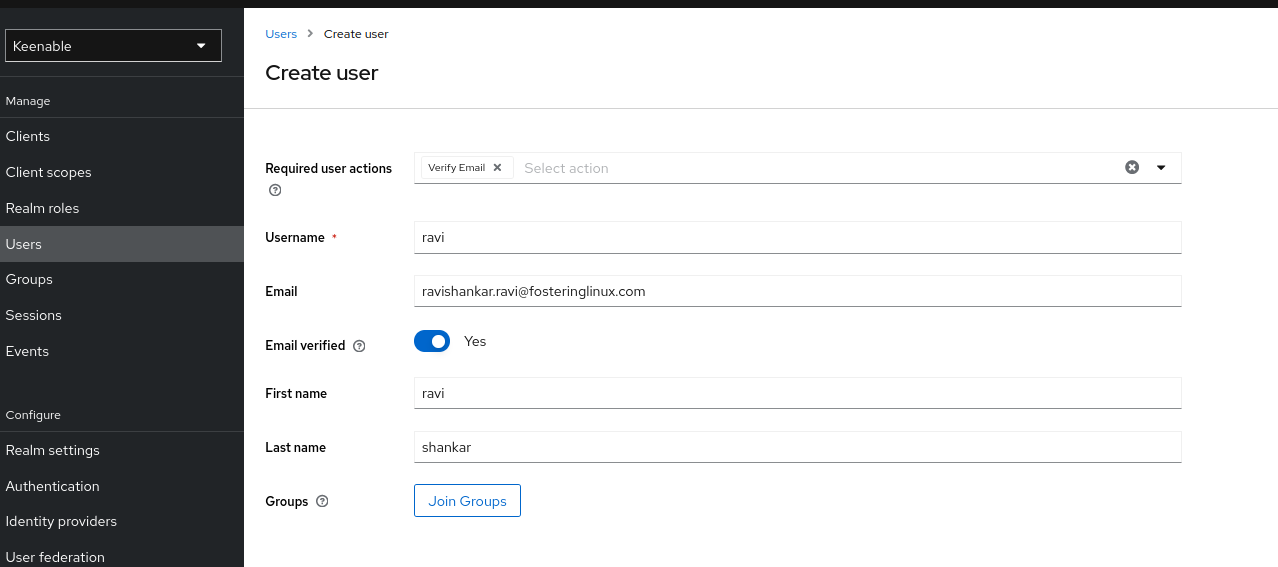
****

****

**Assign the role to given user**

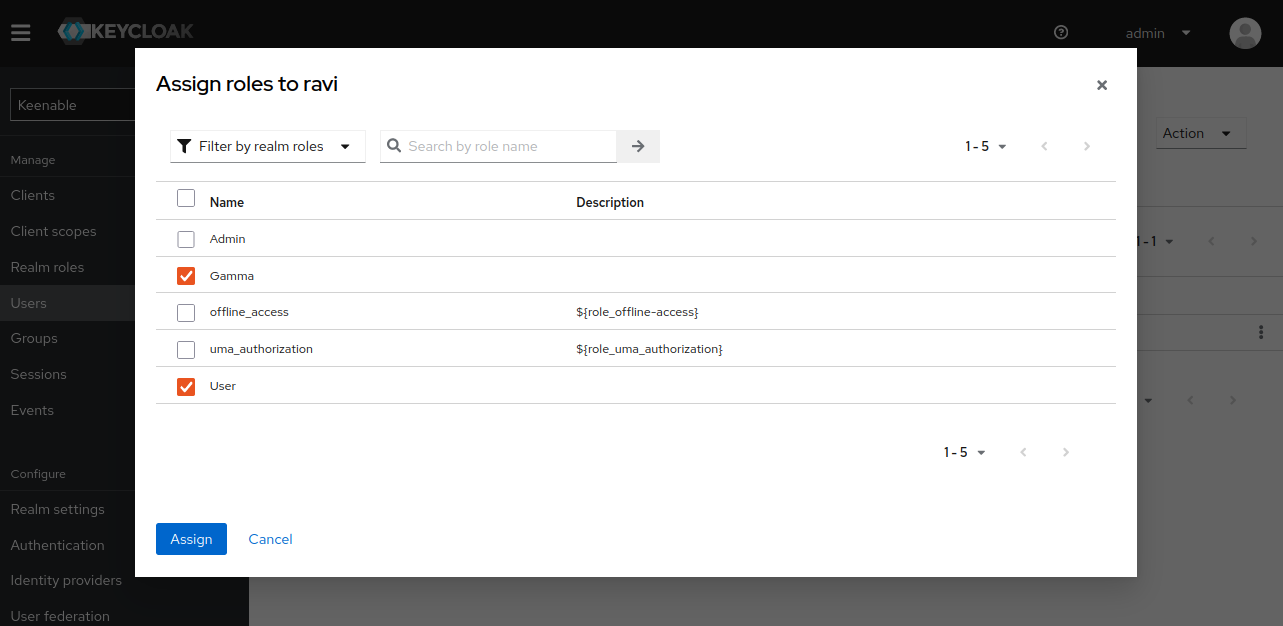
****

****

****

**And now give it roles**

**Here ravi have two roles one is gamma and another is User role**

****

**Create a client scope named openid in Clinet\_Scopes -**

**Add the following mappers -**

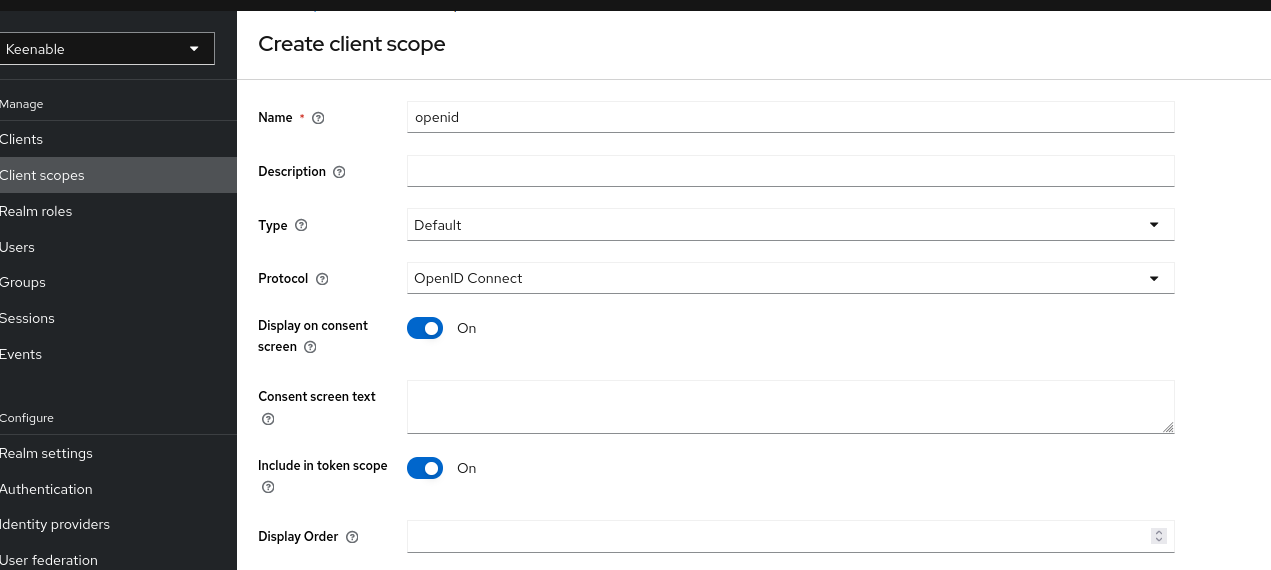
**username -- of type user property -- token claim name should be - user\_name**

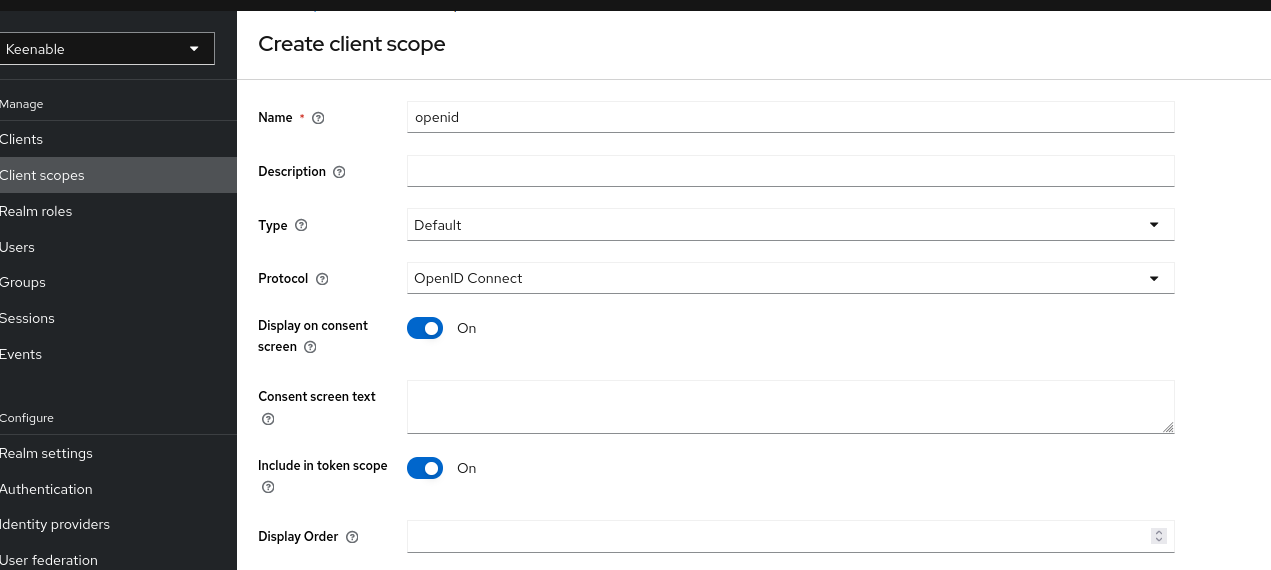
**member\_of -- of type user realm role -- token claim name should be member\_of**

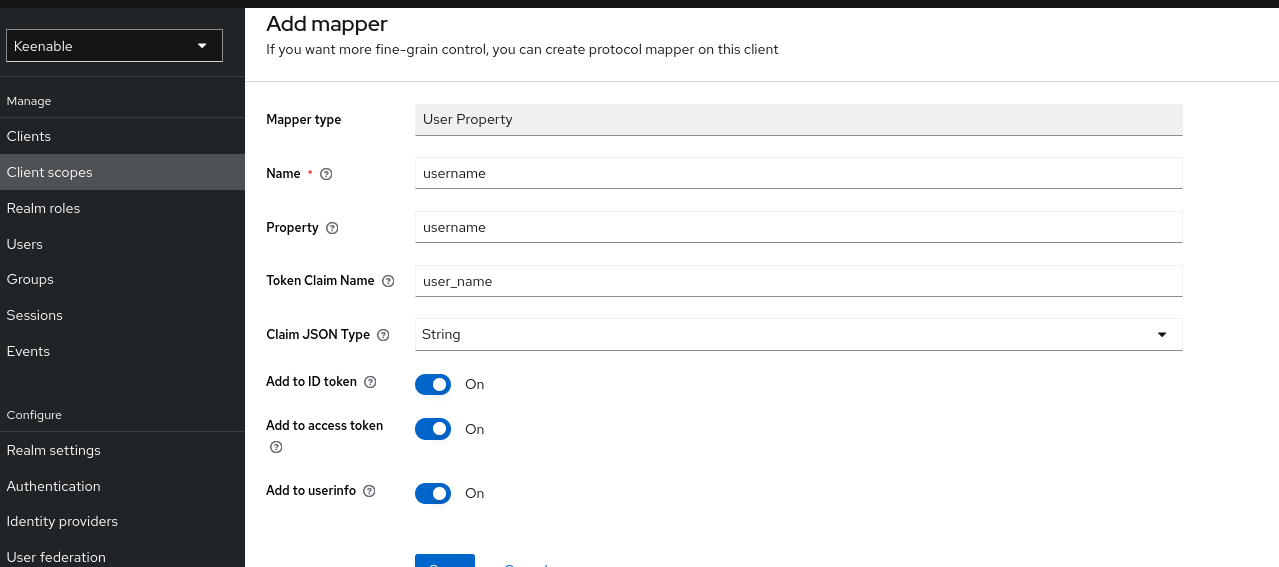
**roles --- of type user realm role -- token claim name should be roles**

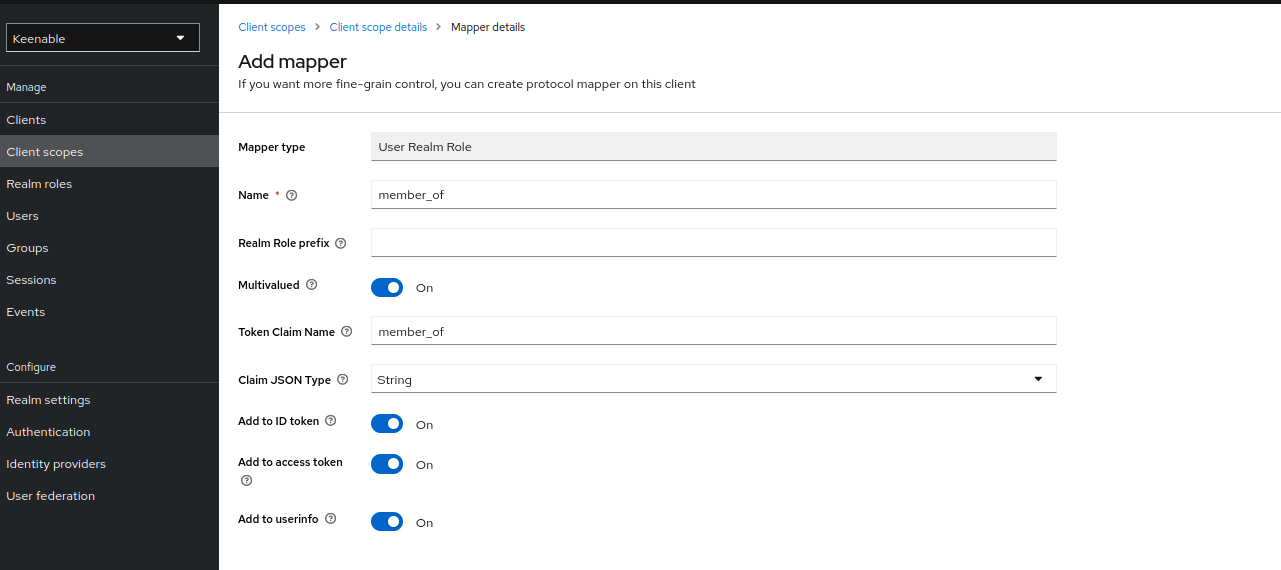
**add all these mappers to id\_token , userinfo and access\_token.**

