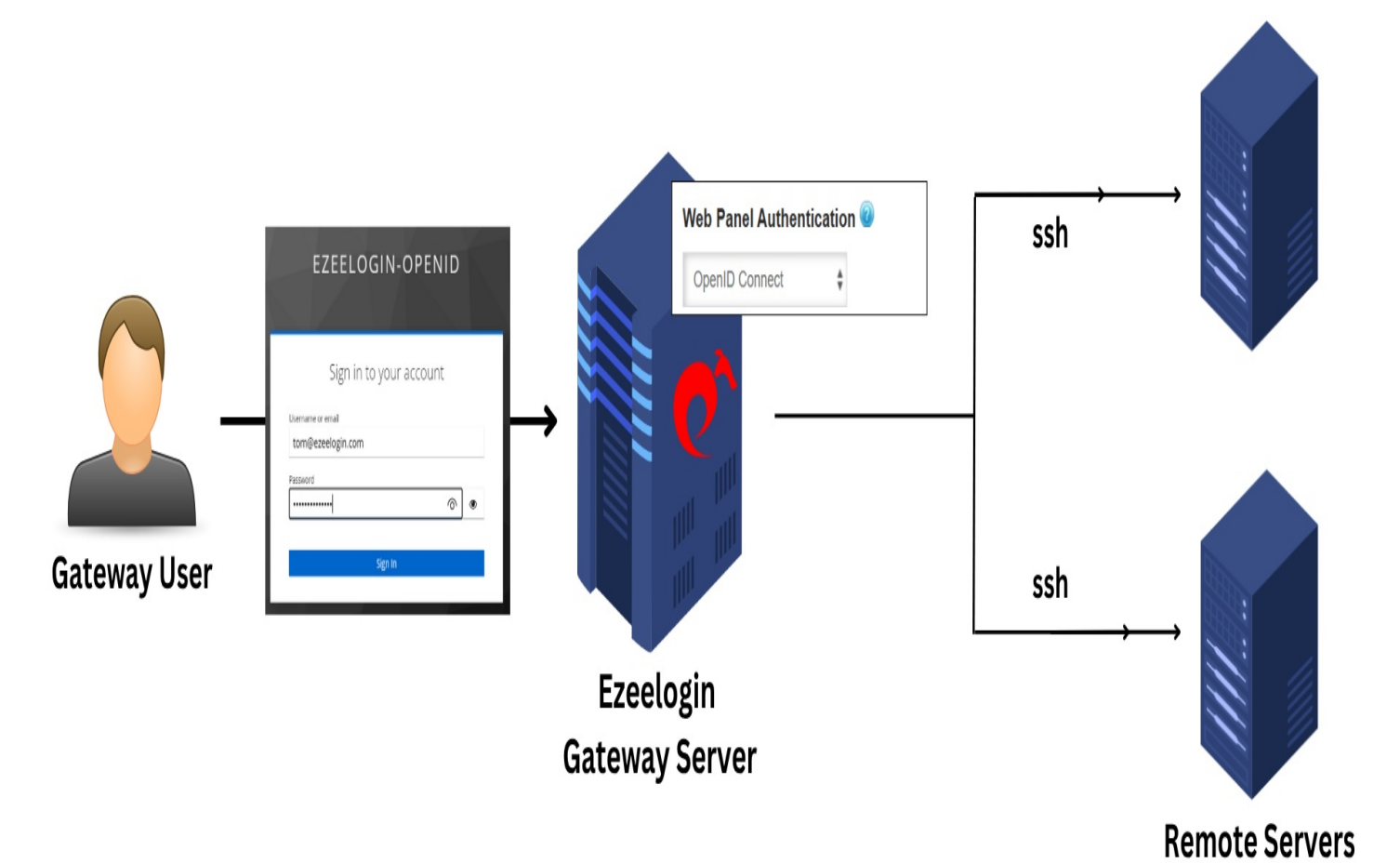


Integrating Keycloak OIDC

How to integrate Keycloak OpenID Connect with Ezeelogin Jumpserver?

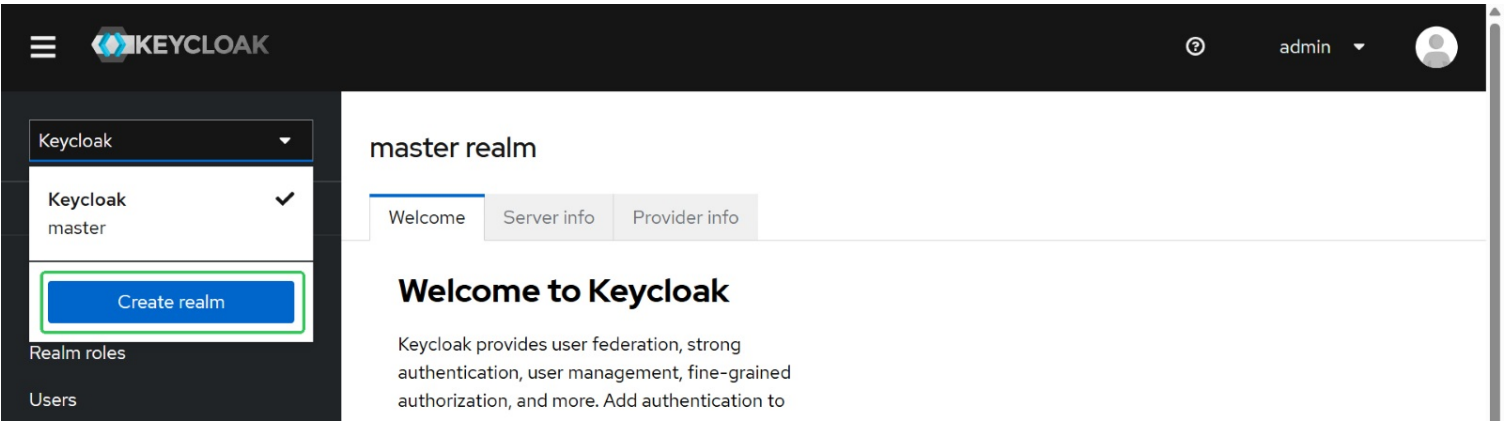
Overview: This article will help the super admin user to integrate keycloak openid with the Ezeelogin jump server.

OpenID connect is an authentication mechanism for web applications. It's based on web protocols and it cannot be used for user authentication over SSH.



Step 1: Install the latest [Keycloak](#) server.

Step 2: Click on **Create realm**.



Step 2(A): Enter the **Realm name** and click on **Create**.

KEYCLOAK admin

Keycloak

Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control.

Resource file

Drag a file here or browse to upload [Browse...](#) [Clear](#)

1

Upload a JSON file

Realm name *

Enabled ☒ On

[Create](#) [Cancel](#)

Step 3: Click on Create client -> Enter client type(openid) and client id(eg- ezeelogin-sso) ->Next

KEYCLOAK admin

Ezeelogin-openid

Manage

Clients

Client scopes

Realm roles

Users

Groups

Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

[Clients list](#) [Initial access token](#) [Client registration](#)

[Create client](#) [Import client](#) [Refresh](#) 1-6

Client ID	Name	Type	Description	Home URL
account	\${client_account}	OpenID Connect	—	http://192.168.0.107:8080/realms/Ezeelogin-openid/account/

KEYCLOAK admin

Ezlogin-oidc

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.

1 General settings

2 Capability config

3 Login settings

Client type

Client ID *

Name

Description

Always display in UI ☒ On

[Next](#) [Back](#) [Cancel](#)

Step 3(A): Enable client authentication -> Next.

