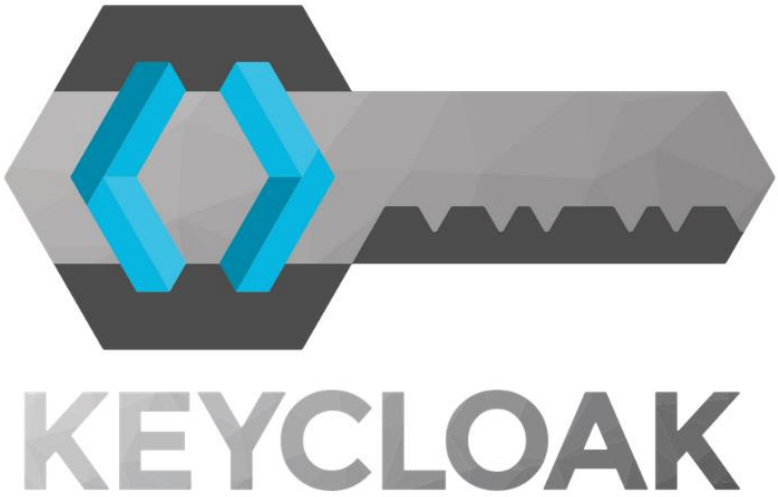
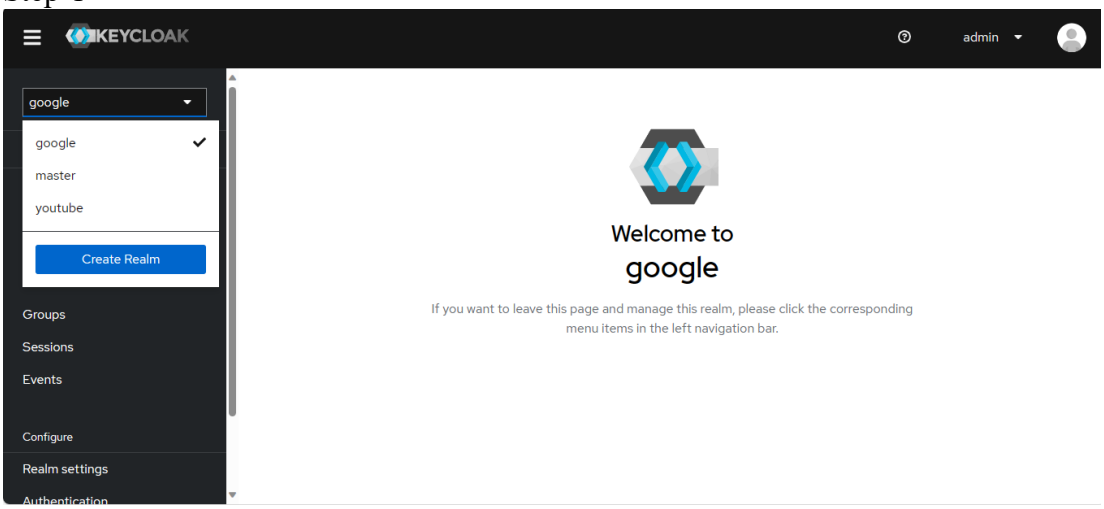