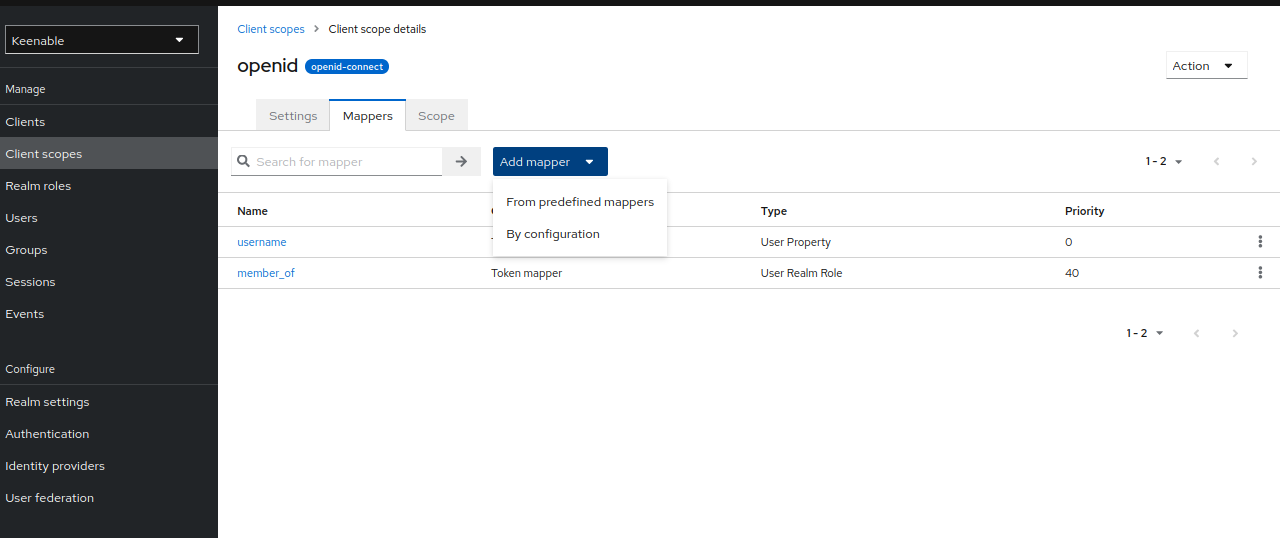
**NOW create client scop named= openid**