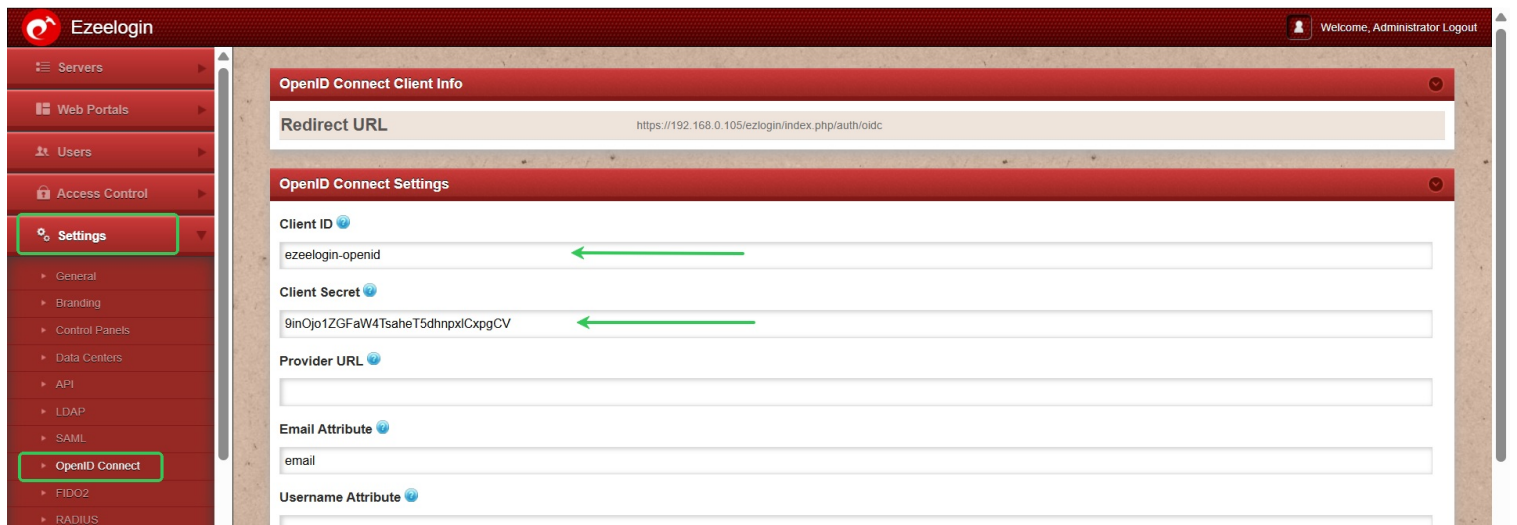
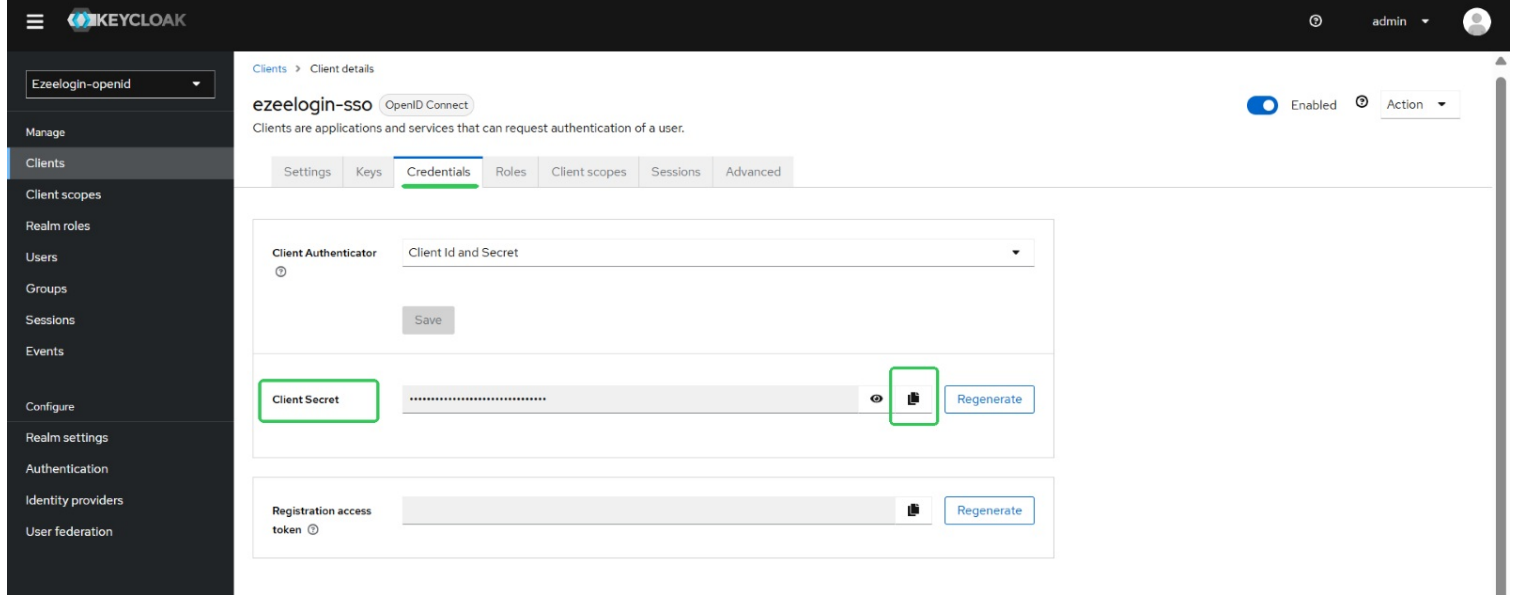
Keycloak interface showing the 'Create client' page. The 'Client authentication' toggle is set to 'On'. The 'Next' button is highlighted.

Step 3(B): Enter Root URL(eg:https://192.168.1.4/ezlogin) -> Valid redirect URIs(copy paste from ezeelogin settings) -> Save it.

Keycloak interface showing the 'Create client' page, 'Login settings' tab. The 'Root URL' and 'Valid redirect URIs' fields are filled with the values from the previous step. The 'Save' button is highlighted.

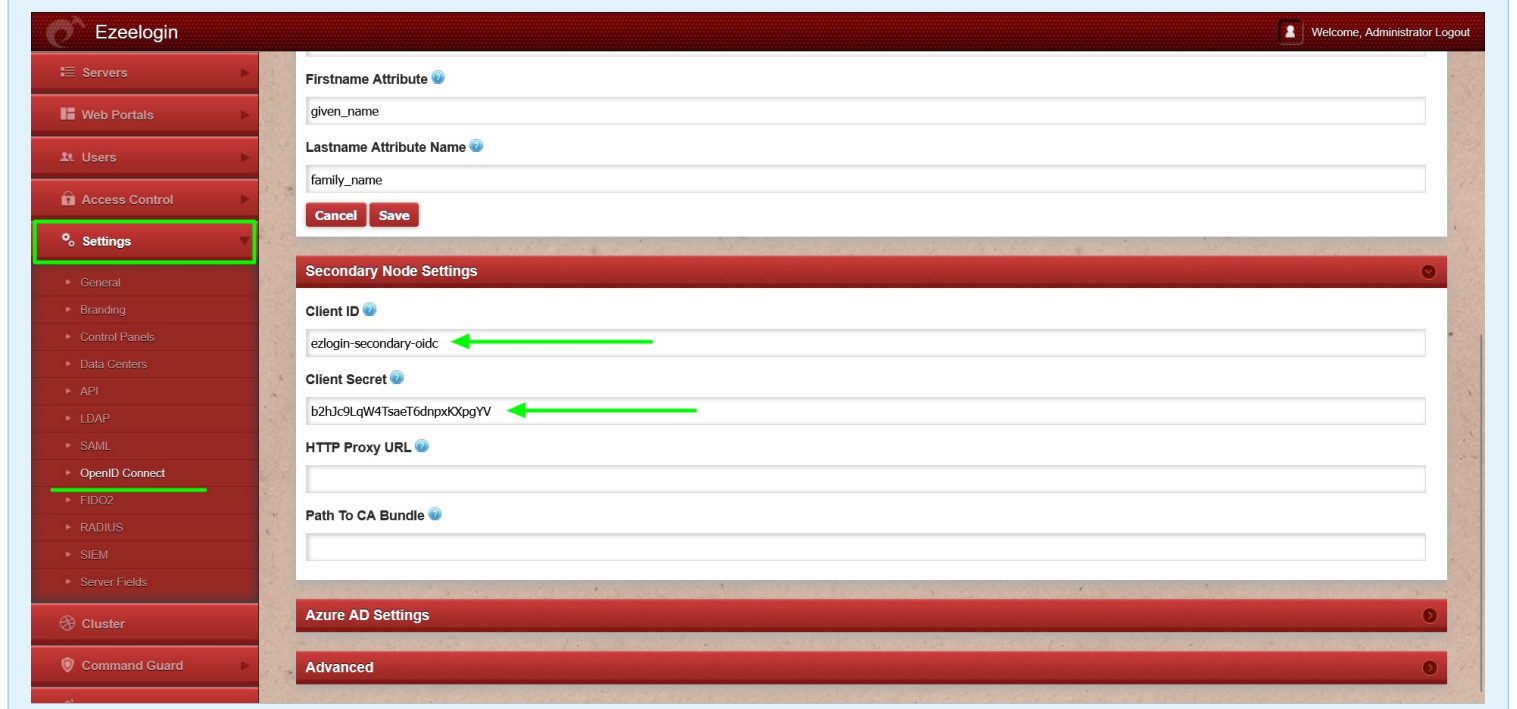
Ezeelogin interface showing the 'Settings' page. The 'OpenID Connect' section is expanded, and the 'Redirect URL' field is highlighted with the same URL as in the previous steps.

Step 4: Click on Clients (in the same section after saving client) -> Credentials tab -> copy Client Secret and paste it in ezeelogin gui.