Keycloak 22 Guide by Sakib

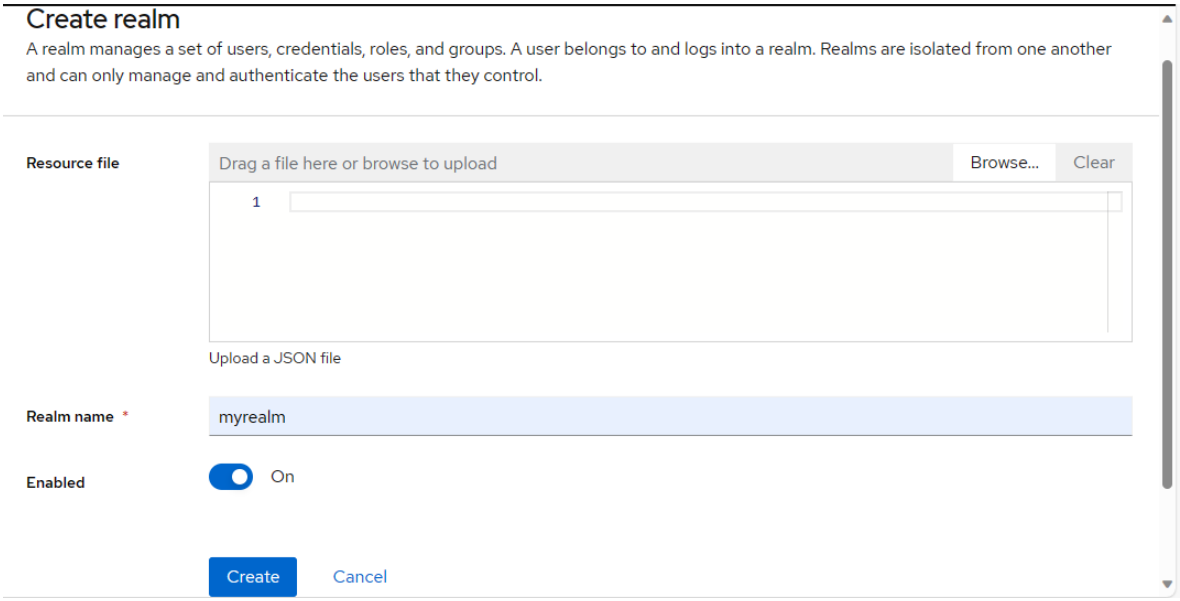
Contents

Create Realm

Step-1

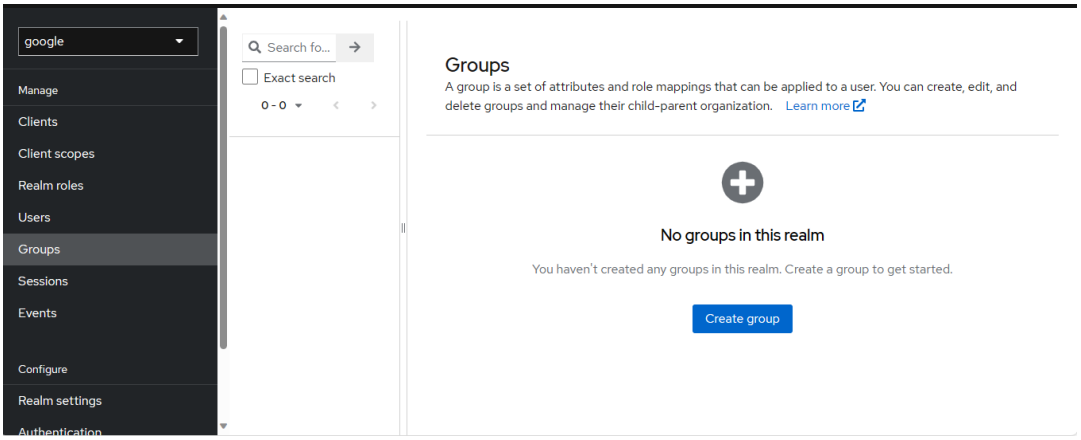


Step-2

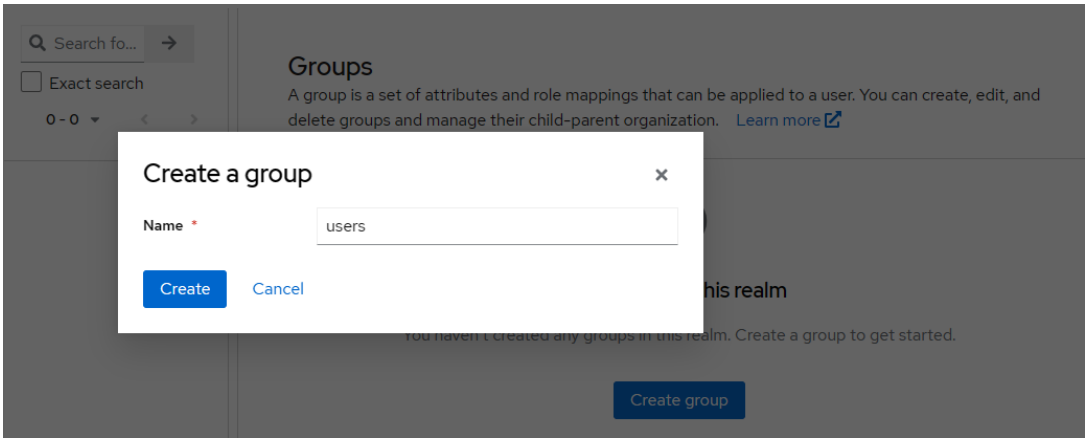


Setup Group and Assign Realm Roles

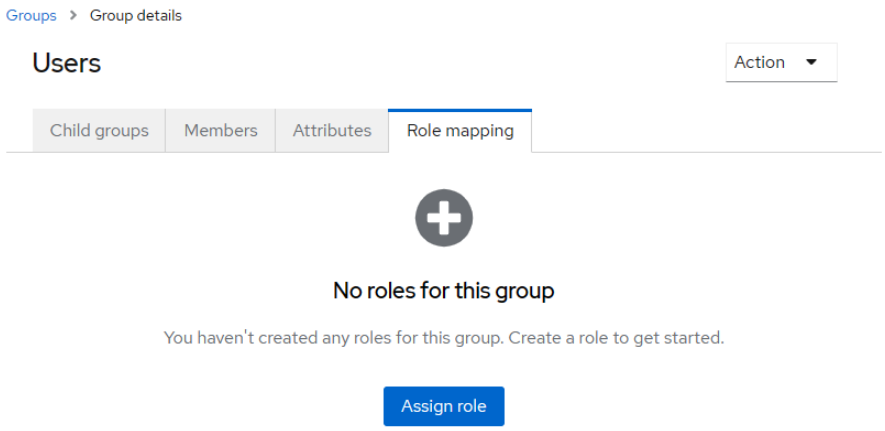
Step:1



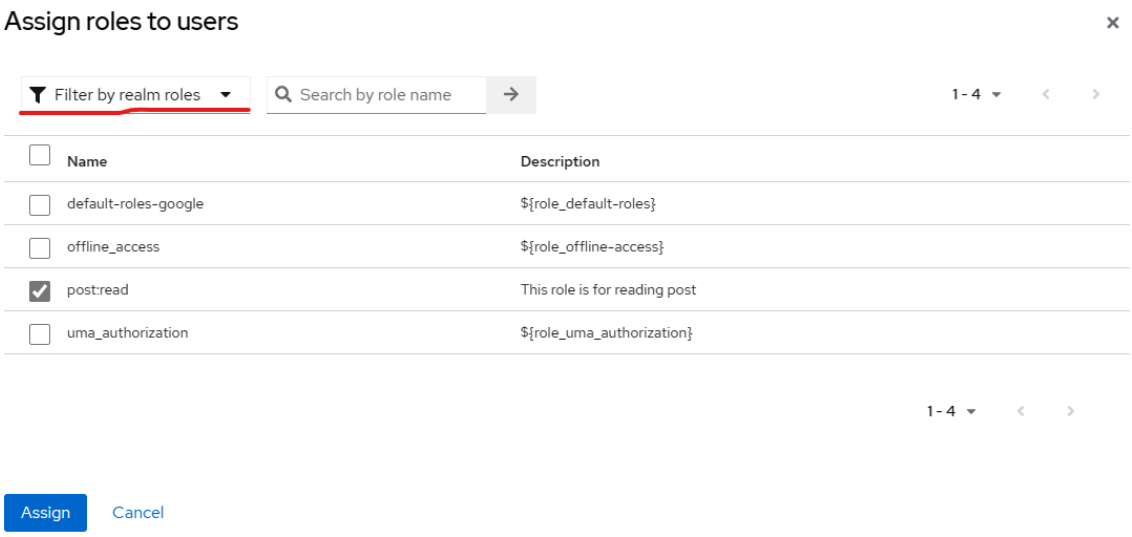
Step:2



Step:3 - Assign roles in group