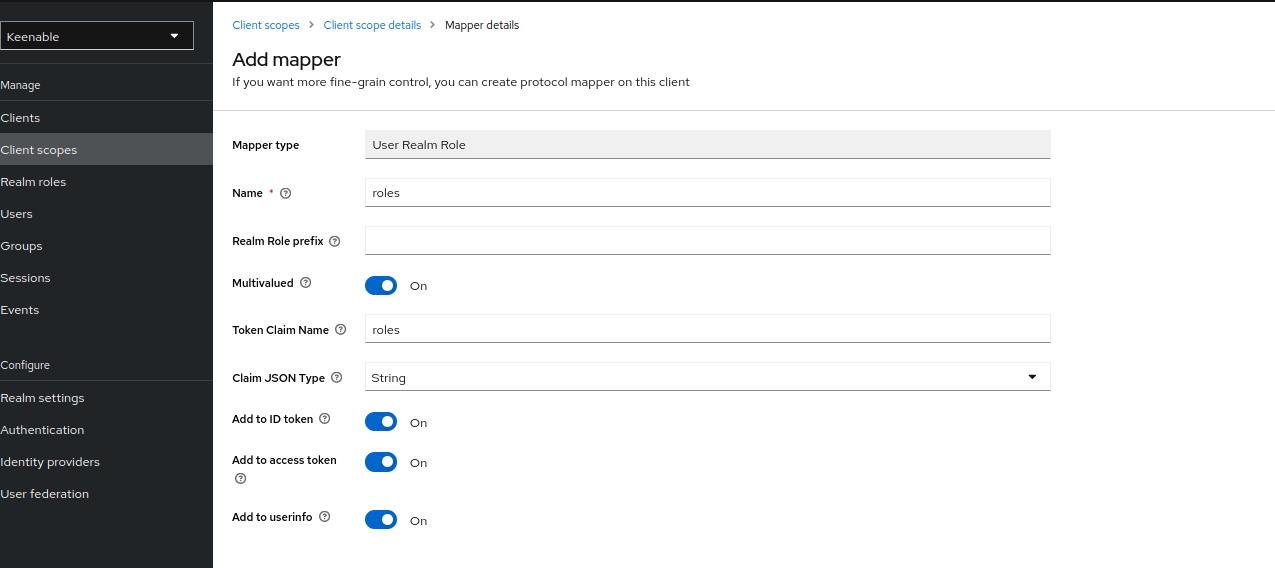
****

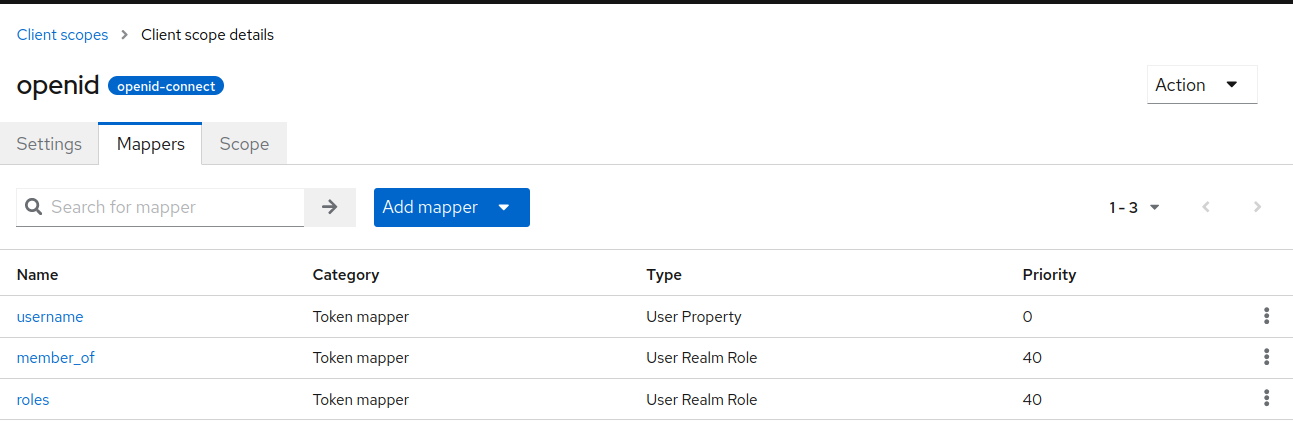
****

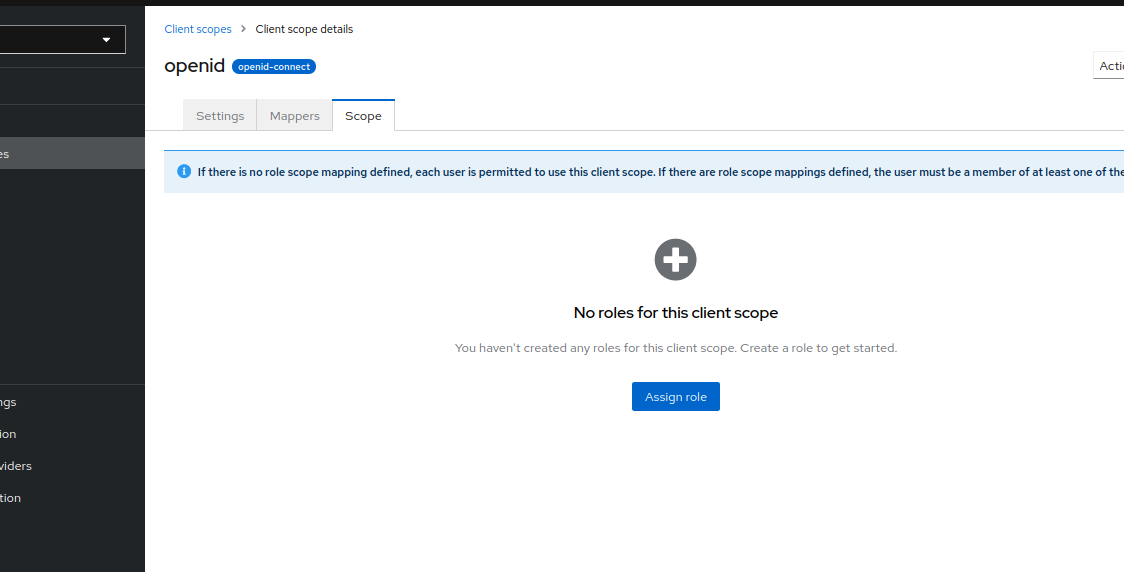
****

****

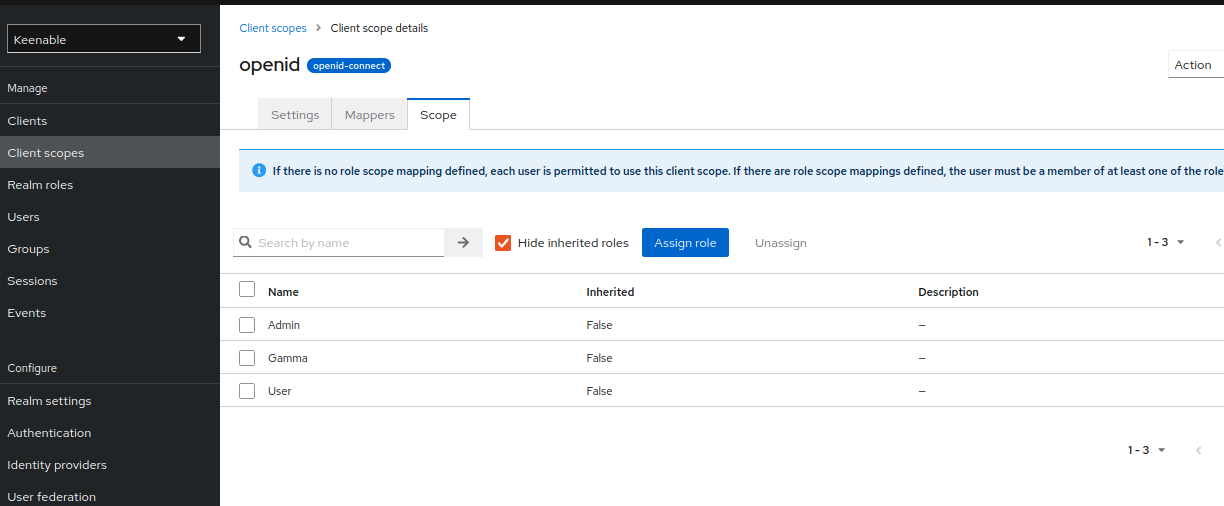
****

****

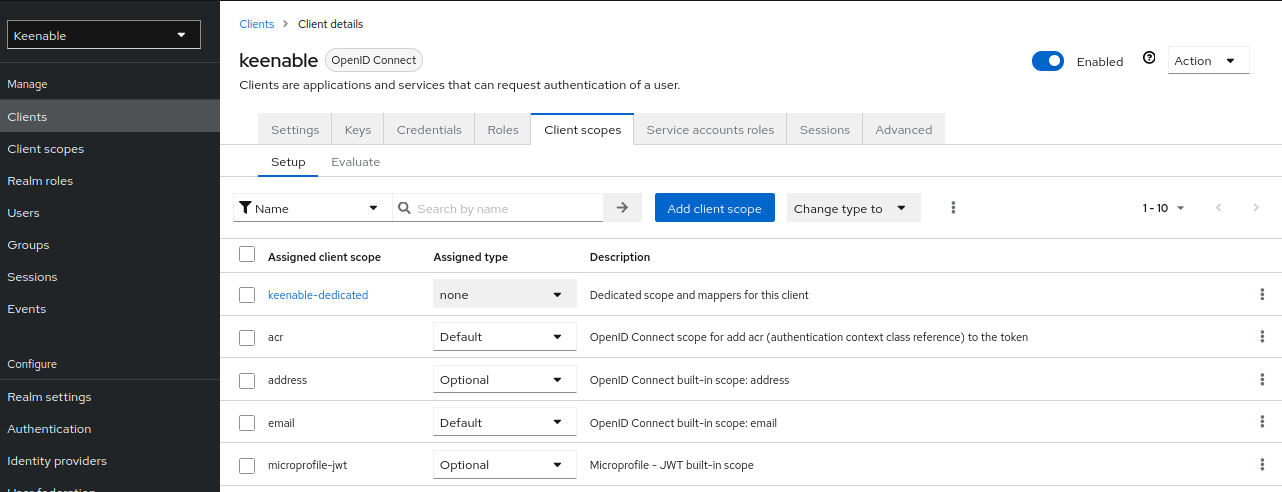
**Now we are total 3 mapper**

****

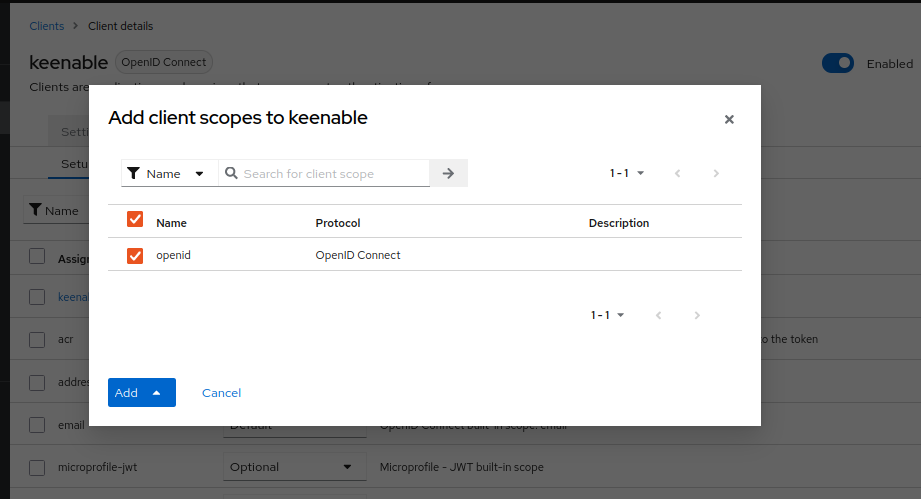
**Now assign the roles to scope**

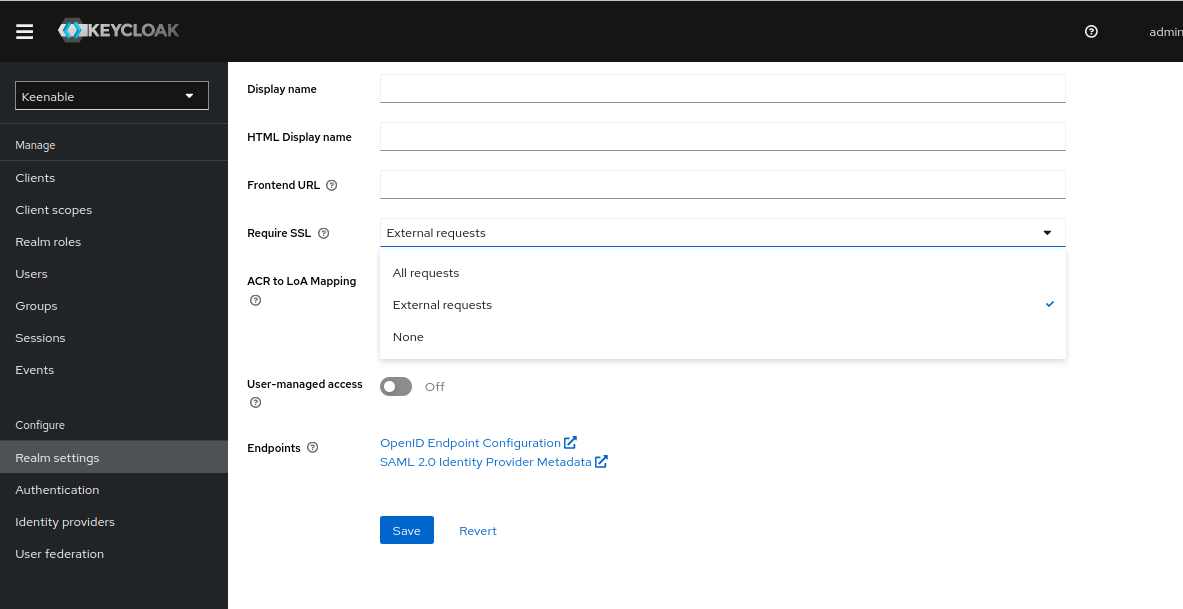
****

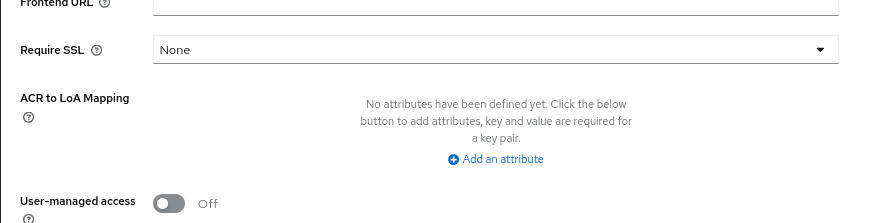
**Now go to client section search for**

****

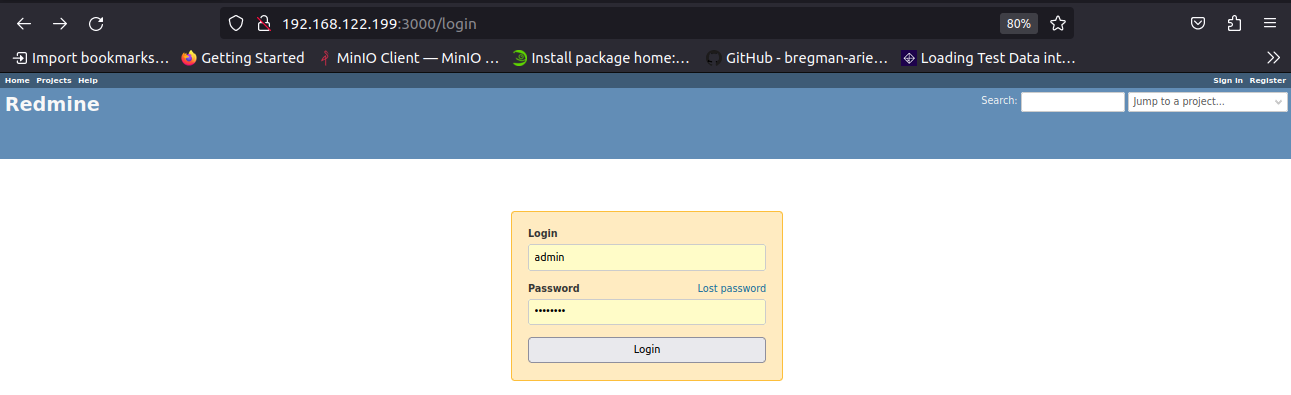
**Add client scope name open id as we previously created**

****

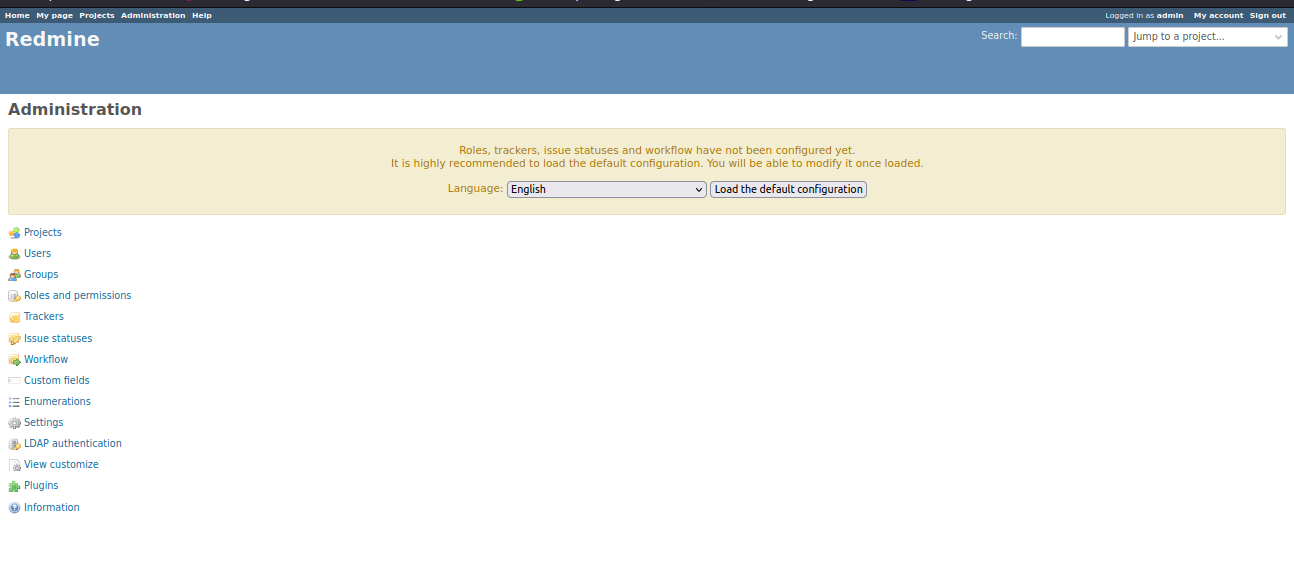
**Go to Realm setting section and disable the ssl  
  