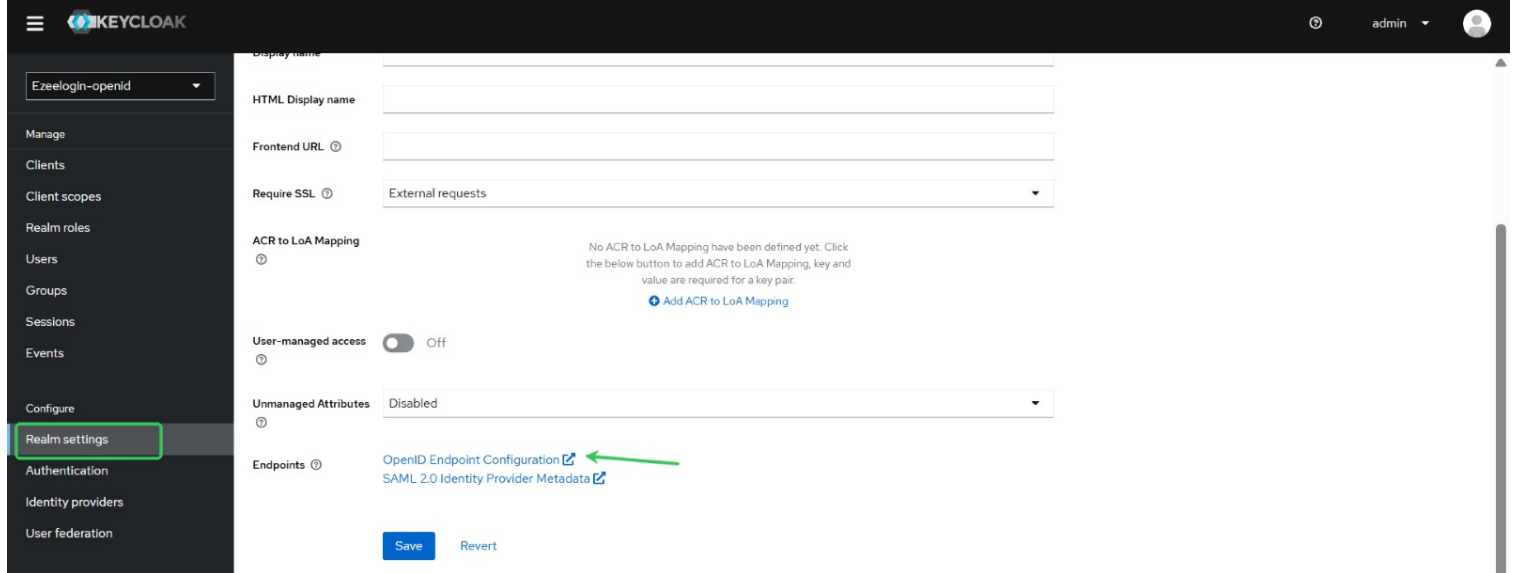


For cluster, add the Client ID and Client Secret from the other OIDC configuration:

Follow steps 1 to 4 to create new application for secondary node. While adding the application details in Keycloak, replace the IP address with the secondary node's IP address.



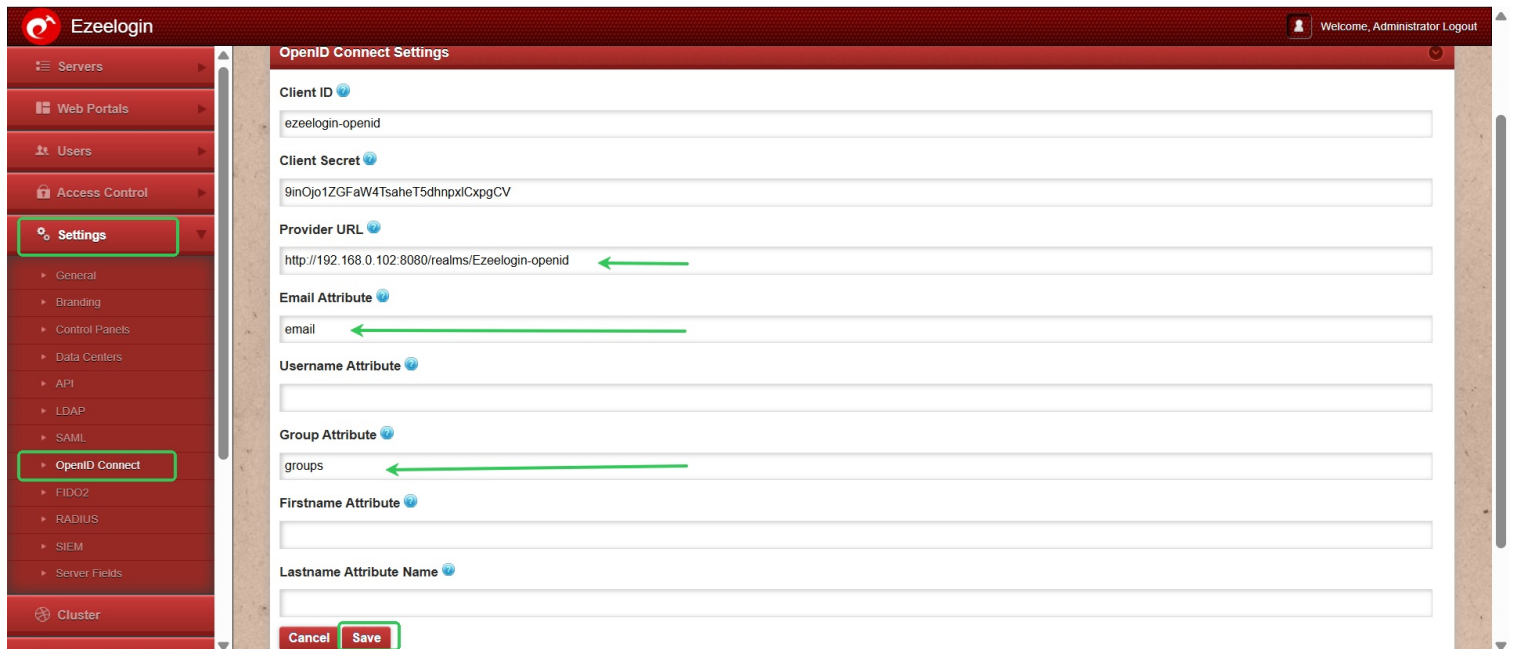
Step 5: Under Realm settings -> Endpoints-> click on the **openid endpoint configuration url** -> copy issuer url and paste in ezeelogin gui under provider url and provide the attribute values -> save it.



The screenshot shows the Keycloak Admin Console interface. On the left is a dark sidebar with navigation options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, **Realm settings** (highlighted with a green box), Authentication, Identity providers, and User federation. The main content area is titled 'Ezeelogin-openid' and contains several configuration fields: 'HTML Display name' (empty), 'Frontend URL' (empty), 'Require SSL' (set to 'External requests'), 'ACR to LoA Mapping' (with a message: 'No ACR to LoA Mapping have been defined yet. Click the below button to add ACR to LoA Mapping, key and value are required for a key pair.' and a link 'Add ACR to LoA Mapping'), 'User-managed access' (toggle set to 'Off'), 'Unmanaged Attributes' (set to 'Disabled'), and 'Endpoints' (containing links for 'OpenID Endpoint Configuration' and 'SAML 2.0 Identity Provider Metadata', both highlighted with green arrows). At the bottom are 'Save' and 'Revert' buttons.