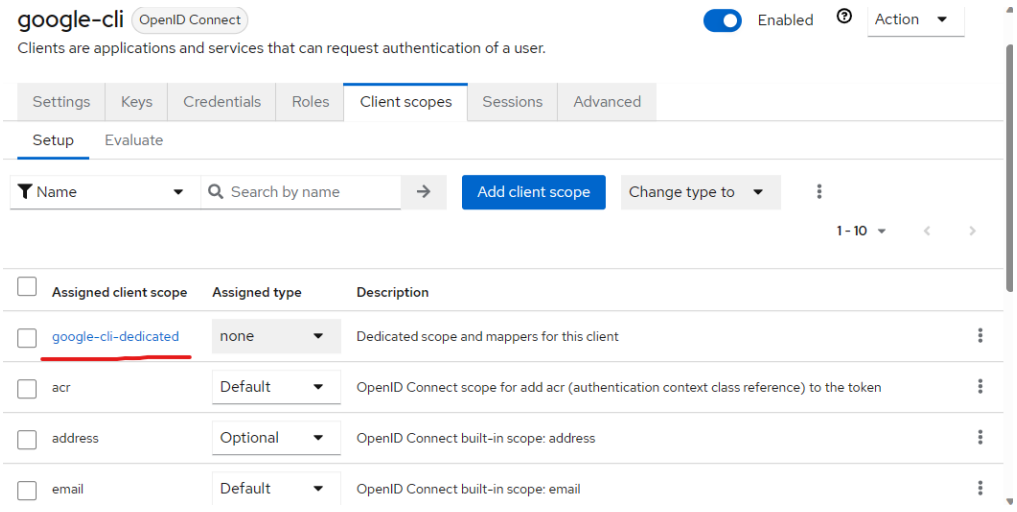
Step:4 – Adding reading post permission for this USERS group



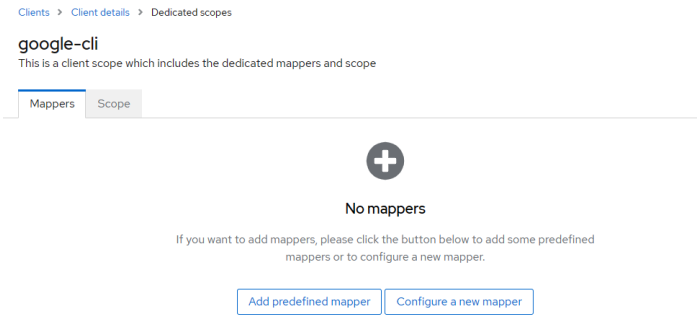
Step:5 – Adding this USERS group in the client

Go to Client>{CLIENT_NAME}>Client Scopes

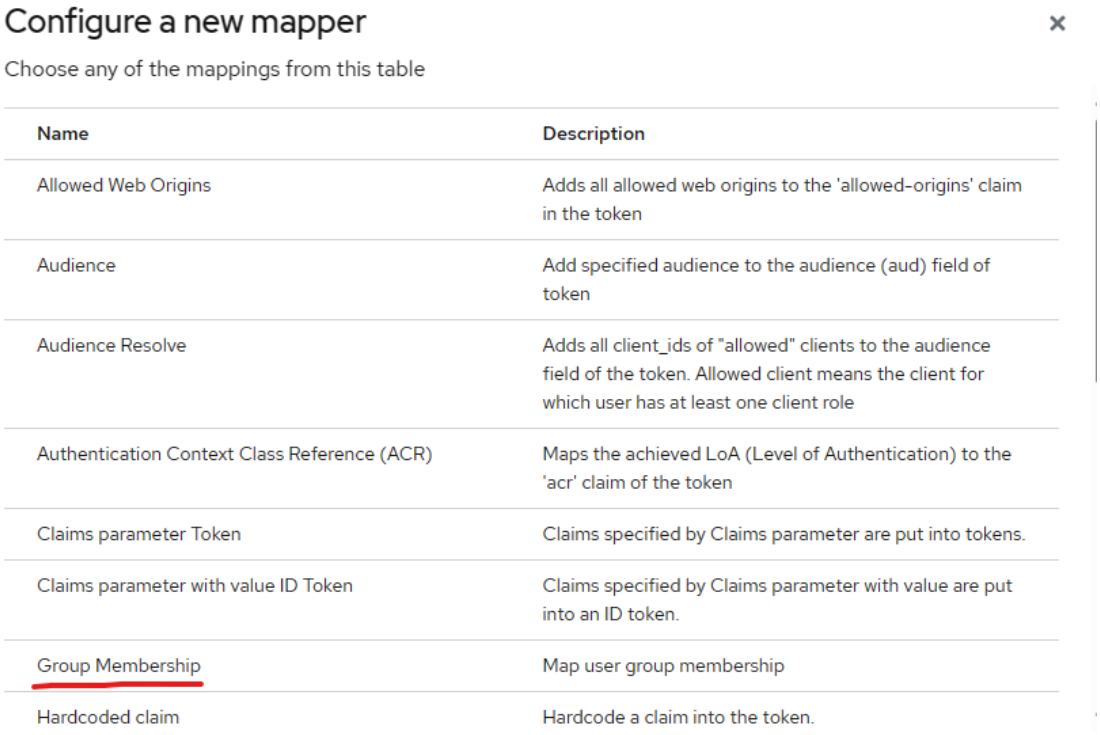
i) Go to the google-cli-dedicated (client scope name will depend on client name)