**

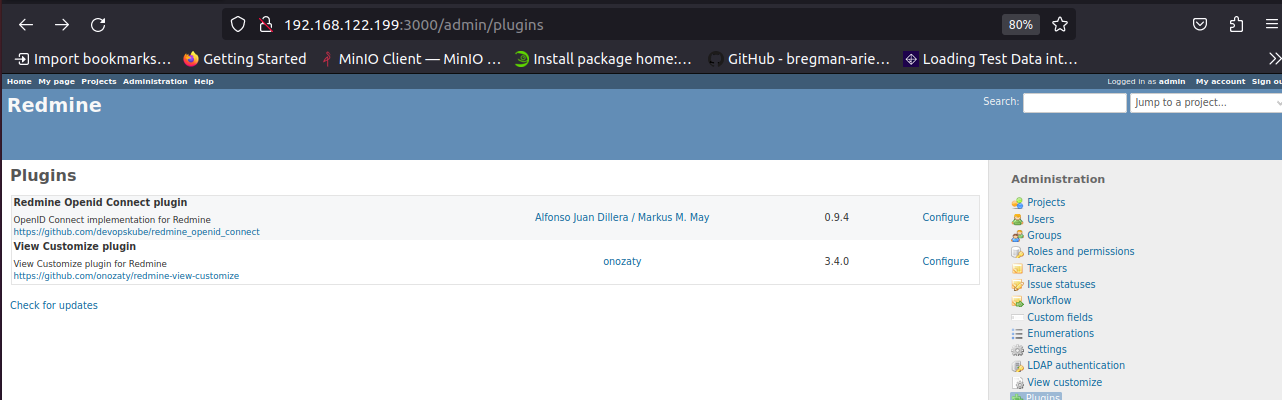
****

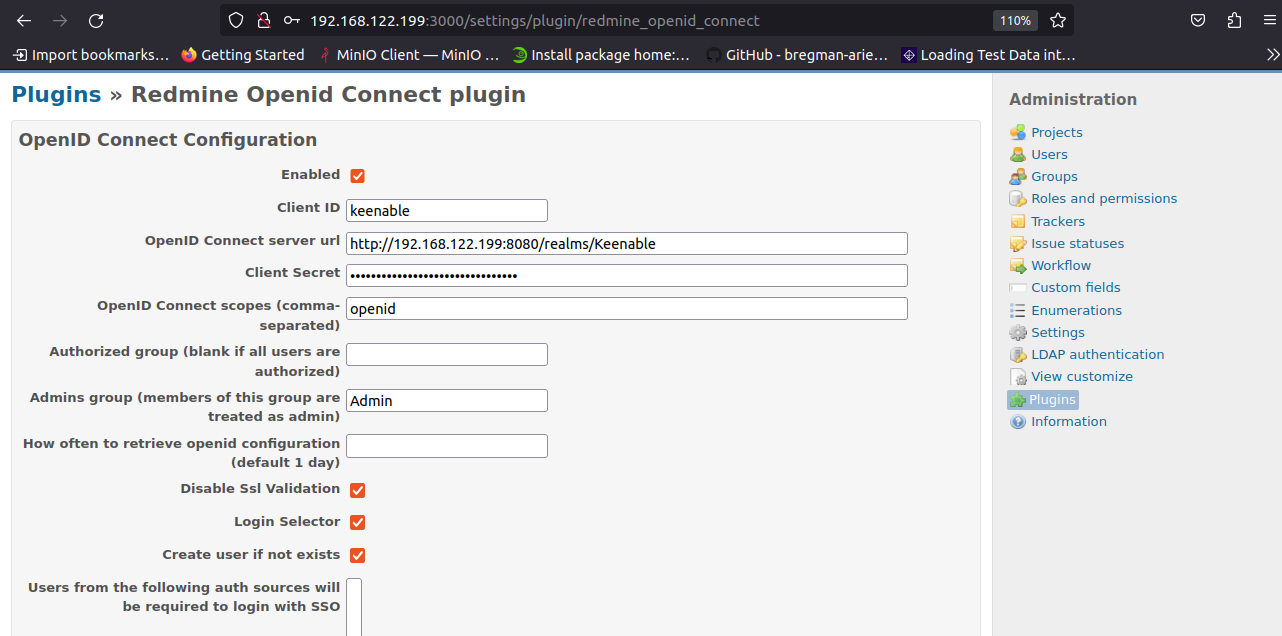
**Now login in redmine first**

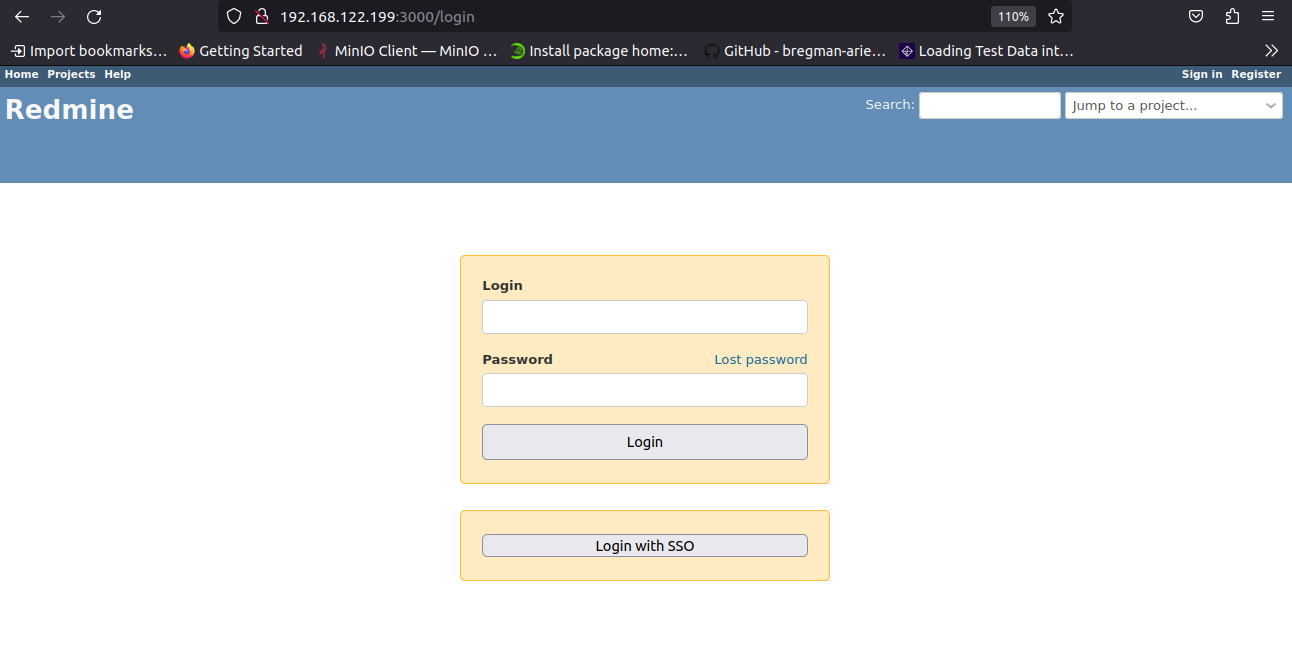
****

**Now go for plugin section**

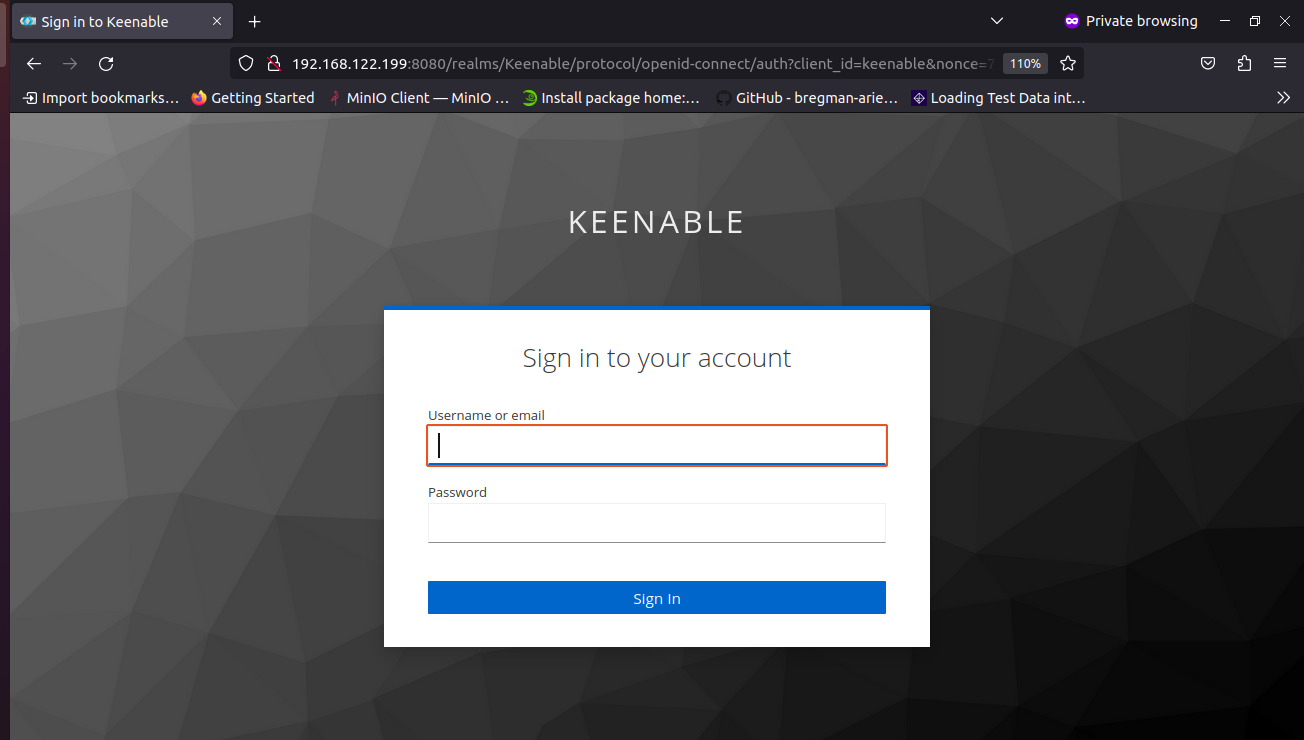
****

****

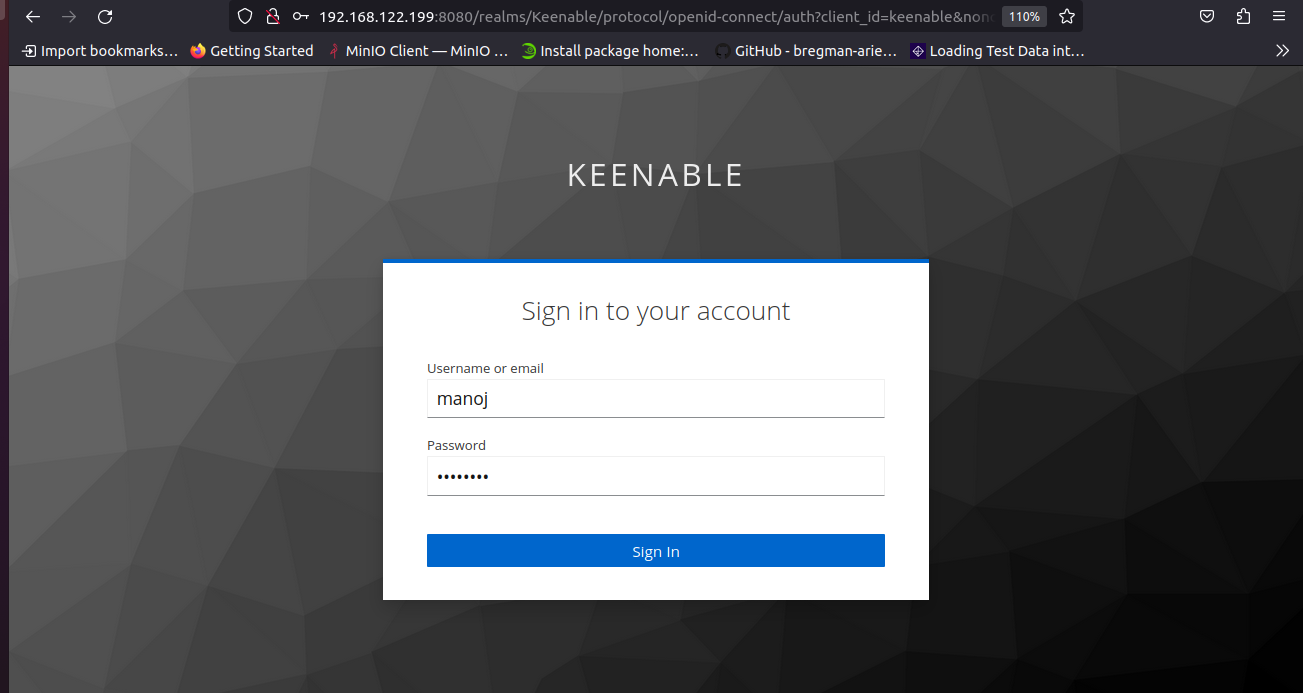
****

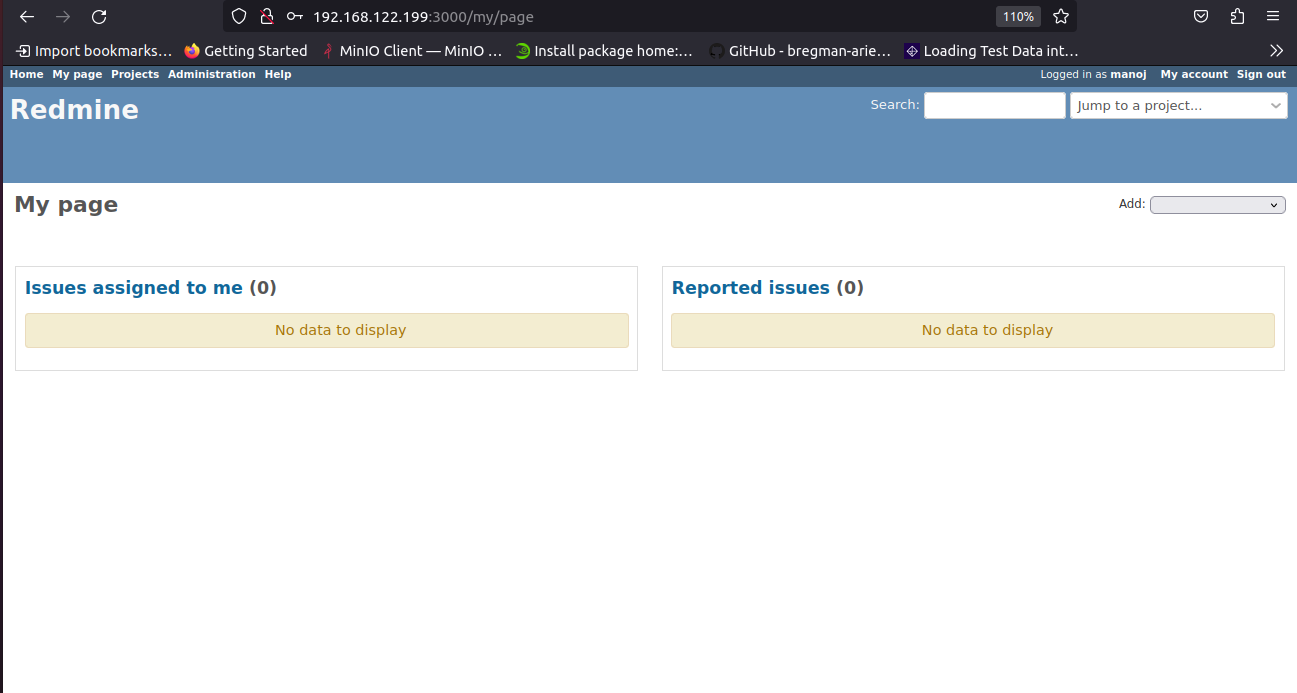
****

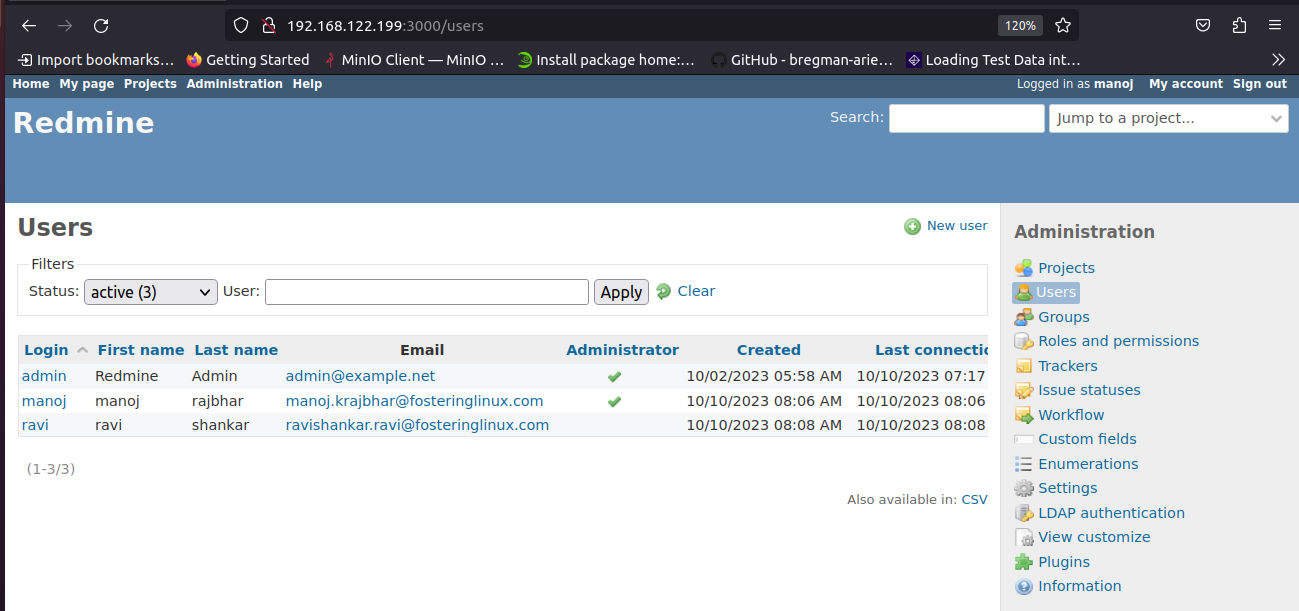
**When i Click the single sign On**

****

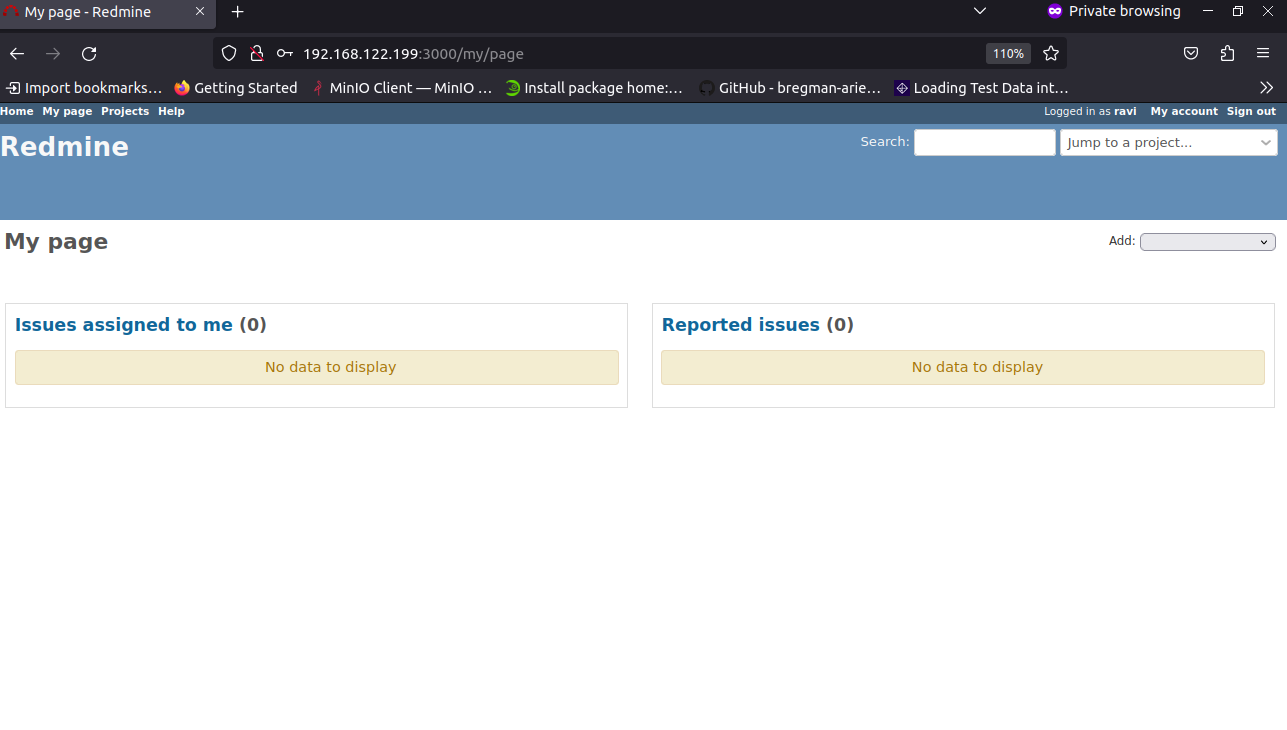
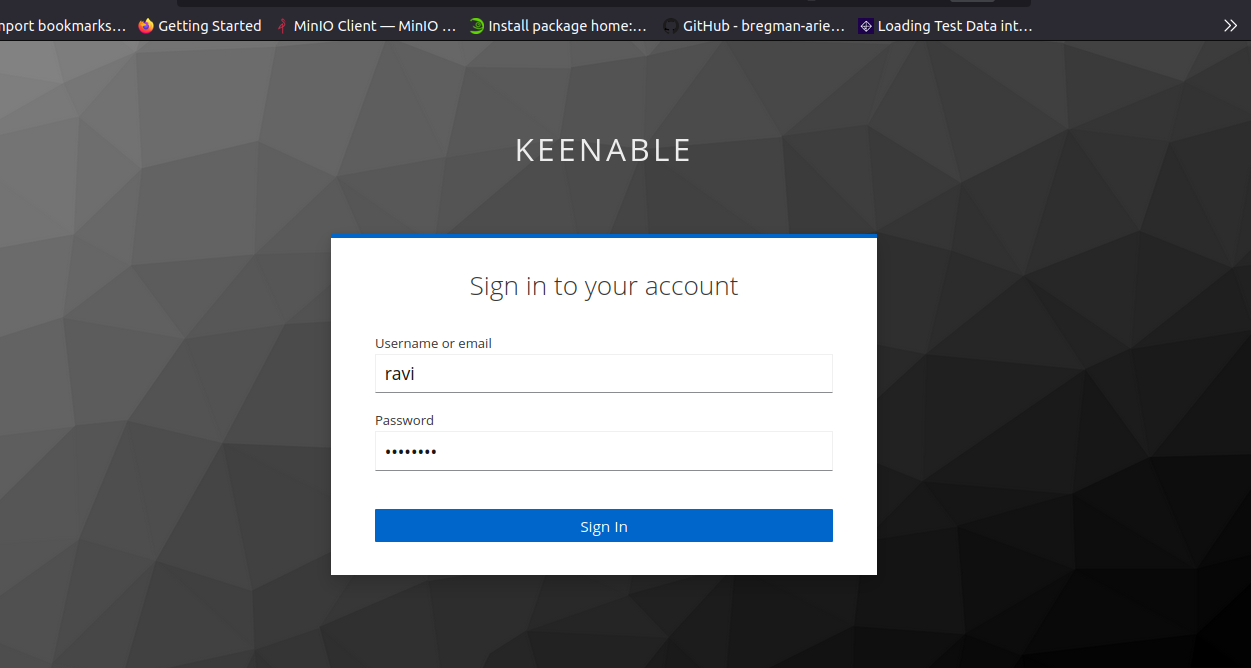
**Now I am going to login with my manoj user which have Admin role**