Pretty-print ☐

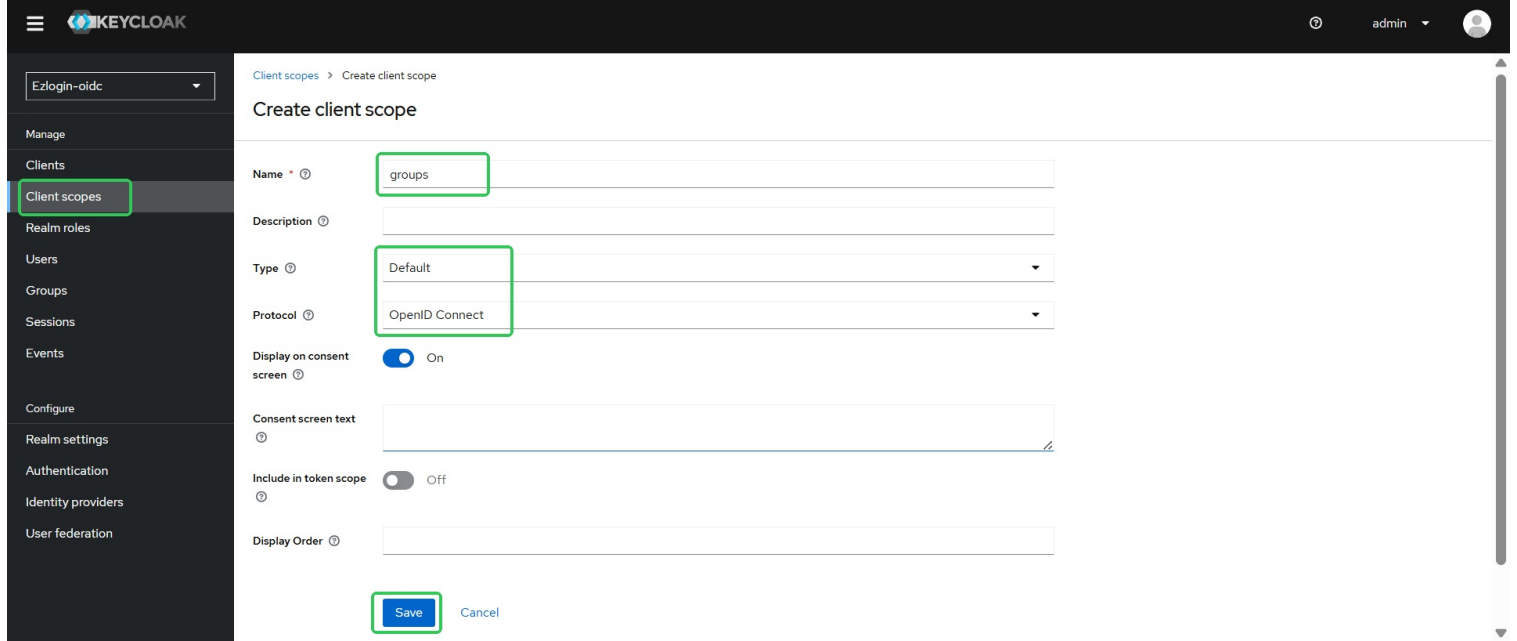
```
{
  "issuer": "http://192.168.0.102:8080/realms/Ezlogin-oidc",
  "authorization_endpoint": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/auth",
  "token_endpoint": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/token",
  "introspection_endpoint": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/token/introspect",
  "userinfo_endpoint": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/userinfo",
  "end_session_endpoint": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/logout",
  "frontchannel_logout_session_supported": true,
  "frontchannel_logout_supported": true,
  "jwks_uri": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/certs",
  "check_session_iframe": "http://192.168.0.102:8080/realms/Ezlogin-oidc/protocol/openid-connect/login-status-iframe.html",
  "grant_types_supported": [
    "authorization_code",
    "implicit",
    "password",
    "refresh_token"
  ]
}
```



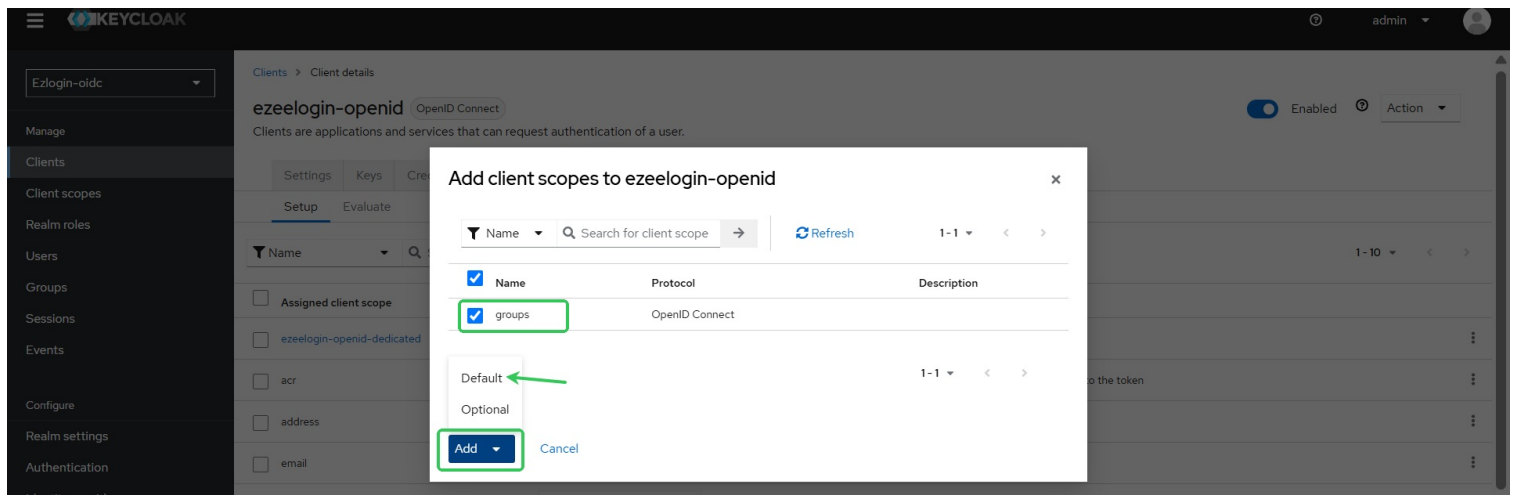
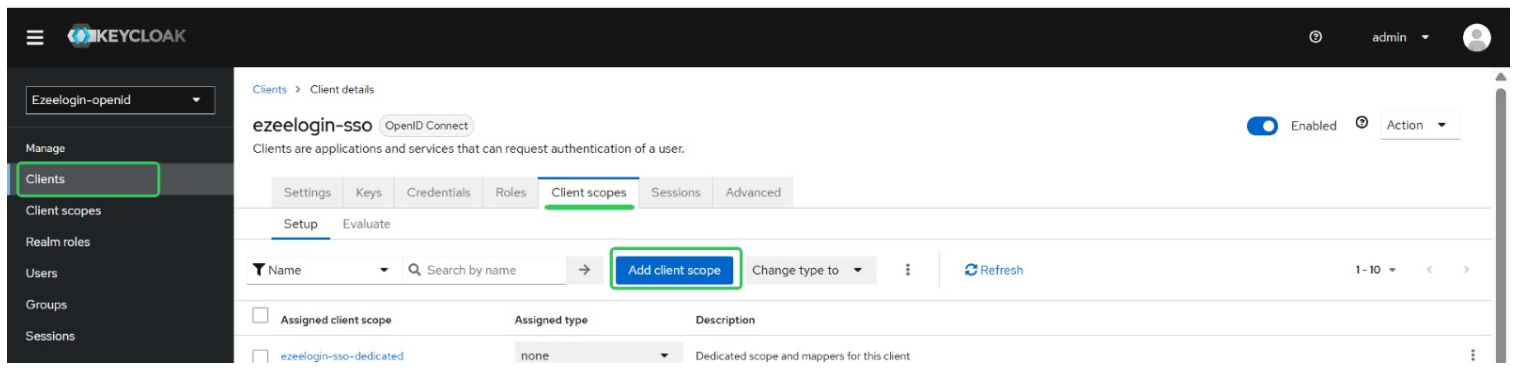
The screenshot shows the Ezeelogin OpenID Connect Settings interface. On the left is a red sidebar with navigation options: Servers, Web Portals, Users, Access Control, **Settings** (highlighted with a green box), and a list of protocols: General, Branding, Control Panels, Data Centers, API, LDAP, SAML, **OpenID Connect** (highlighted with a green box), FIDO2, RADIUS, SIEM, and Server Fields. The main content area is titled 'OpenID Connect Settings' and contains several configuration fields: 'Client ID' (set to 'ezeelogin-openid'), 'Client Secret' (set to '9inOjo1ZGFaW4TsaheT5dhnpxlCxpGCV'), 'Provider URL' (set to 'http://192.168.0.102:8080/realms/Ezeelogin-openid', highlighted with a green arrow), 'Email Attribute' (set to 'email', highlighted with a green arrow), 'Username Attribute' (empty), 'Group Attribute' (set to 'groups', highlighted with a green arrow), 'Firstname Attribute' (empty), and 'Lastname Attribute Name' (empty). At the bottom are 'Cancel' and 'Save' buttons, with 'Save' highlighted with a green box.

Step 6: Create client scope for groups if it not exist in client scopes.