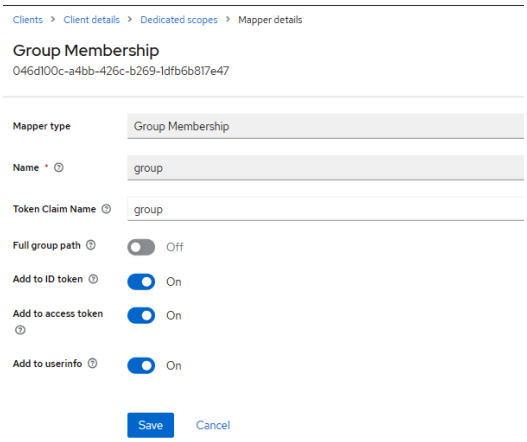
ii) Click on Configure a new mapper



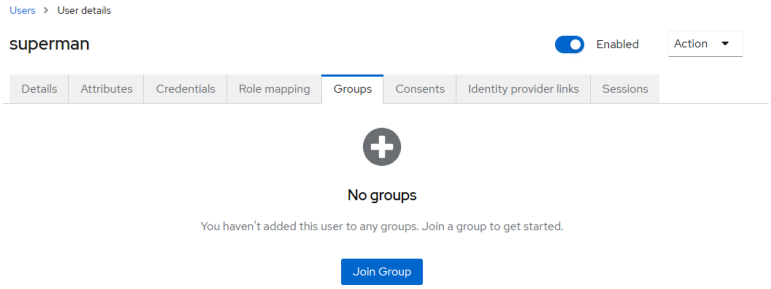
iii) Click on Group Membership



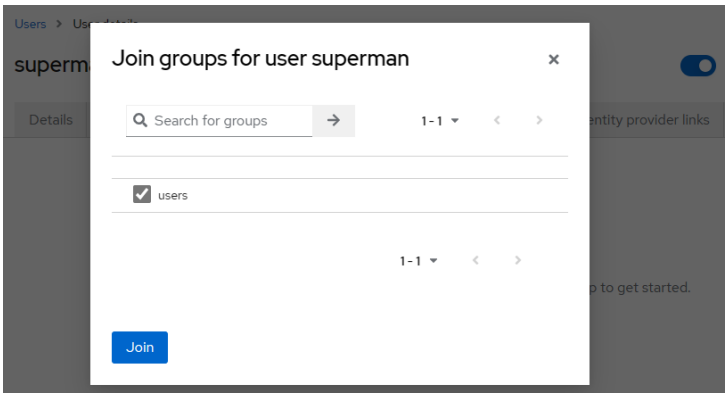
iv) Click on Save



Step-6: Adding user to the group

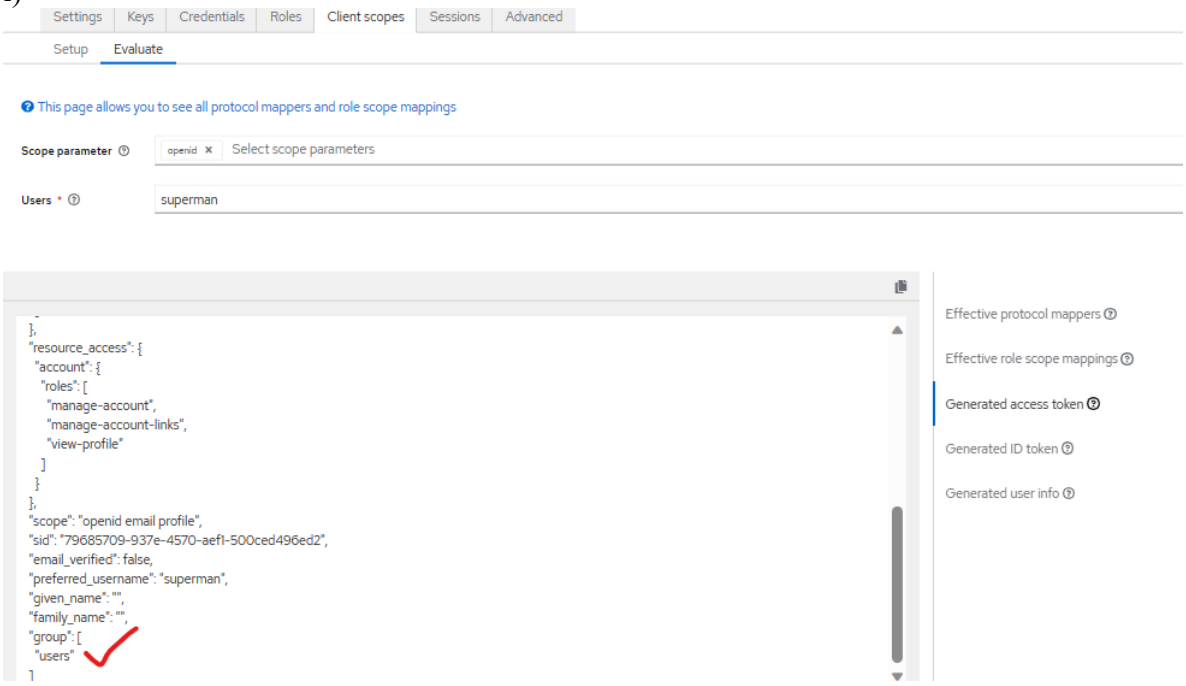


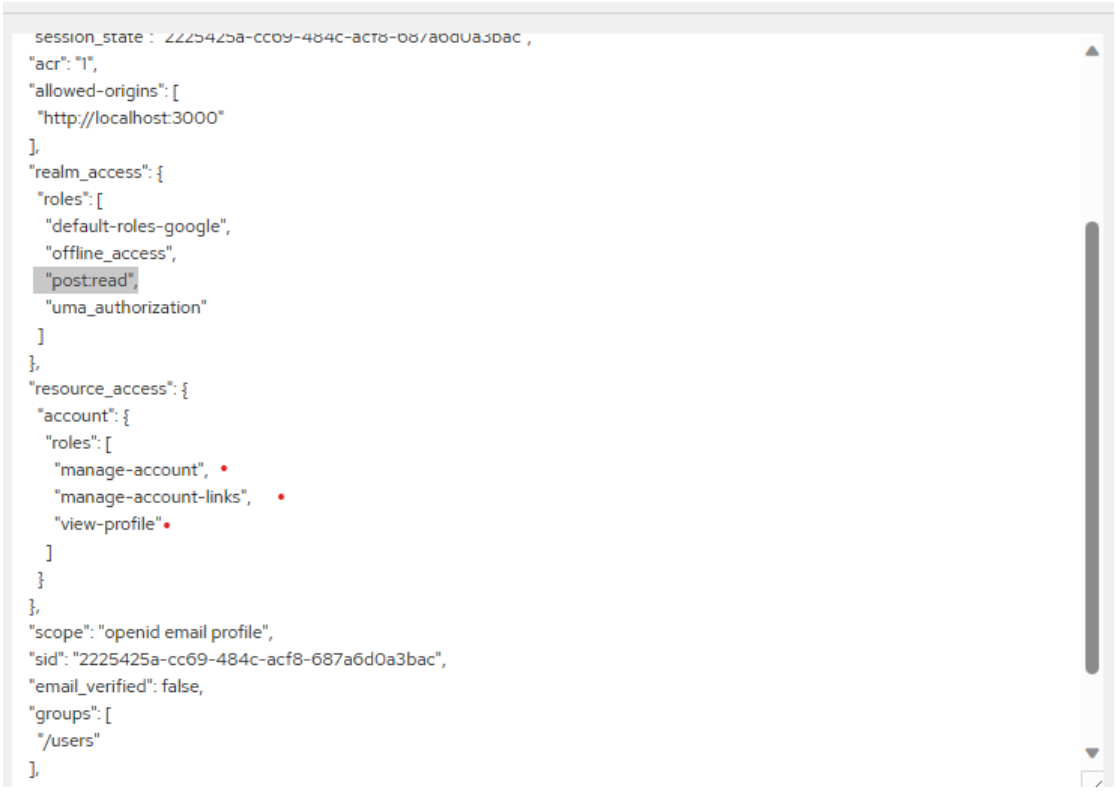
Add the user to the USERS group. Or we can add the user from group directly.