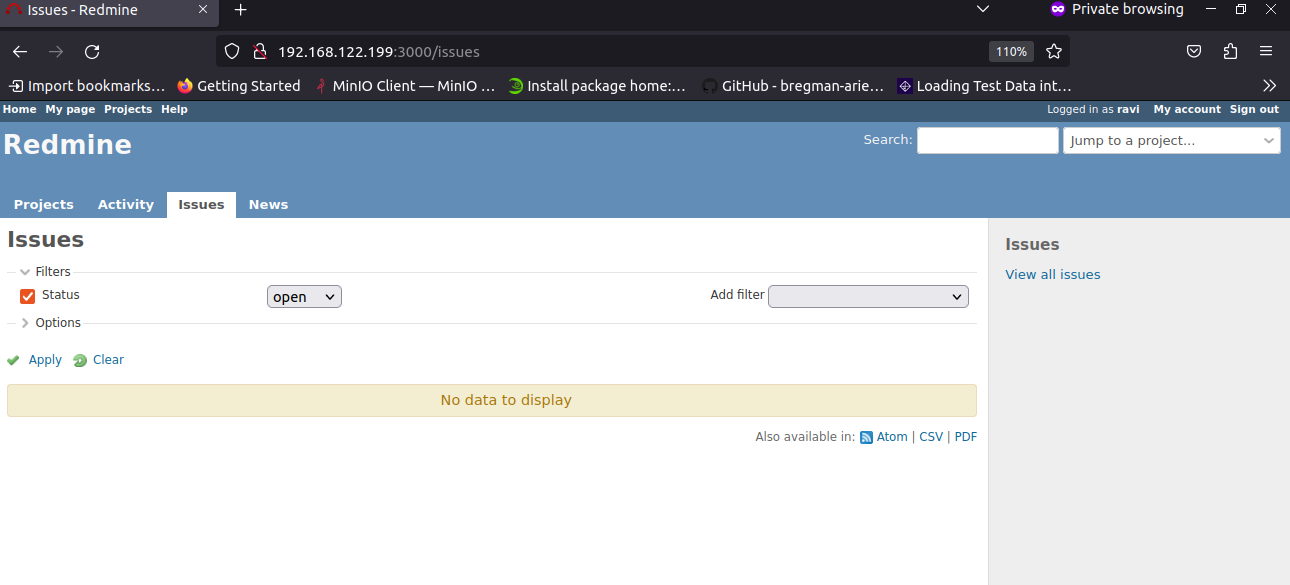
****

****

****

**Also I am going to login with another user which is ravi and he is a normal user**

****

****

**And then log out**

**##########################################################################################**

**Integrate Superset with keycloak**

**##########################################################################################**

**Superset Dockerfile**

| **manoj@crunchy-postgres:~$ cat Dockerfile # syntax=docker/dockerfile:1  FROM apache/superset # Switching to root to install the required packages USER root  # if you prefer Postgres, you may want to use `psycopg2-binary` instead RUN pip install psycopg2-binary RUN pip install flask-oidc==1.3.0 RUN pip install flask\_openid RUN pip install python-keycloak RUN pip install itsdangerous==2.0.1  # Switching back to using the `superset` user USER superset** |
| --- |

**Build this image with tag**

| **manoj@crunchy-postgres:~$ podman build -t superset:manoj . STEP 1/8: FROM apache/superset STEP 2/8: USER root --> Using cache 9f0efc66f1c77f0f265e54d60ecd6b3cfe0008b5edc1f75f6d154cc27f18a2ec --> 9f0efc66f1c STEP 3/8: RUN pip install psycopg2-binary --> Using cache b88bb033dda729ae992c2fee56d237c4b555fdc2beb5e8f22697a73882100c02 --> b88bb033dda STEP 4/8: RUN pip install flask-oidc==1.3.0 --> Using cache a1497a4553fcb3711db32695fb1c1796a12977e6be12bb4da8b6ee8169d1e436 --> a1497a4553f STEP 5/8: RUN pip install flask\_openid --> Using cache cc85c687110d41c157a64e4e96a3fb0376d58971e6c574cf136a5397017238c0 --> cc85c687110 STEP 6/8: RUN pip install python-keycloak --> Using cache 8537ebfb684174f69952b7da8125bef2fde6f7c1ecdcc71a02ac2f86e8fae77f --> 8537ebfb684 STEP 7/8: RUN pip install itsdangerous==2.0.1 --> Using cache d170cde073db53e3d65f49d24b9e39def15295e65806f19fd3cc6f572427fa96 --> d170cde073d STEP 8/8: USER superset --> Using cache 37735fa096a458d96e7c84f71350197f8dffc5814d9d471ec12196539525c8f1 COMMIT superset:manoj --> 37735fa096a Successfully tagged localhost/superset:manoj Successfully tagged localhost/superset\_apache:mj 37735fa096a458d96e7c84f71350197f8dffc5814d9d471ec12196539525c8f1 manoj@crunchy-postgres:~$** |
| --- |

| **manoj@crunchy-postgres:~$ podman images REPOSITORY TAG IMAGE ID CREATED SIZE localhost/superset manoj 37735fa096a4 4 days ago 1.33 GB** |
| --- |

**manoj@crunchy-postgres:~$ mkdir -p superset/app**

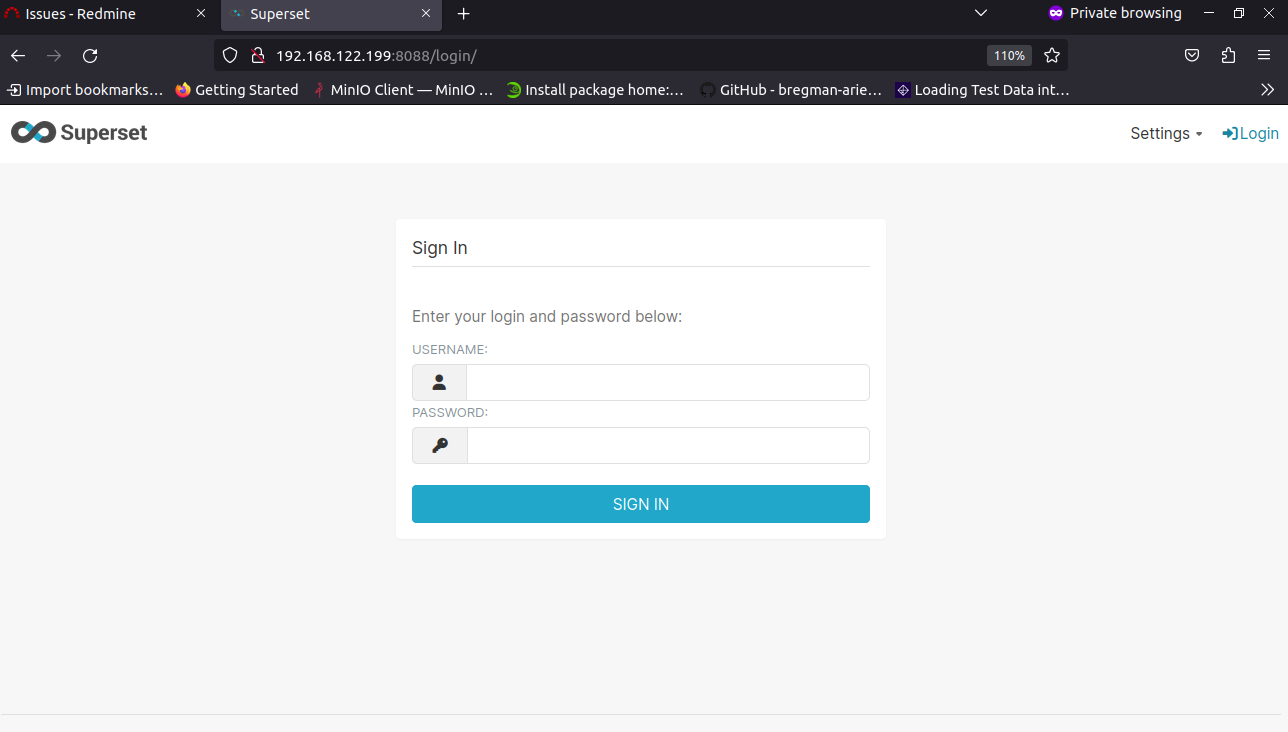
**Then run the superset container with volume**

**Run this command for generating secret key**

| **manoj@crunchy-postgres:~$ openssl rand -base64 42 0s3Hcl9cqUPBvWZdULVNygH2gjmn44/JFX//HhQXDGqYwSYlufMAXq0O** |
| --- |

| **manoj@crunchy-postgres:~$ podman run -d -p 8088:8088 -v /home/manoj/superse/app:/app/pythonpath -e "SUPERSET\_SECRET\_KEY=0s3Hcl9cqUPBvWZdULVNygH2gjmn44/JFX//HhQXDGqYwSYlufMAXq0O" --name superHIT superset:manoj 26890be963c6a3ddaa929466b32c921115e556e70432f2ebec6ae910fb376416 manoj@crunchy-postgres:~$** |
| --- |

**Go on browser and type http://<ip>:8088**

****

| **manoj@crunchy-postgres:~$ podman exec -it superHIT superset fab create-admin --username admin --firstname Superset --lastname Admin --email admin@superset.com --password admin**  **logging was configured successfully 2023-10-10 08:45:45,888:INFO:superset.utils.logging\_configurator:logging was configured successfully 2023-10-10 08:45:45,897:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'> /usr/local/lib/python3.9/site-packages/flask\_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.  warnings.warn( Recognized Database Authentications. Admin User admin created.** |
| --- |

**Migrate local DB to latest**

| **manoj@crunchy-postgres:~$ podman exec -it superHIT superset db upgrade logging was configured successfully 2023-10-10 08:48:23,761:INFO:superset.utils.logging\_configurator:logging was configured successfully 2023-10-10 08:48:23,767:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'> /usr/local/lib/python3.9/site-packages/flask\_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.  warnings.warn( WARNI [alembic.env] SQLite Database support for metadata databases will be removed in a future version of Superset. INFO [alembic.runtime.migration] Context impl SQLiteImpl. INFO [alembic.runtime.migration] Will assume transactional DDL.** |
| --- |

**Setup roles**

| **manoj@crunchy-postgres:~$ podman exec -it superHIT superset init logging was configured successfully** |
| --- |

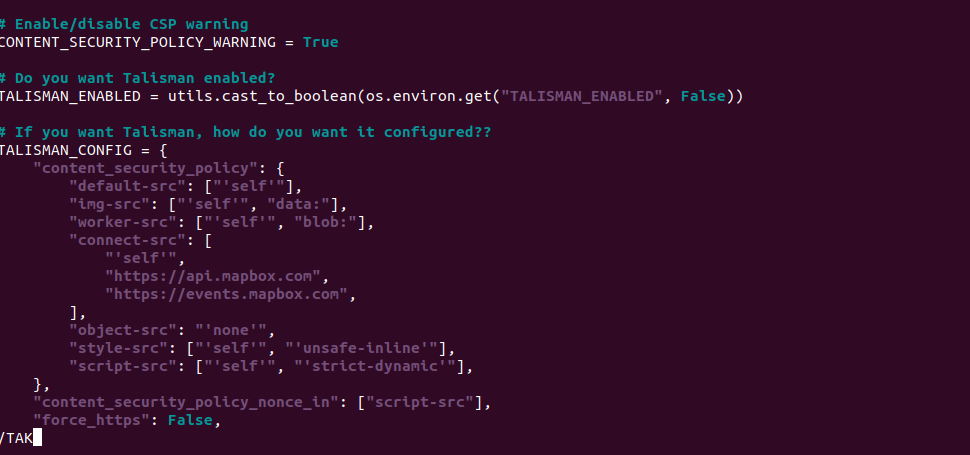
**After done this steps restart superset you will be able to login with admin user with password (admin)**

**Go inside the container with root privileges and update it and install vim for some changes**

**> TALISMAN\_ENABLED defaults to True; set this to False in order to disable CSP**

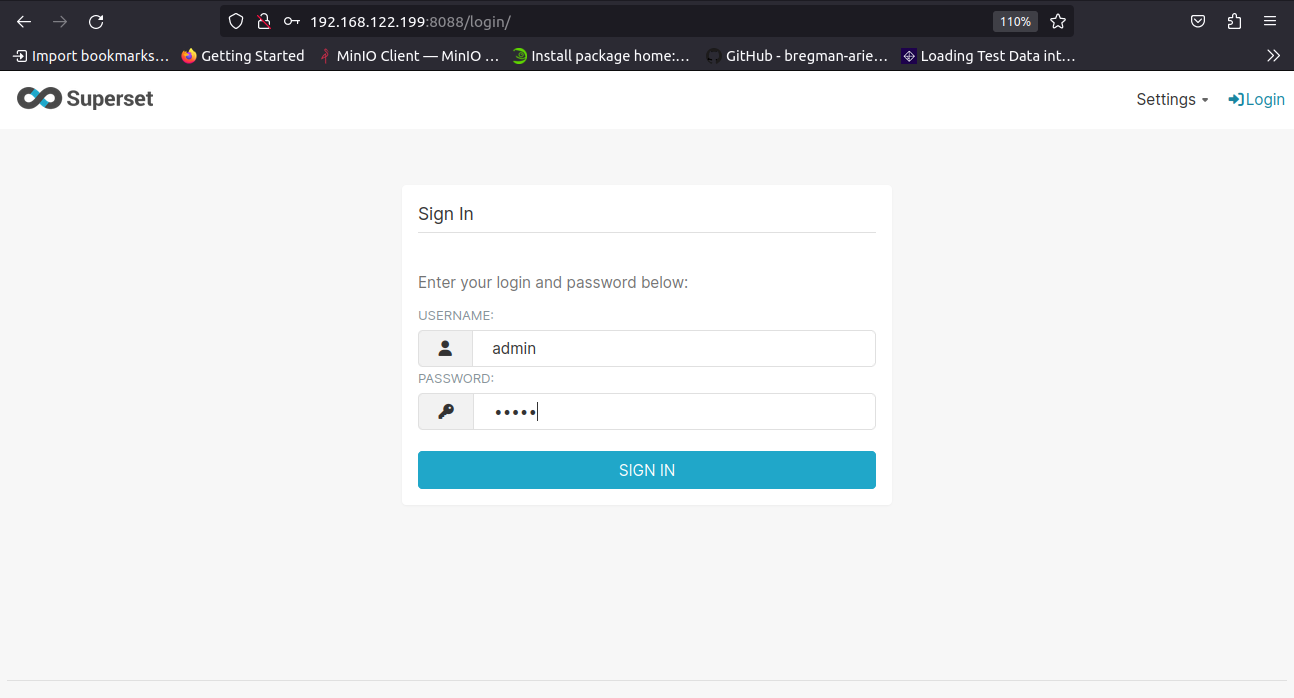
**Make changes in config.py in superset directory inside the container**

| **manoj@crunchy-postgres:~$ podman exec -it -u root superHIT bash root@26890be963c6:/app# apt-get update Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB] Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB] Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB] Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8780 kB] Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [6408 B] Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [78.6 kB] Fetched 9116 kB in 3s (3405 kB/s)  Reading package lists... Done  root@26890be963c6:/app# apt install vim Reading package lists... Done Building dependency tree... Done Reading state information... Done The following additional packages will be installed:**  **oot@26890be963c6:/app/superset# vim config.py** |
| --- |