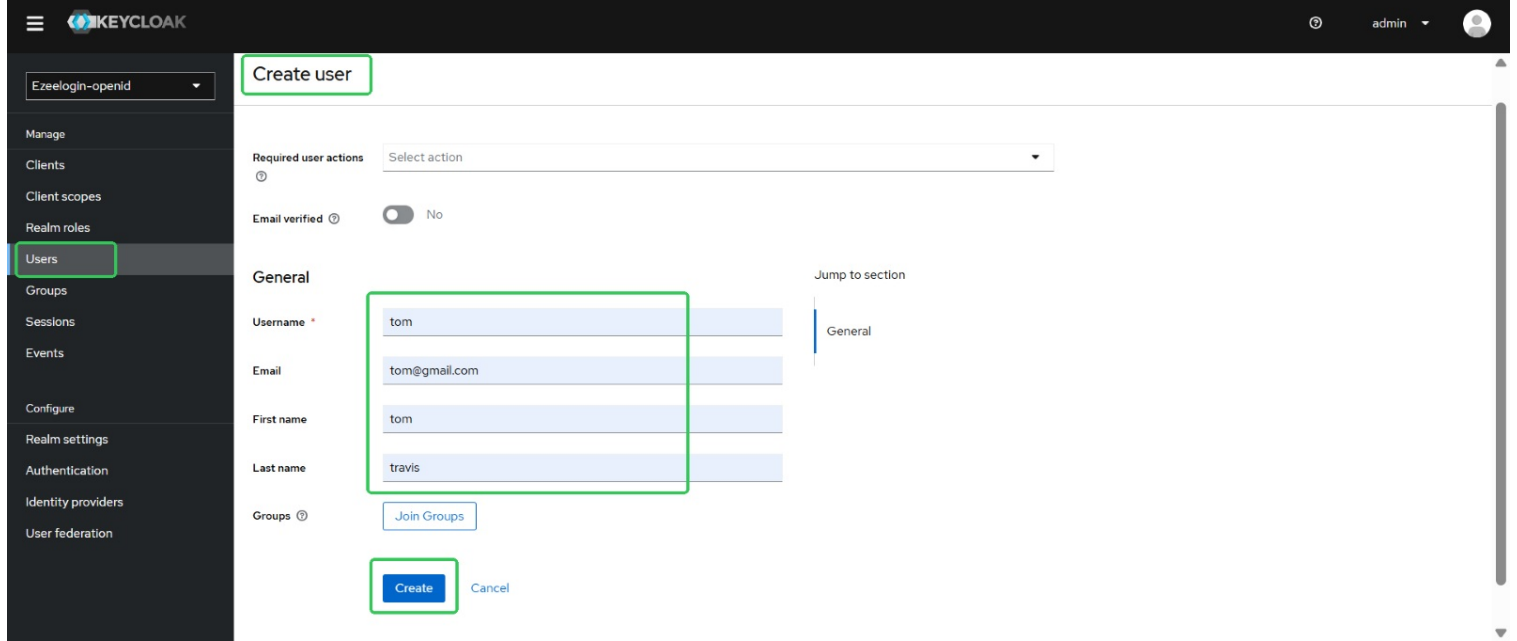
Click on Client scopes --> create client scope -> Enter the name and type --> mention the Protocol(OpenID Connect) -> save it.



Step 7: Under Clients tab -> select client -> Client scopes -> Add client scope -> select groups scope and add it .

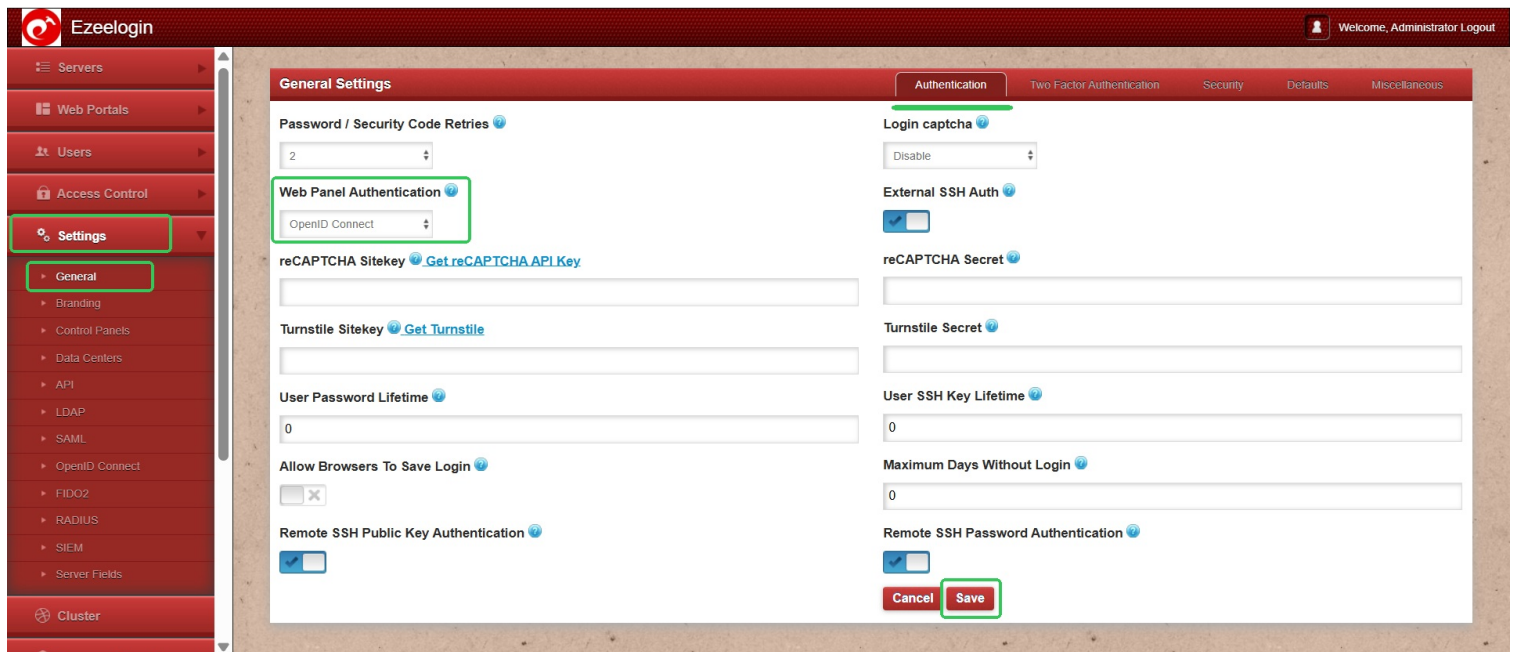


Step 8: Click on Users tab -> add user -> provide Username, Email, Firstname and Lastname -> Create. Click on Credentials tab and set the password for the user(disable temporary).



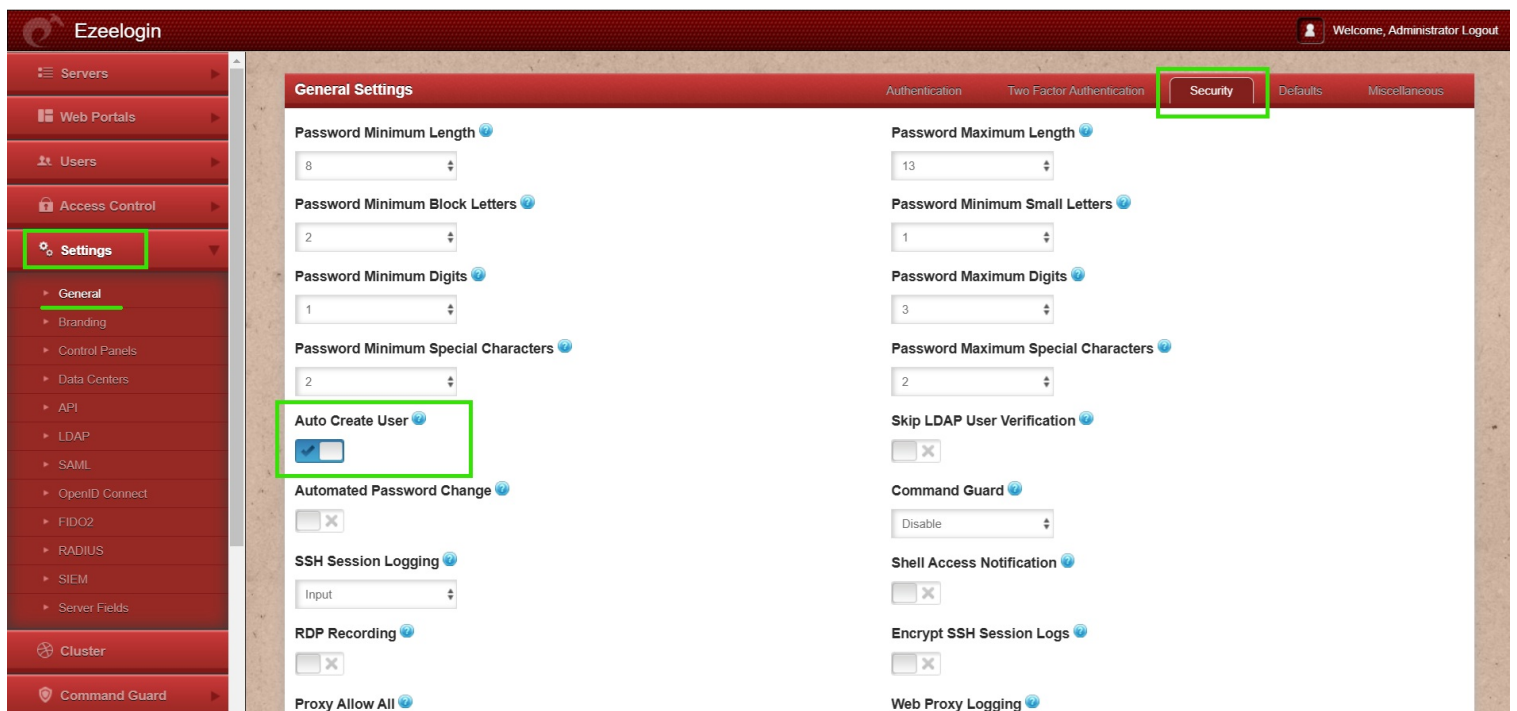
The image shows the Keycloak administration interface. On the left is a dark sidebar with a menu including 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users' (highlighted with a green box), 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication', 'Identity providers', and 'User federation'. The main content area is titled 'Create user' (also highlighted with a green box). It features a 'Required user actions' dropdown set to 'Select action' and an 'Email verified' toggle set to 'No'. Under the 'General' tab, there are input fields for 'Username' (tom), 'Email' (tom@gmail.com), 'First name' (tom), and 'Last name' (travis), all of which are grouped by a green box. Below these is a 'Groups' section with a 'Join Groups' button. At the bottom, there are 'Create' and 'Cancel' buttons, with the 'Create' button highlighted by a green box. A 'Jump to section' dropdown is set to 'General'.