Step-7 - Let's check that user's access token
Go to Client>{CLIENT_NAME}>Client Scopes. Then Evaluate tab

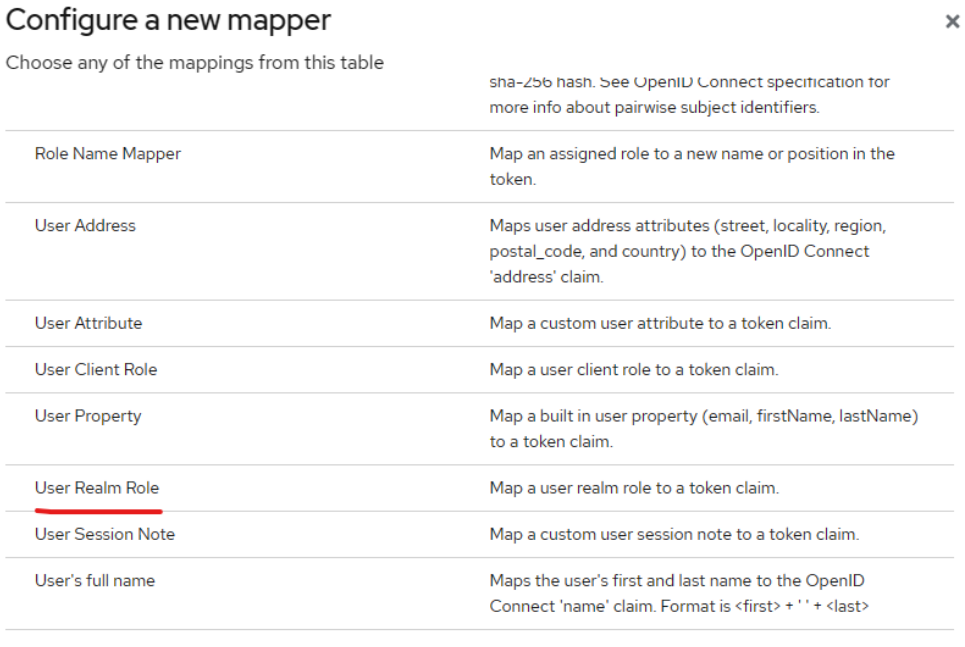
i)





Adding custom property for roles in access token

This can be done by adding a new mapper in client scope



Clients > Client details > Dedicated scopes > Mapper details

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type	User Realm Role
Name *	realm_roles
Realm Role prefix	
Multivalued	On
Token Claim Name	realm_roles
Claim JSON Type	String
Add to ID token	On
Add to access token	On
Add to userinfo	On

Save Cancel

Users * superman


```
      "view-profile"
    ]
  }
},
"scope": "openid email profile",
"sid": "723645a9-4c02-4ed1-9d05-eeb8694fc6f4",
"realm_roles": [
  "default-roles-google",
  "offline_access",
  "post:read",
  "uma_authorization"
],
"email_verified": false,
"groups": [
  "/users"
],
"preferred_username": "superman",
"given_name": "",
"family_name": ""
}
```