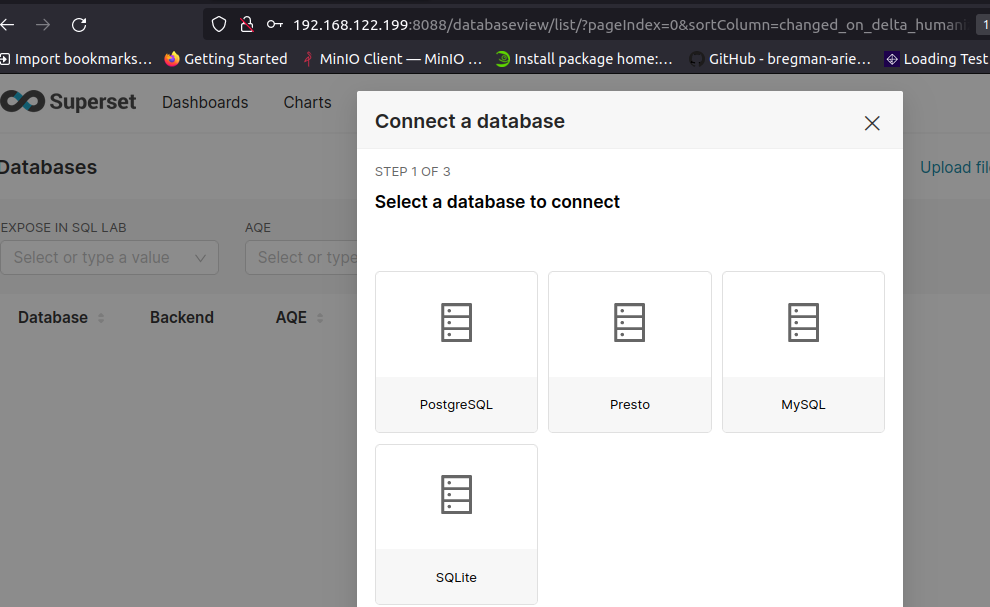


**After making changes restart the container**

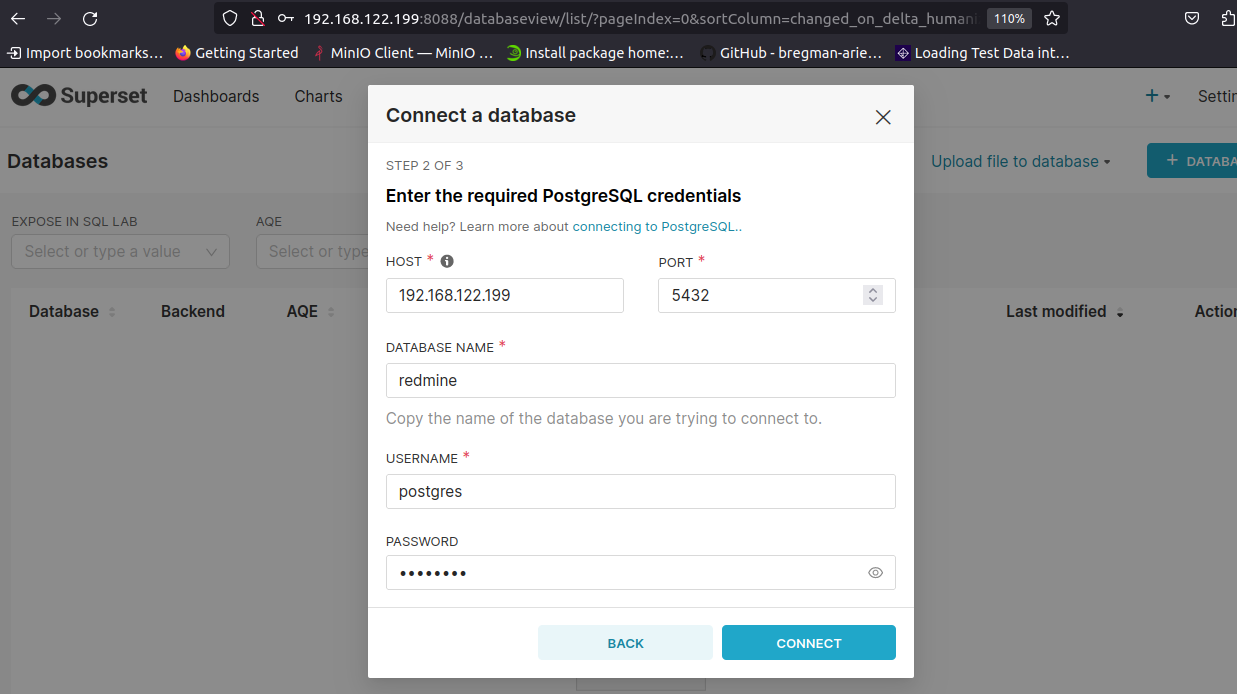
| **manoj@crunchy-postgres:~$ podman restart superHIT 26890be963c6a3ddaa929466b32c921115e556e70432f2ebec6ae910fb376416** |
| --- |

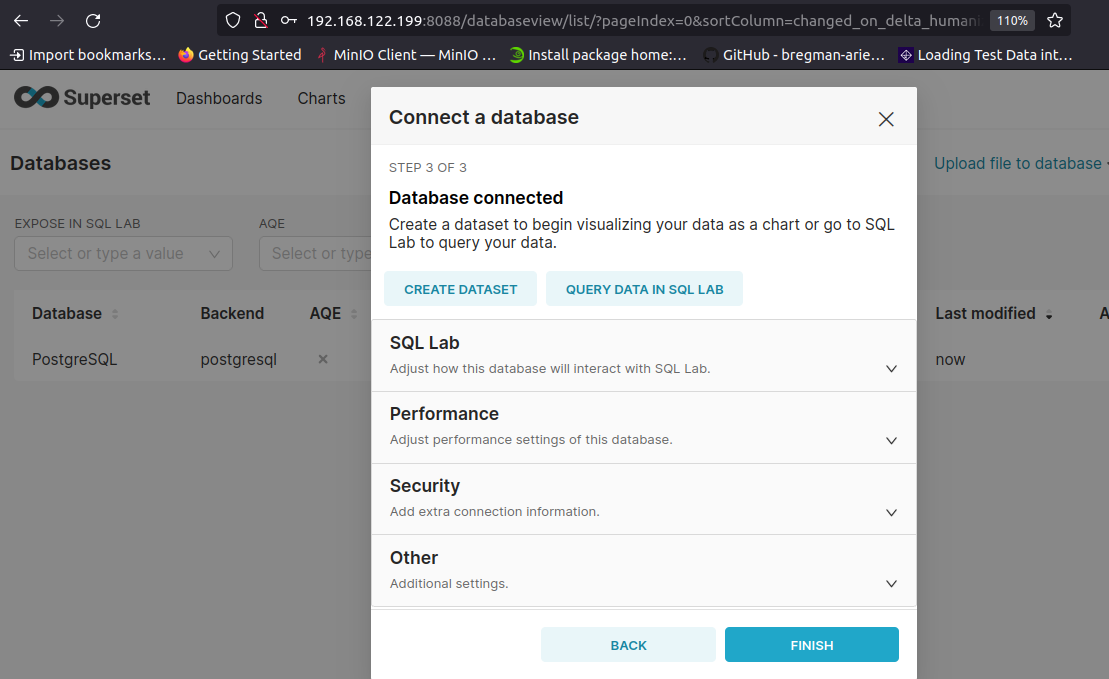
****

**Go to setting section and add postgres database which was already running**

****

**Select postgres in my case**

****

****

**NOW database has connected successfully,**

**ALL THE ABOVE STEPS ARE TAKEN FOR ONLY LOGIN WITH LOCAL Admin**

**NOW we are going toward the implement superset with keycloak**

**Reference link:-**

[**https://github.com/apache/superset/discussions/13915**](https://github.com/apache/superset/discussions/13915)

**NOW I HAVE CREATED 3 files inside app directory**

**manoj@crunchy-postgres:~/superse/app$ ls**

**client\_secret.json keycloak\_security\_manager.py superset\_config.py**

**manoj@crunchy-postgres:~/superse/app$**

| **manoj@crunchy-postgres:~/superse/app$ vim client\_secret.json**  **manoj@crunchy-postgres:~/superse/app$ cat client\_secret.json {  "web": {  "issuer": "http://192.168.122.199:8080/realms/Keenable",  "auth\_uri": "http://192.168.122.199:8080/realms/Keenable/protocol/openid-connect/auth",  "client\_id": "keenable",  "client\_secret": "aGwAujXoaLO9AFWyYSitPSdNESn77e5r",  "verify\_ssl\_server": false,  "redirect\_uris": ["http://192.168.122.199:8088/\*"],  "userinfo\_uri": "http://192.168.122.199:8080/realms/Keenable/protocol/openid-connect/userinfo",  "token\_uri": "http://192.168.122.199:8080/realms/Keenable/protocol/openid-connect/token",  "token\_introspection\_uri": "http://192.168.122.199:8080/realms/Keenable/protocol/openid-connect/introspect"  } }** |
| --- |

| **manoj@crunchy-postgres:~/superse/app$ vim keycloak\_security\_manager.py**  **manoj@crunchy-postgres:~/superse/app$ cat keycloak\_security\_manager.py import os from flask\_appbuilder.security.manager import AUTH\_OID from superset.security import SupersetSecurityManager from flask\_oidc import OpenIDConnect from flask\_login import login\_user from urllib.parse import quote from flask import redirect, request from flask\_appbuilder.security.views import AuthOIDView from flask\_appbuilder.views import expose import logging  logger = logging.getLogger(\_\_name\_\_)  class OIDCSecurityManager(SupersetSecurityManager):   def \_\_init\_\_(self, appbuilder):  super(OIDCSecurityManager, self).\_\_init\_\_(appbuilder)  if self.auth\_type == AUTH\_OID:  self.oid = OpenIDConnect(self.appbuilder.get\_app)  self.authoidview = AuthOIDCView  class AuthOIDCView(AuthOIDView):   @expose('/login/', methods=['GET', 'POST'])  def login(self, flag=True):  sm = self.appbuilder.sm  oidc = sm.oid  superset\_roles = ["Admin", "Alpha", "Gamma", "Public", "granter", "sql\_lab"]  default\_role = "Gamma"   @self.appbuilder.sm.oid.require\_login  def handle\_login():  user = sm.auth\_user\_oid(oidc.user\_getfield('email'))   if user is None:  info = oidc.user\_getinfo(['preferred\_username', 'given\_name', 'family\_name', 'email', 'roles'])  roles = [role for role in superset\_roles if role in info.get('roles', [])]  roles += [default\_role, ] if not roles else []  user = sm.add\_user(info.get('preferred\_username'), info.get('given\_name'), info.get('family\_name'),  info.get('email'), [sm.find\_role(role) for role in roles])   login\_user(user, remember=False)  return redirect(self.appbuilder.get\_url\_for\_index)   return handle\_login()   @expose('/logout/', methods=['GET', 'POST'])  def logout(self):  oidc = self.appbuilder.sm.oid   oidc.logout()  super(AuthOIDCView, self).logout()  redirect\_url = 'http://192.168.122.199:8088/login/'  logging.debug('This is redirect uri %s ',self.appbuilder.get\_url\_for\_login)  return redirect(oidc.client\_secrets.get('issuer') + '/protocol/openid-connect/logout?client\_id='+ oidc.client\_secrets.get('client\_id')+'&post\_logout\_redirect\_uri=' + quote(redirect\_url))** |
| --- |

| **manoj@crunchy-postgres:~/superse/app$ vim superset\_config.py**  **manoj@crunchy-postgres:~/superse/app$ cat superset\_config.py from keycloak\_security\_manager import OIDCSecurityManager from flask\_appbuilder.security.manager import AUTH\_OID import os  # Get current directory curr = os.path.abspath(os.getcwd())  # Superset configurations for OIDC authentication and session management AUTH\_TYPE = AUTH\_OID #SECRET\_KEY = 'U66tVZgRFbQXTarU5mf/IYw2wpTOLZ8Tz6r6kSxYw732HqhzBbjiRokf' OIDC\_CLIENT\_SECRETS = curr + '/pythonpath/client\_secret.json' OIDC\_ID\_TOKEN\_COOKIE\_SECURE = False OIDC\_REQUIRE\_VERIFIED\_EMAIL = False OIDC\_OPENID\_REALM = 'Keenable' OIDC\_INTROSPECTION\_AUTH\_METHOD = 'client\_secret\_post' CUSTOM\_SECURITY\_MANAGER = OIDCSecurityManager AUTH\_USER\_REGISTRATION = False AUTH\_USER\_REGISTRATION\_ROLE = 'Gamma' OIDC\_TOKEN\_TYPE\_HINT = 'access\_token' OVERWRITE\_REDIRECT\_URI = 'http://192.168.122.199:8088/oidc\_callback'** |
| --- |

**Again update DB after making changes**

| **manoj@crunchy-postgres:~/superse/app$ podman exec -it superHIT superset db upgrade Loaded your LOCAL configuration at [/app/pythonpath/superset\_config.py] logging was configured successfully 2023-10-10 09:46:12,231:INFO:superset.utils.logging\_configurator:logging was configured successfully 2023-10-10 09:46:12,239:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'> /usr/local/lib/python3.9/site-packages/flask\_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.  warnings.warn( WARNI [alembic.env] SQLite Database support for metadata databases will be removed in a future version of Superset. INFO [alembic.runtime.migration] Context impl SQLiteImpl. INFO [alembic.runtime.migration] Will assume transactional DDL. manoj@crunchy-postgres:~/superse/app$** |
| --- |

| **manoj@crunchy-postgres:~/superse/app$ podman exec -it superHIT superset db upgrade Loaded your LOCAL configuration at [/app/pythonpath/superset\_config.py] logging was configured successfully 2023-10-10 09:52:49,196:INFO:superset.utils.logging\_configurator:logging was configured successfully 2023-10-10 09:52:49,201:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'> /usr/local/lib/python3.9/site-packages/flask\_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.  warnings.warn( WARNI [alembic.env] SQLite Database support for metadata databases will be removed in a future version of Superset. INFO [alembic.runtime.migration] Context impl SQLiteImpl. INFO [alembic.runtime.migration] Will assume transactional DDL.** |
| --- |

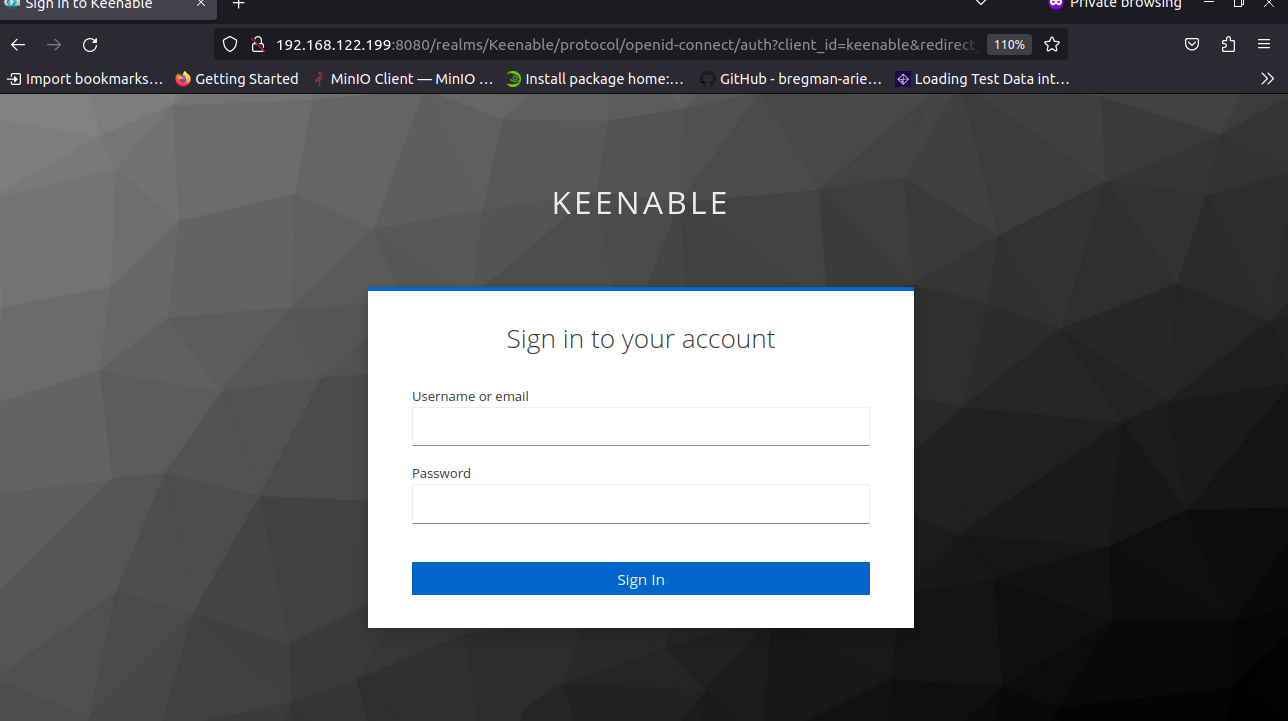
| **manoj@crunchy-postgres:~/superse/app$ podman exec -it superHIT superset init Loaded your LOCAL configuration at [/app/pythonpath/superset\_config.py] logging was configured successfully 2023-10-10 09:48:34,301:INFO:superset.utils.logging\_configurator:logging was configured successfully 2023-10-10 09:48:34,309:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'> /usr/local/lib/python3.9/site-packages/flask\_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.  warnings.warn( Syncing role definition 2023-10-10 09:48:38,015:INFO:superset.security.manager:Syncing role definition Syncing Admin perms 2023-10-10 09:48:38,054:INFO:superset.security.manager:Syncing Admin perms Syncing Alpha perms 2023-10-10 09:48:38,322:INFO:superset.security.manager:Syncing Alpha perms Syncing Gamma perms 2023-10-10 09:48:38,605:INFO:superset.security.manager:Syncing Gamma perms Syncing sql\_lab perms 2023-10-10 09:48:38,937:INFO:superset.security.manager:Syncing sql\_lab perms** |
| --- |