Step 9: Now change auth type in ezeelogin to OpenID Connect and login.



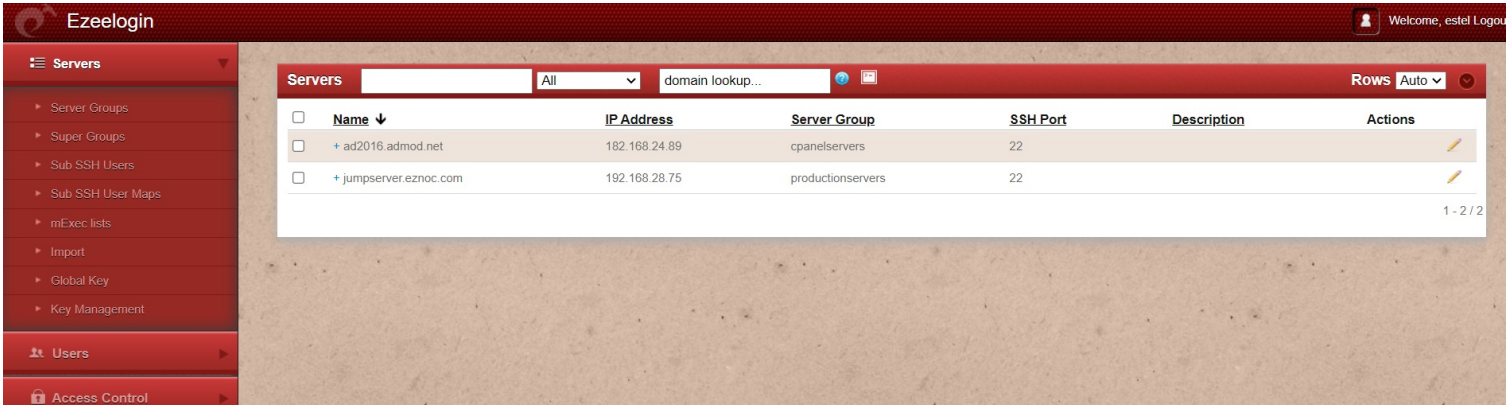
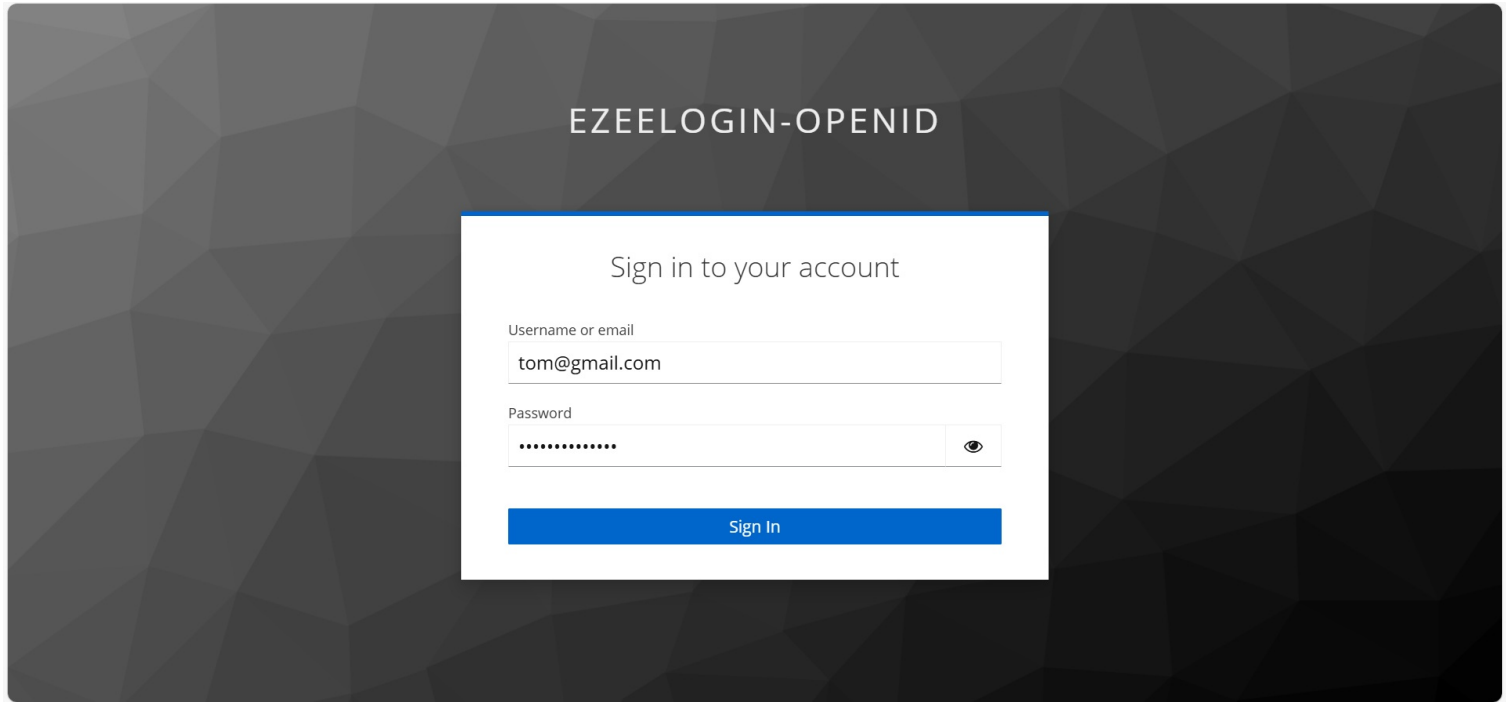
The image shows the Ezeelogin administration interface. The left sidebar has a red theme with a menu including 'Servers', 'Web Portals', 'Users', 'Access Control', 'Settings' (highlighted with a green box), 'General' (highlighted with a green box), 'Branding', 'Control Panels', 'Data Centers', 'API', 'LDAP', 'SAML', 'OpenID Connect', 'FIDO2', 'RADIUS', 'SIEM', 'Server Fields', 'Cluster', and 'Command Guard'. The main content area is titled 'General Settings' and has tabs for 'Authentication', 'Two Factor Authentication', 'Security', 'Defaults', and 'Miscellaneous'. The 'Authentication' tab is active. It contains various settings: 'Password / Security Code Retries' (2), 'Web Panel Authentication' (OpenID Connect, highlighted with a green box), 'reCAPTCHA Sitekey', 'Turnstile Sitekey', 'User Password Lifetime' (0), 'Allow Browsers To Save Login' (disabled), 'Remote SSH Public Key Authentication' (checked), 'Login captcha' (Disable), 'External SSH Auth' (checked), 'reCAPTCHA Secret', 'Turnstile Secret', 'User SSH Key Lifetime' (0), 'Maximum Days Without Login' (0), and 'Remote SSH Password Authentication' (checked). At the bottom right, there are 'Cancel' and 'Save' buttons, with the 'Save' button highlighted by a green box.

Step 10: Enable Auto Create User in the Ezeelogin GUI by going to **Settings -> General -> Security**.