Effective protocol mappers

Effective role scope mappings

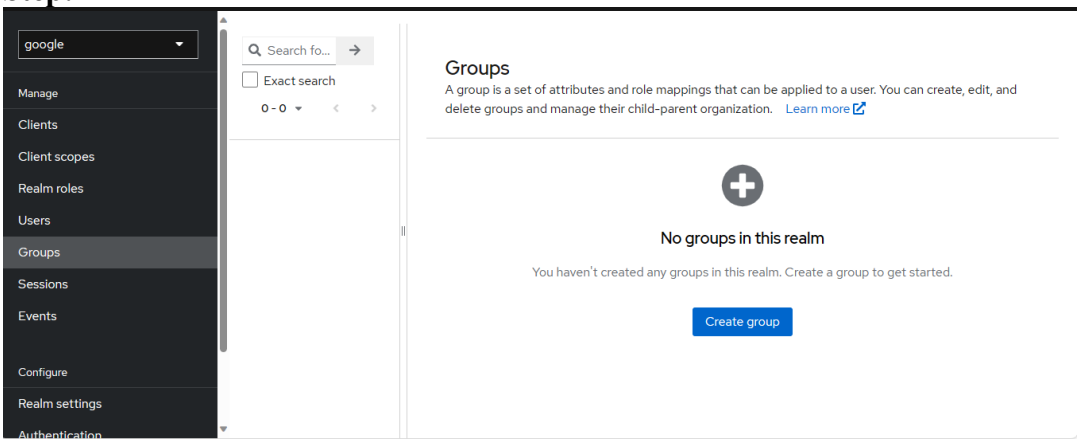
Generated access token

Generated ID token

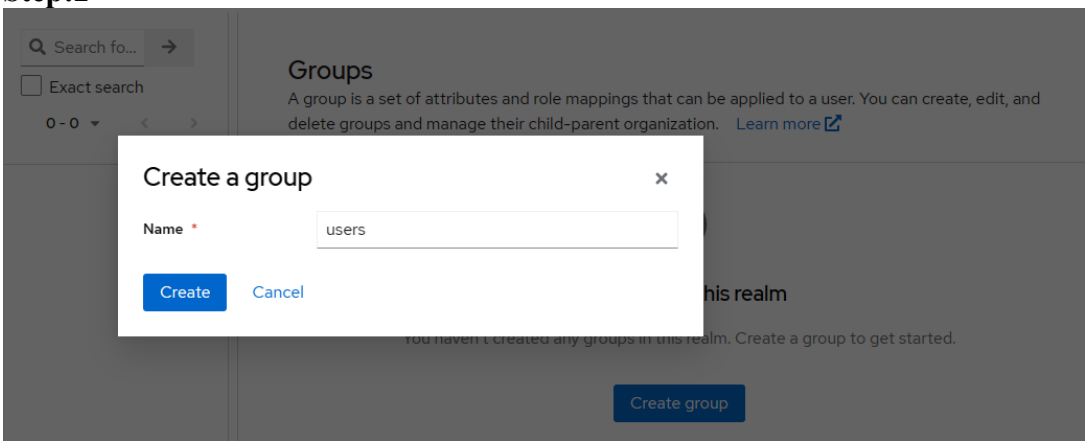
Generated user info

Setup Group and Assign Client Roles

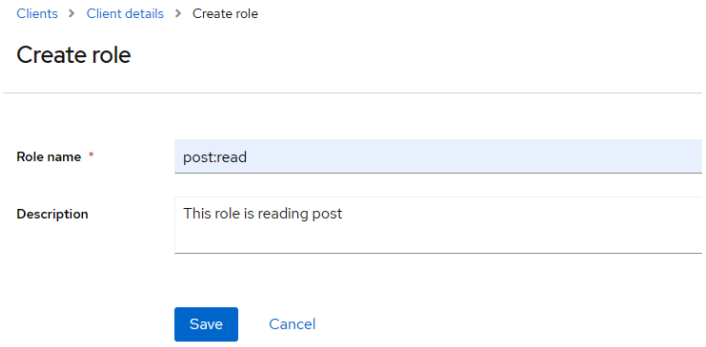
Step:1



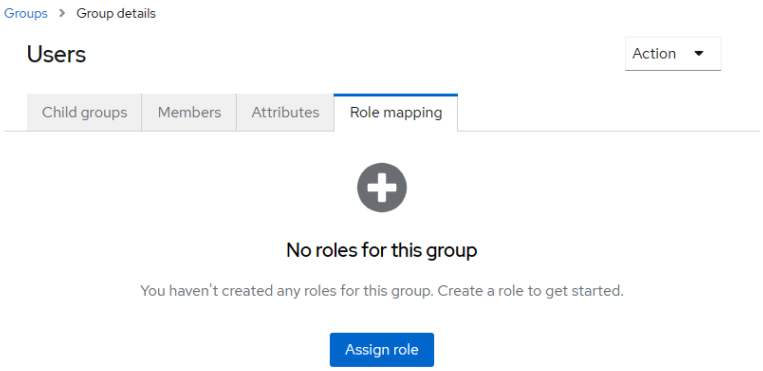
Step:2



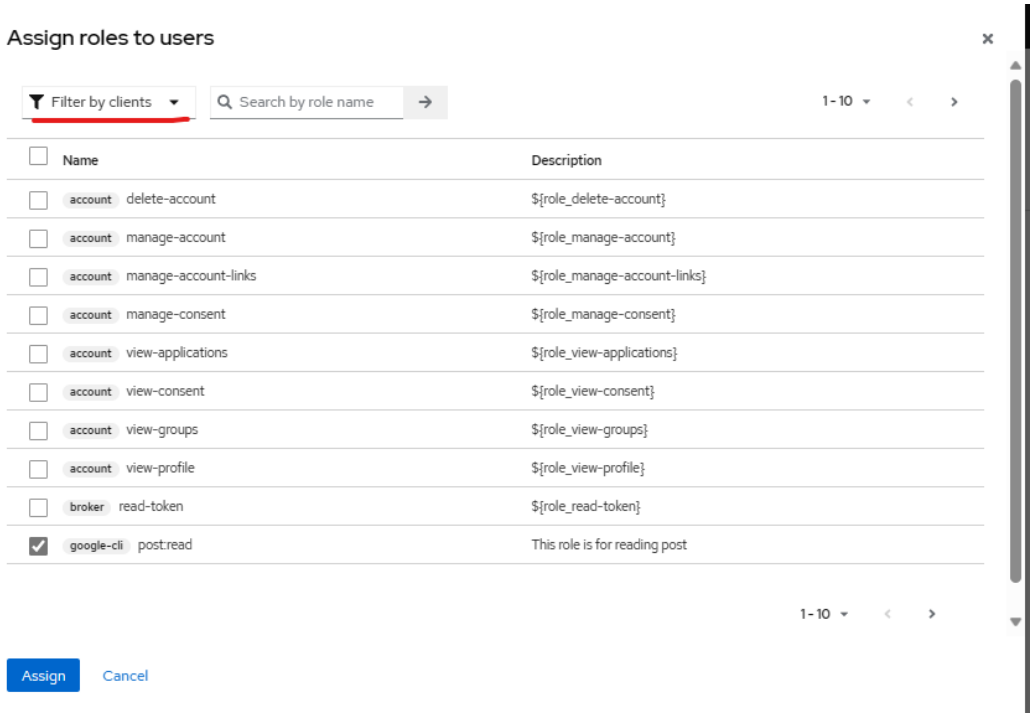
Step:3 – Create Client Role
Go to Client>{CLIENT_NAME}>Roles