**Also restart the container again**

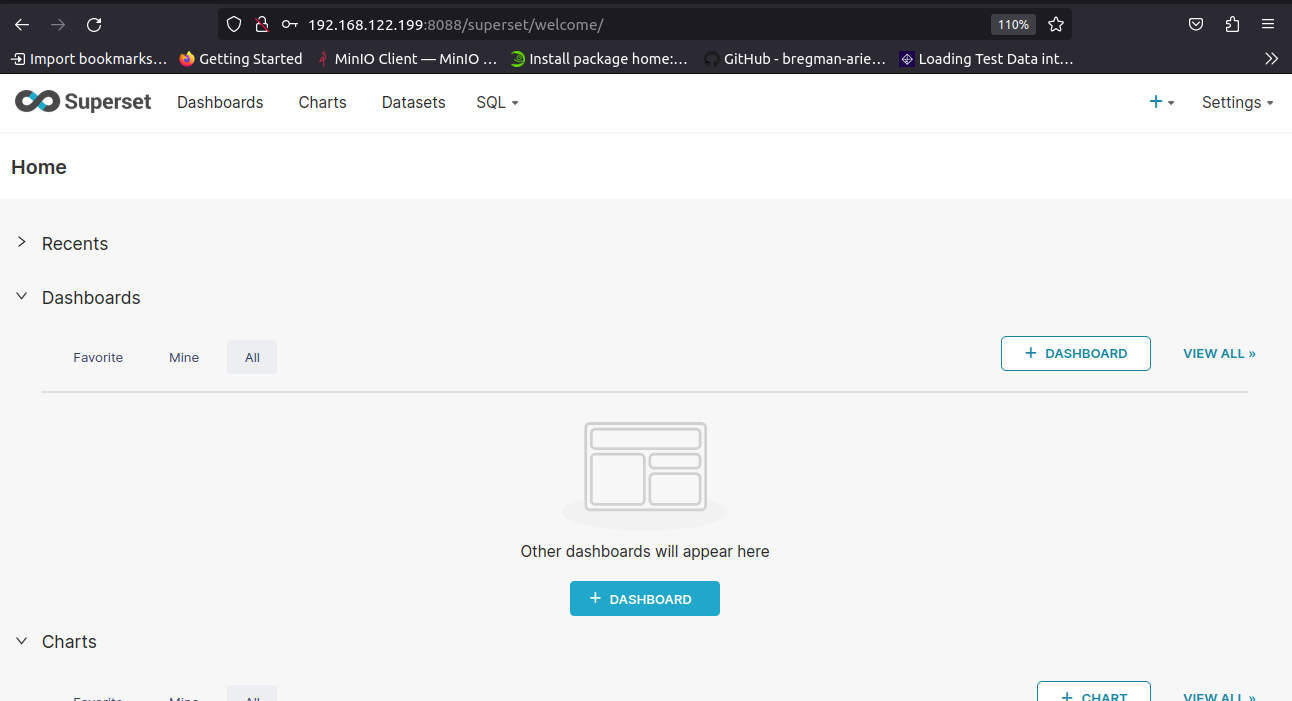
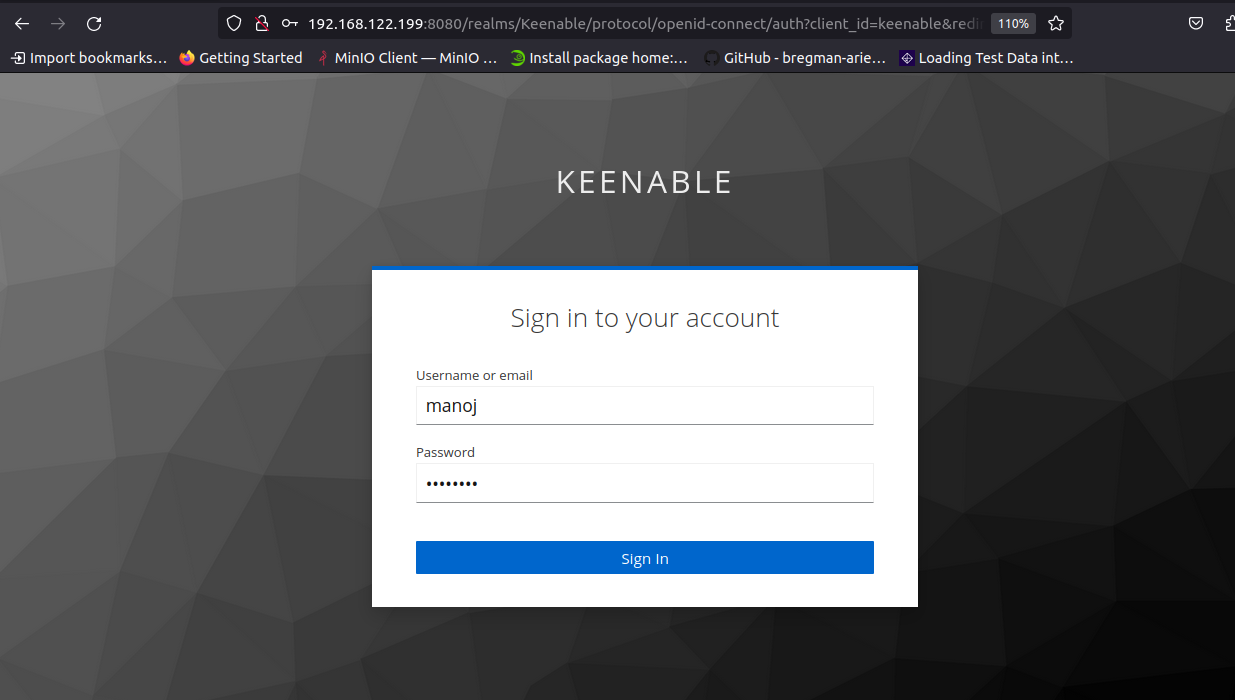
**(Remember after making changes you will have to restart the container)**

| **manoj@crunchy-postgres:~/superse/app$ podman restart superHIT 26890be963c6a3ddaa929466b32c921115e556e70432f2ebec6ae910fb376416** |
| --- |

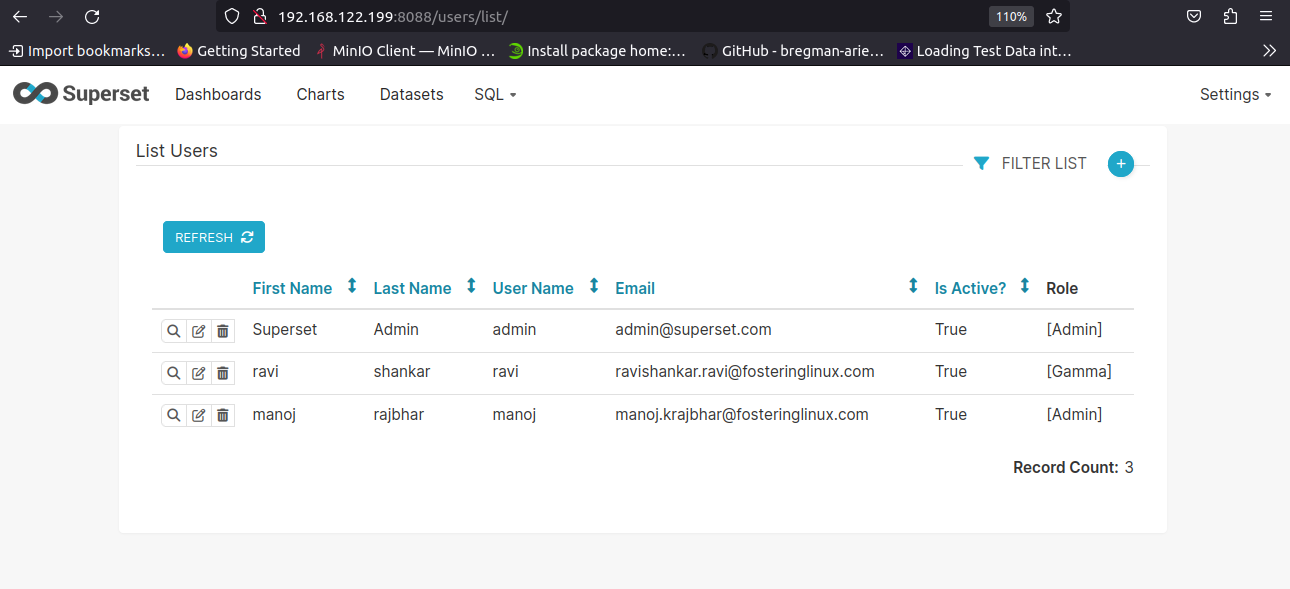
**Hit on browser with http:<ip>:8088 it will automatically redirect toward keycloak sign in**

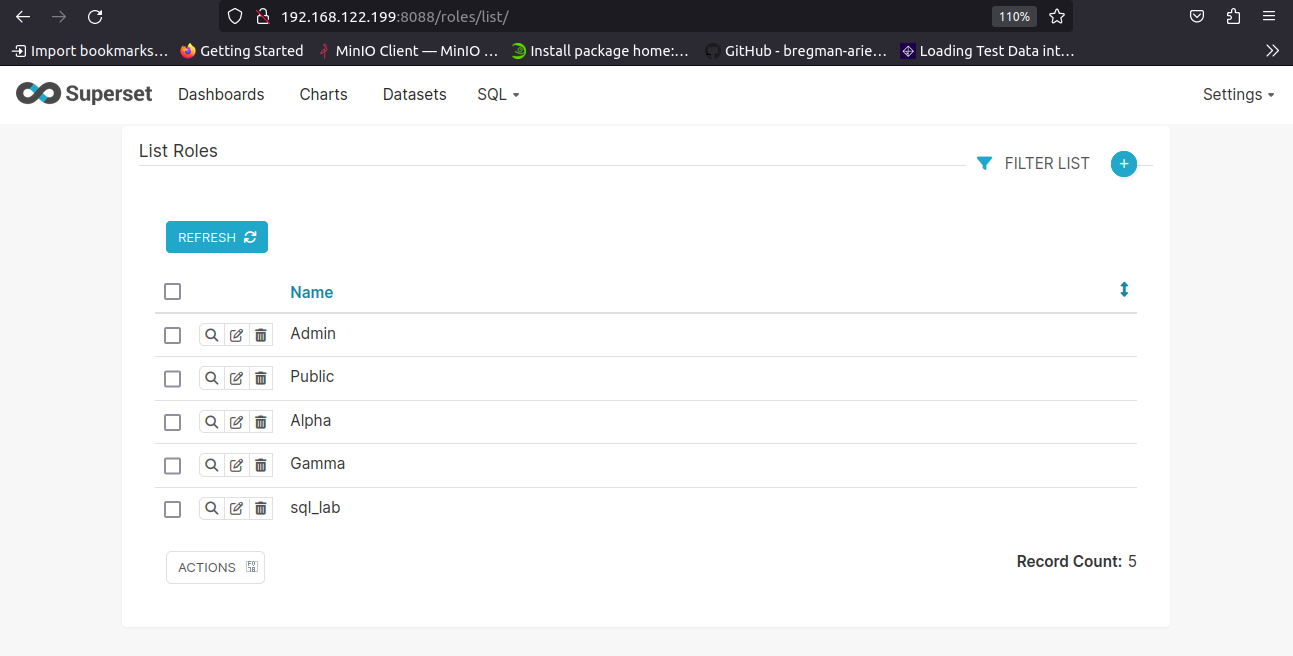
****

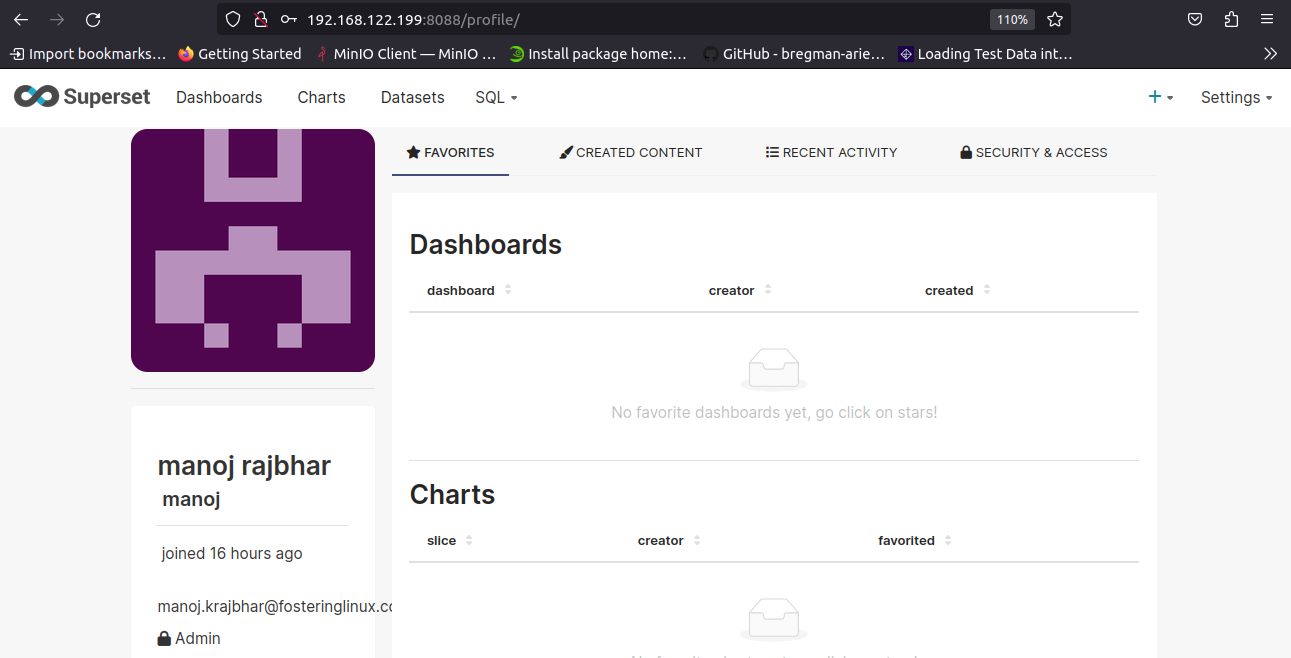
**Login with mano user which have admin credentials as I have given in keycloak configuration**