The image shows the Ezeelogin administration interface, specifically the 'Security' tab under 'General Settings'. The left sidebar is the same as in the previous image, with 'Settings' and 'General' highlighted. The 'Security' tab is now active and highlighted with a green box. It contains various password and security settings: 'Password Minimum Length' (8), 'Password Minimum Block Letters' (2), 'Password Minimum Digits' (1), 'Password Minimum Special Characters' (2), 'Auto Create User' (checked, highlighted with a green box), 'Automated Password Change' (disabled), 'SSH Session Logging' (Input), 'RDP Recording' (disabled), 'Proxy Allow All' (disabled), 'Password Maximum Length' (13), 'Password Minimum Small Letters' (1), 'Password Maximum Digits' (3), 'Password Maximum Special Characters' (2), 'Skip LDAP User Verification' (disabled), 'Command Guard' (Disable), 'Shell Access Notification' (disabled), 'Encrypt SSH Session Logs' (disabled), and 'Web Proxy Logging' (disabled).

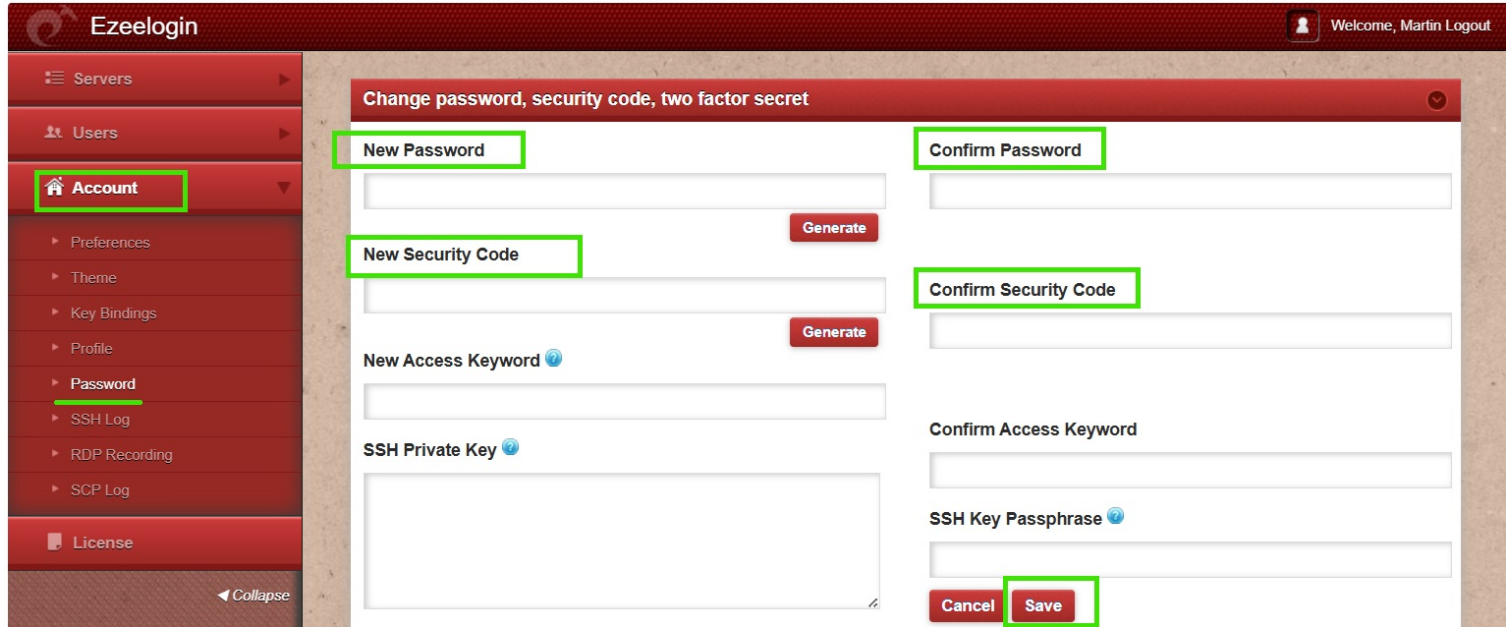
Step 11: Login to the Ezeelogin GUI with OpenID connect authentication



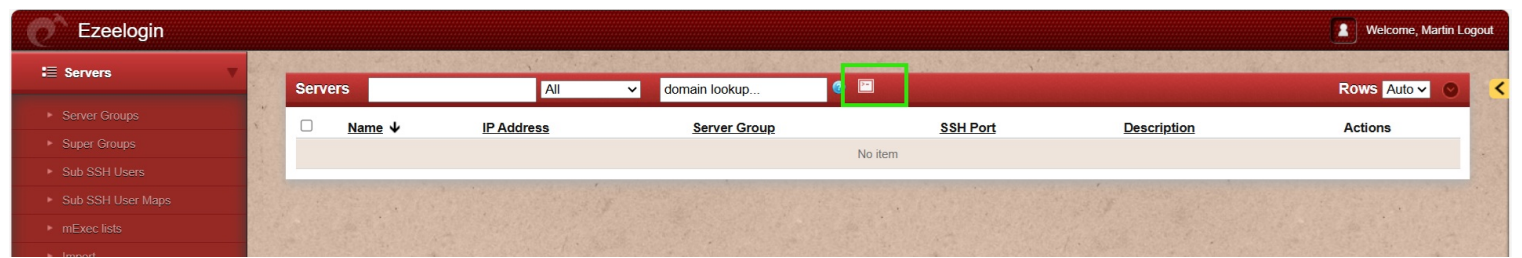
For gateway users who require administrative access, the password must be set manually from the Ezeelogin GUI. Refer below article:

Error: Invalid password

After logging in, set a new password and security code under **Account -> Password**.



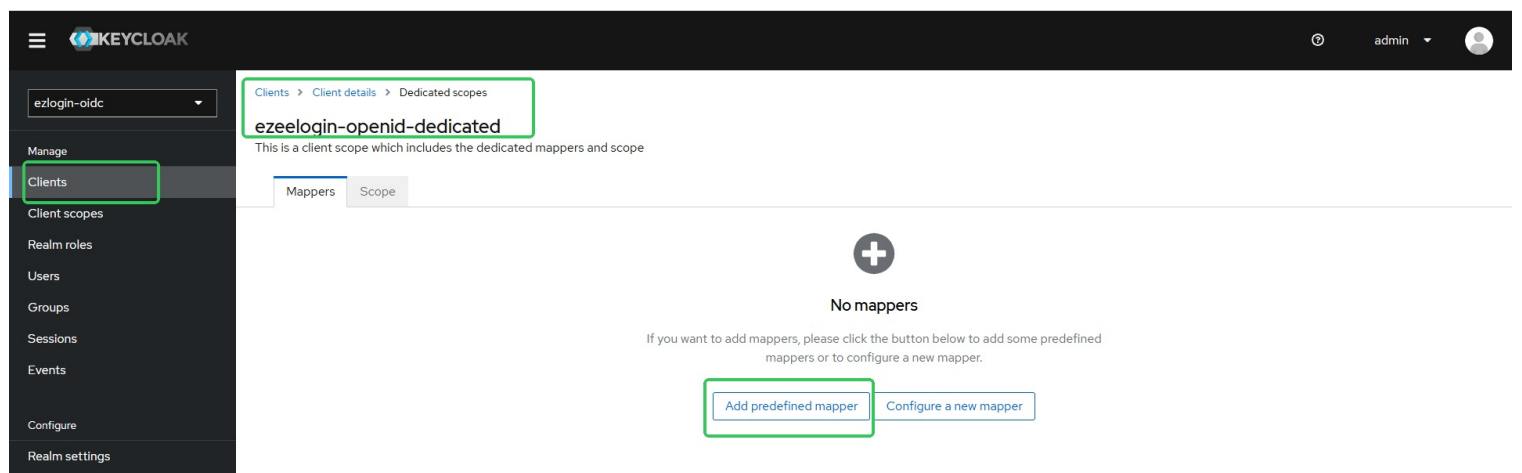
Step 12: Also, try logging in to the Ezeelogin shell using **WebSSH** (refer the below screenshot) or any SSH client such as PuTTY or a terminal.