Step:4 - Assign roles in group

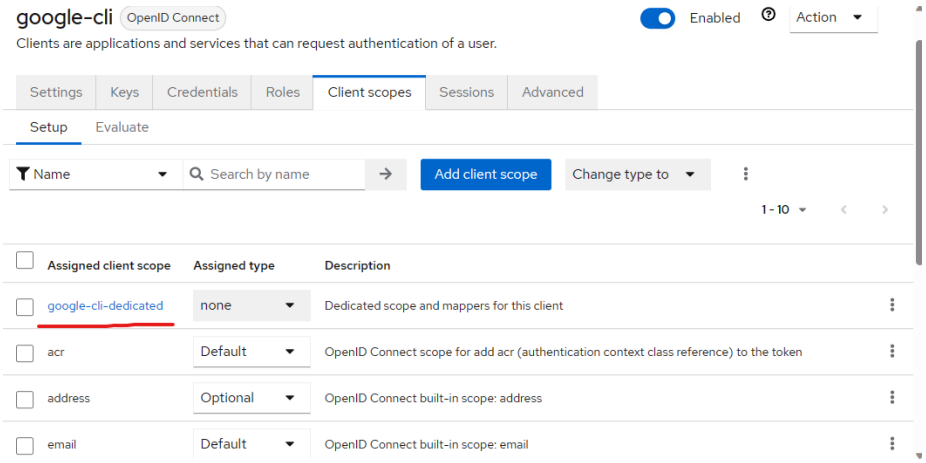


Step:5 – Adding reading post permission for this USERS group

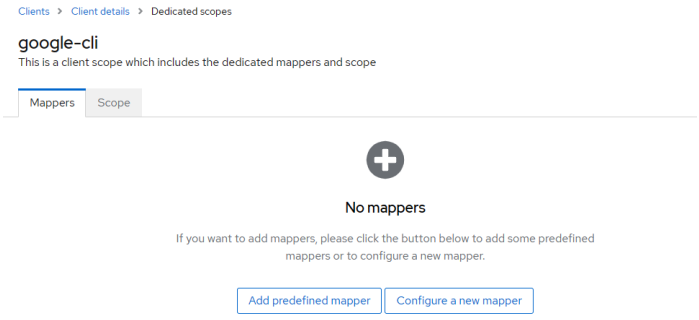


Step:6 – Adding this USERS group in the client

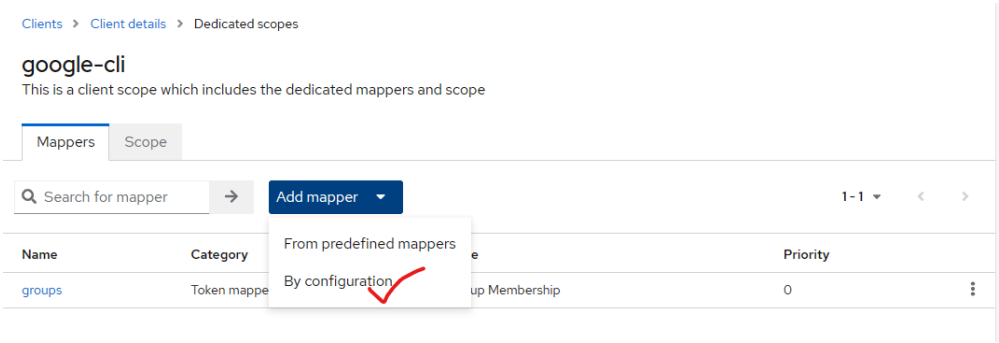
i) Go to Client>{CLIENT_NAME}>Client Scopes



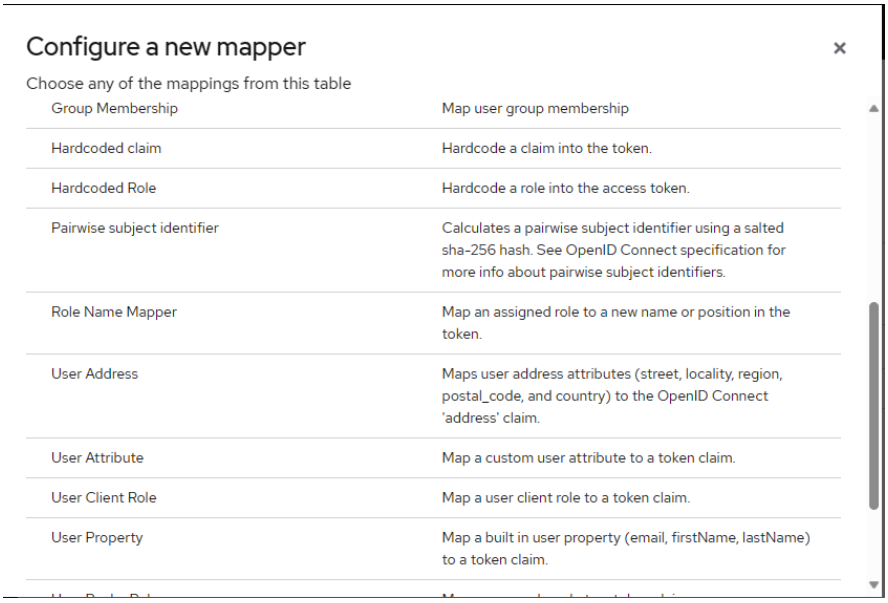
ii) Go to the google-cli-dedicated (client scope name will depend on client name)



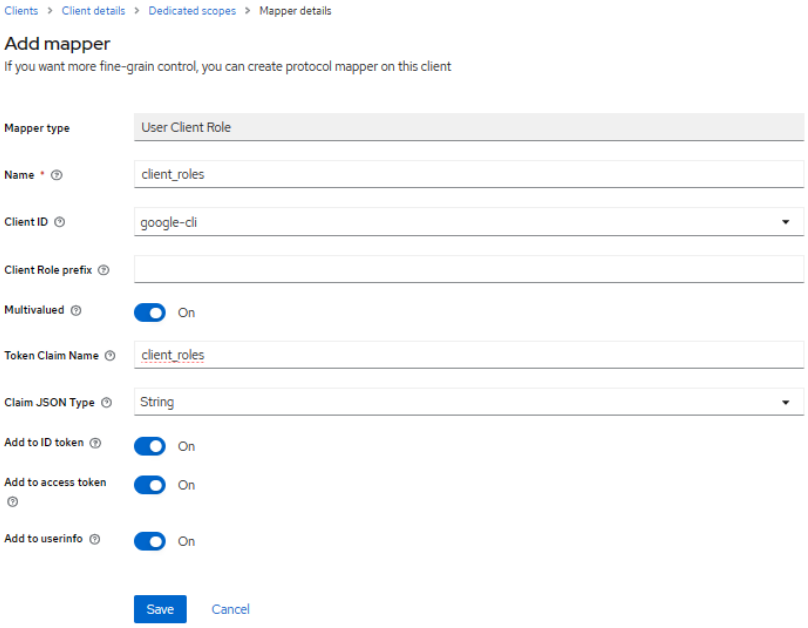
iii) If one MAPPER already exist , click on By configuration