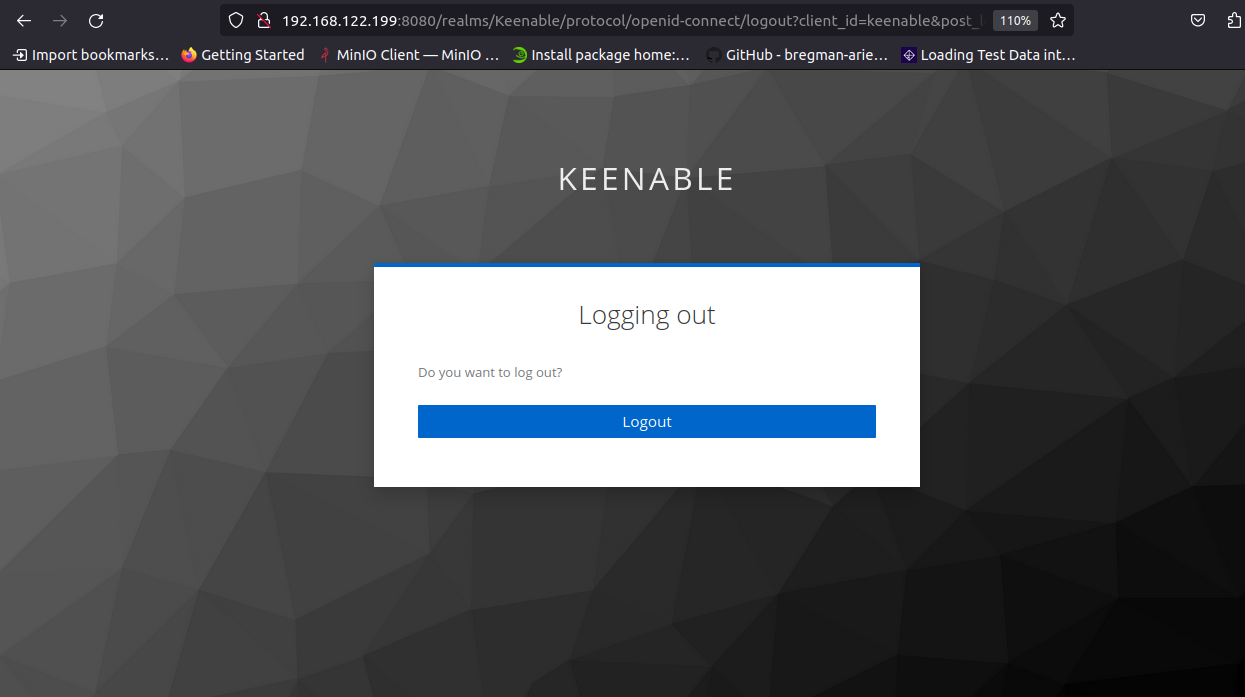
****

**LIST OF USERS AS I HAVE CONFIGURED IN KEYCLOAK**

****

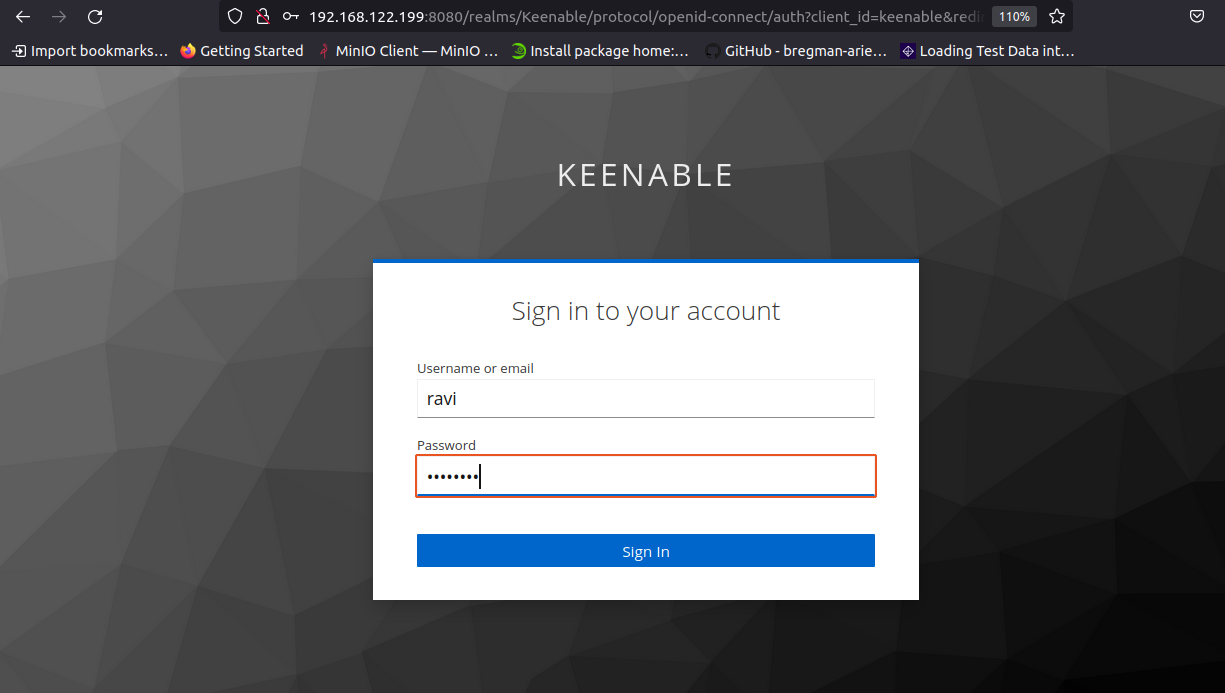
****

****

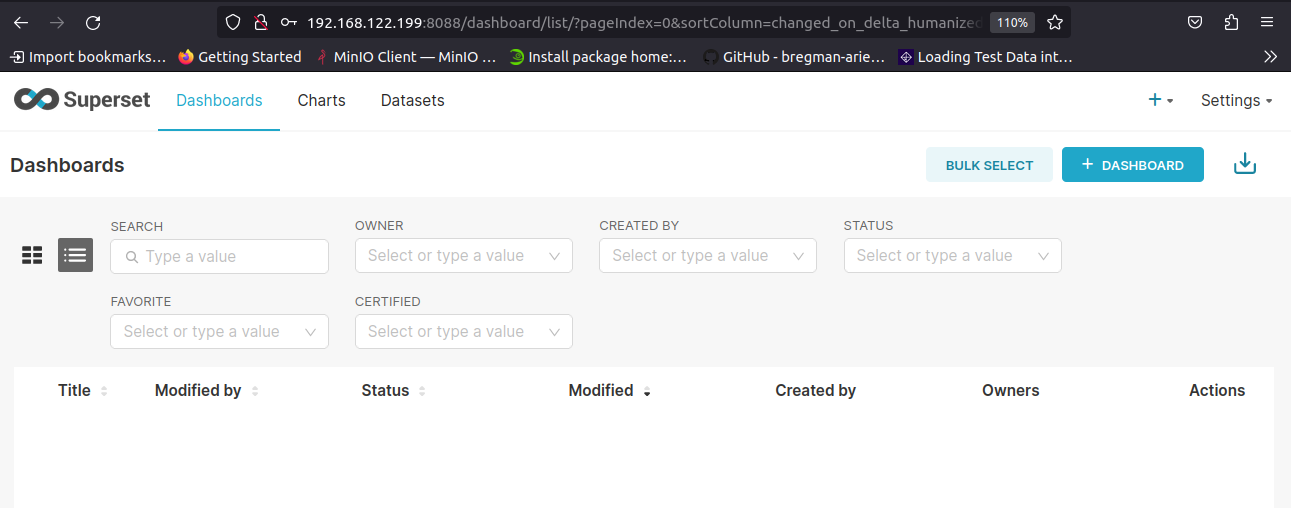
****

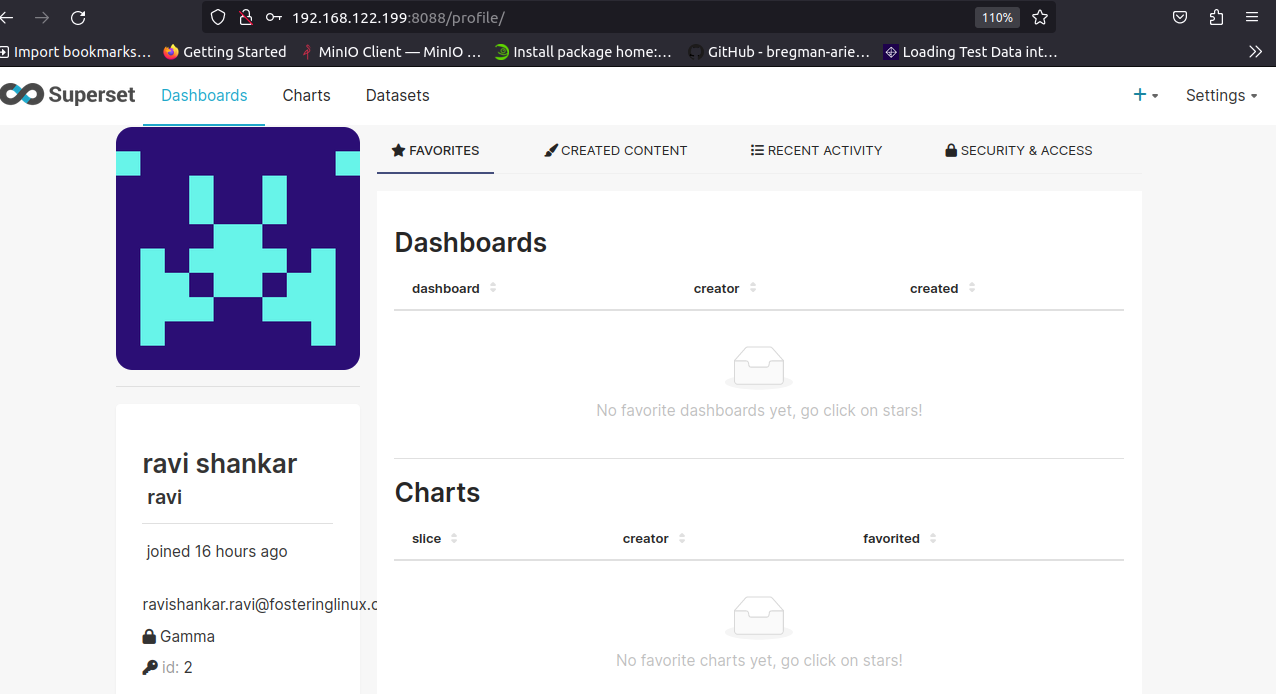
**After logout check another simple user are able to login or not**

**NOW LOGIN WITH SIMPLE USERS WHO HAVE GAMMA ROLE**

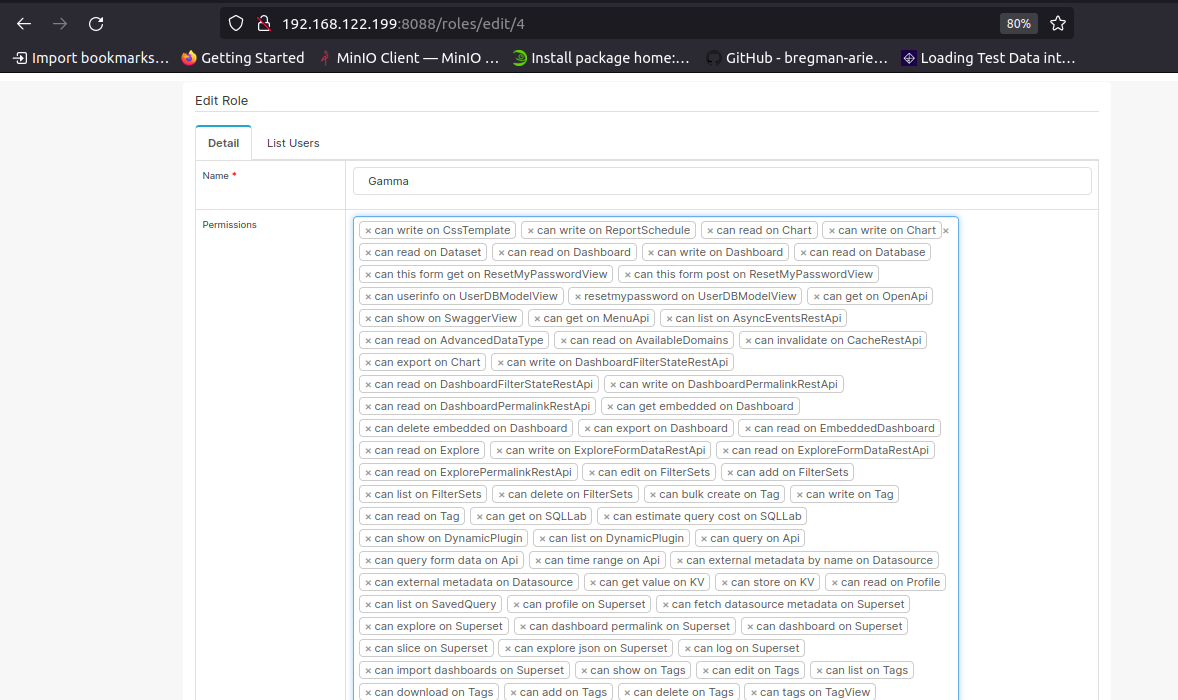
****

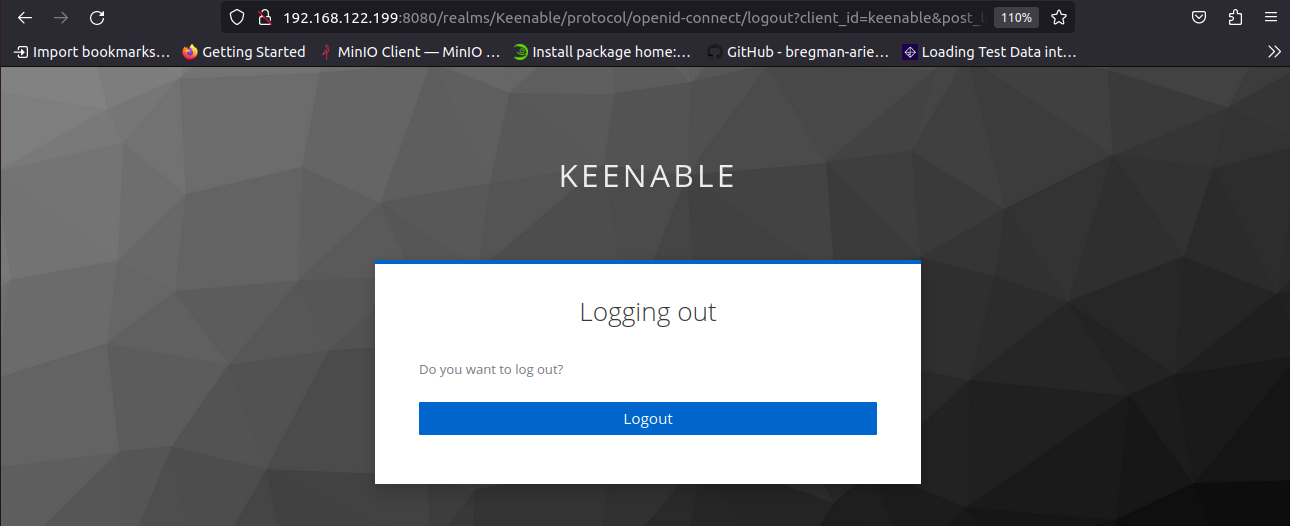
**Yes we are able to login with second users also**

****

****

**Gamma**: Gamma users have restricted access and can only consume data from assigned data sources. They can view slices and dashboards created from these data sources.

****

****