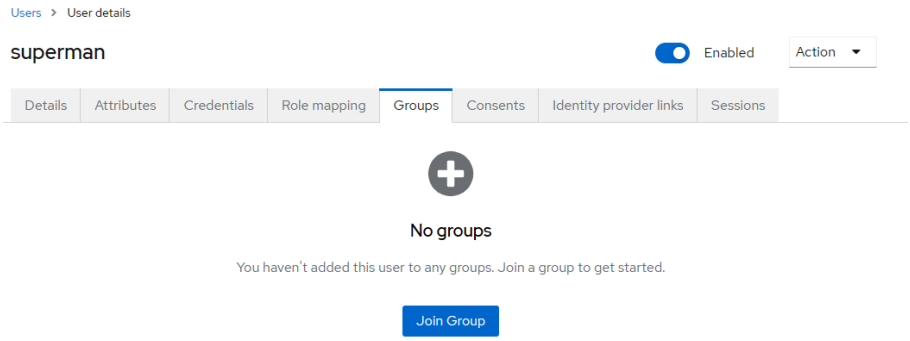
iv) Click on User Client Role



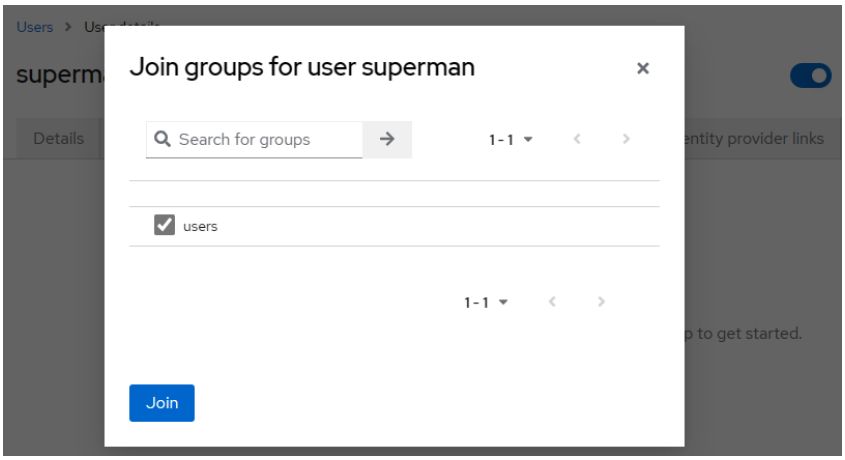
v) Cick on Save



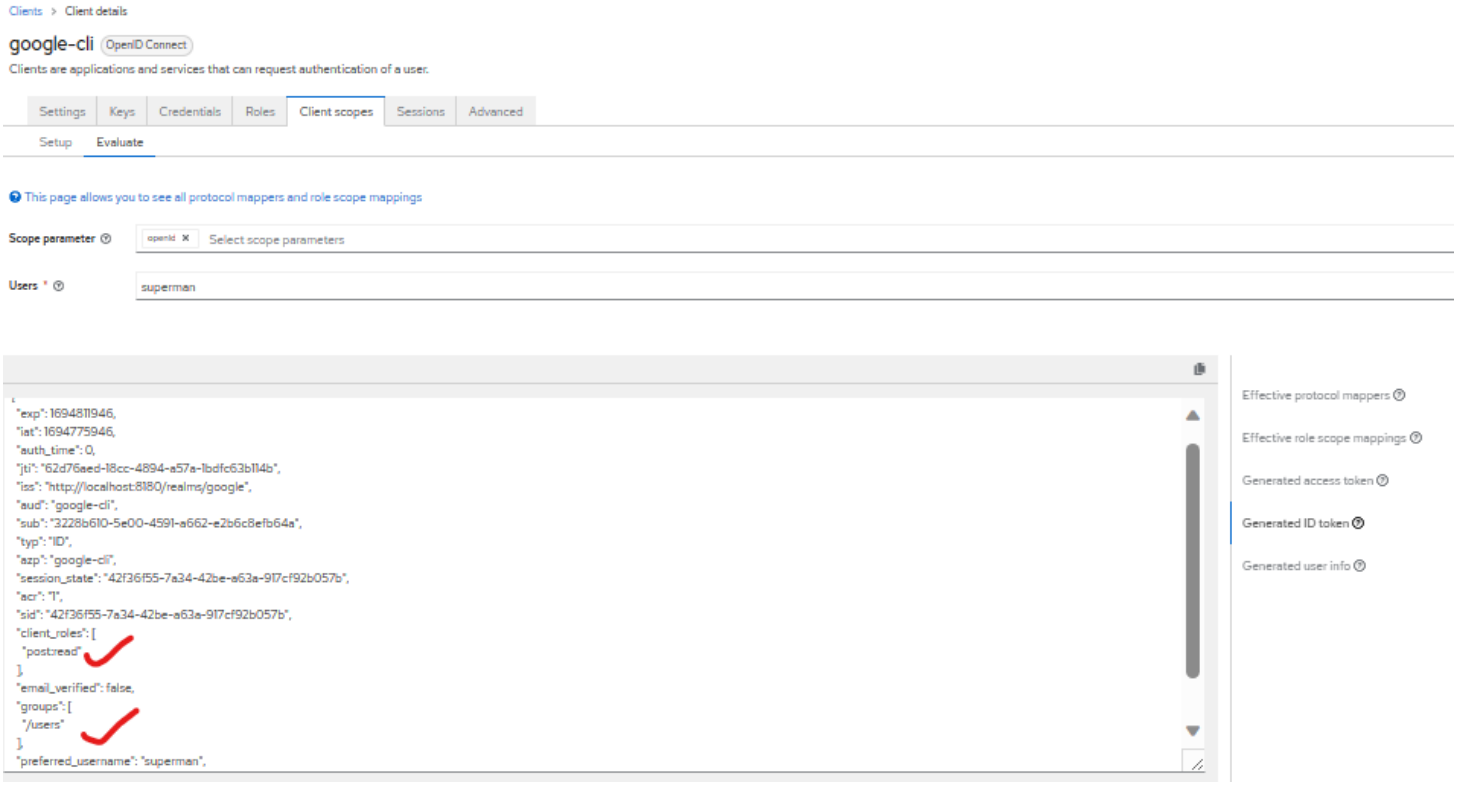
Step-6: Add the user to the USERS group. Or we can add the user from group directly.



Add group



Step-7 - Let's check that user's access token
Go to Client>{CLIENT_NAME}>Client Scopes. Then Evaluate Tab


Clients > Client details
google-cli
OpenID Connect
Clients are applications and services that can request authentication of a user.
Settings
Keys
Credentials
Roles
Client scopes
Sessions
Advanced
Setup
Evaluate
This page allows you to see all protocol mappers and role scope mappings
Scope parameter
openid
Select scope parameters
Users *
superman
{
"exp": 1694811946,
"iat": 1694775946,
"auth_time": 0,
"jti": "62d76aed-18cc-4894-a57a-1bdfc63b114b",
"iss": "http://localhost:8180/realms/google",
"aud": "google-cli",
"sub": "3228b610-5e00-4591-a662-e2b6c8efb64a",
"typ": "ID",
"azp": "google-cli",
"session_state": "42f36f55-7a34-42be-a63a-917cf92b057b",
"acr": "1",
"sid": "42f36f55-7a34-42be-a63a-917cf92b057b",
"client_roles": [
"post:read"
],
"email_verified": false,
"groups": [
"/users"
],
"preferred_username": "superman",
Effective protocol mappers
Effective role scope mappings
Generated access token
Generated ID token
Generated user info