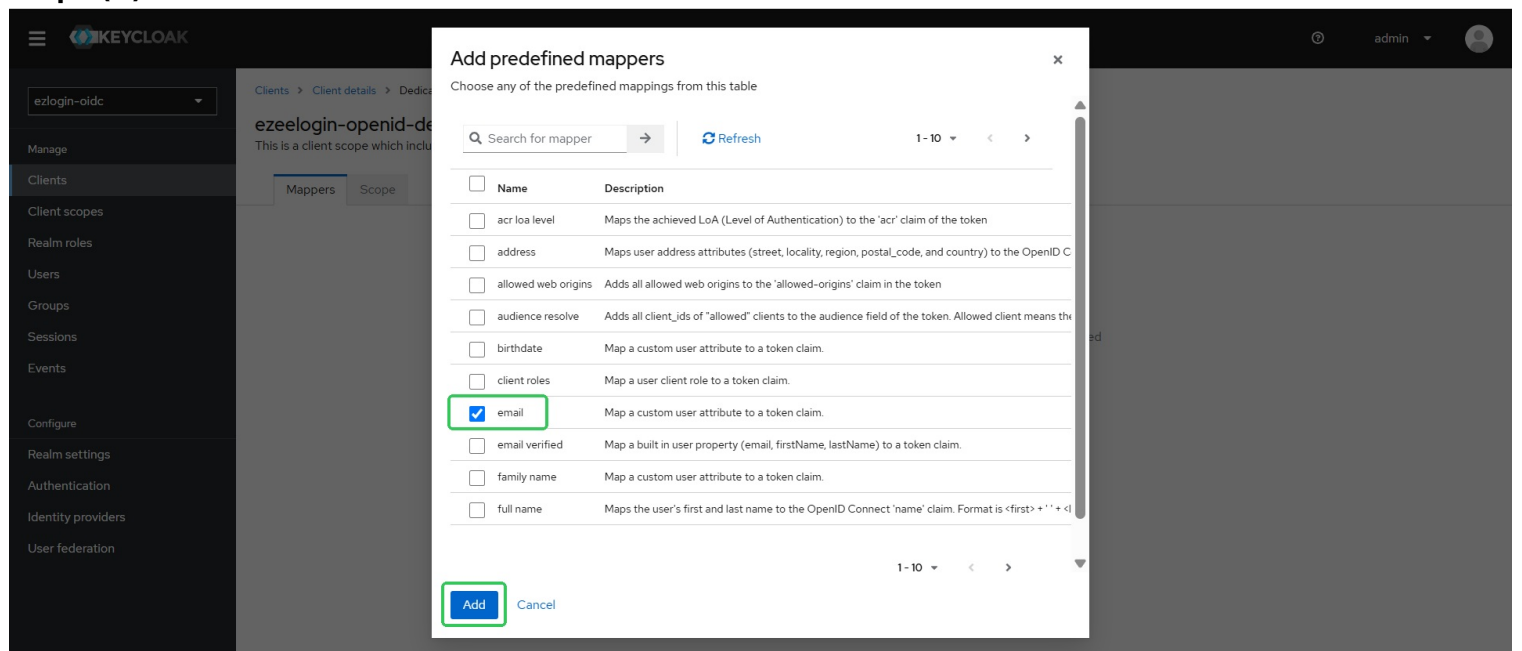
After setting the new password and security code, try logging in using the updated credentials.

How to map a user from keycloak OpenID Connect user group to Ezeelogin user group?

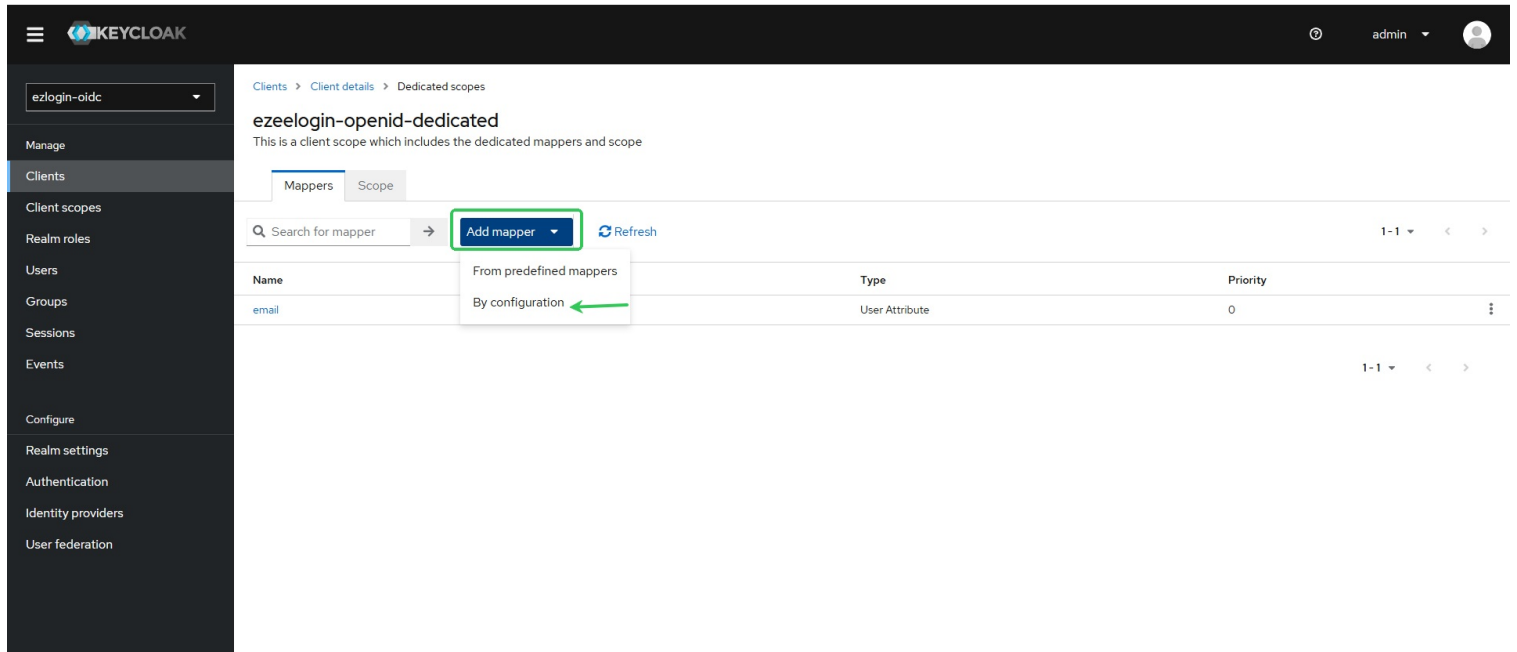
Step 1: Click on Clients > Select the clients > Click on Add predefined mapper.



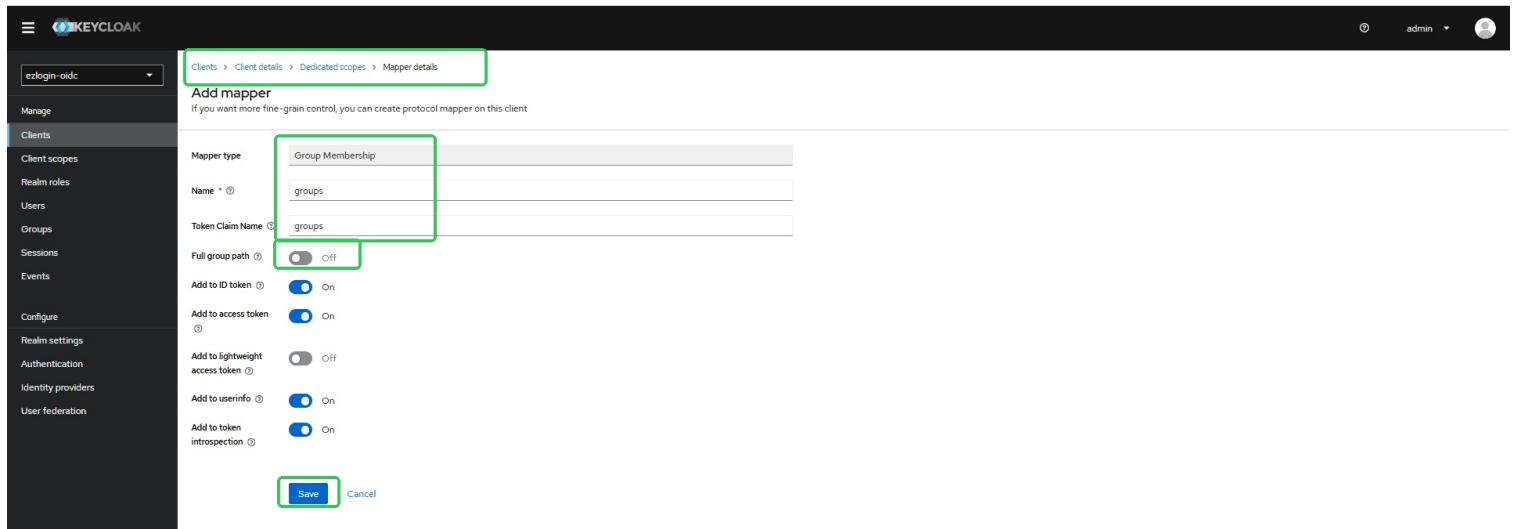
Step 1(A): Click on email and Add it.



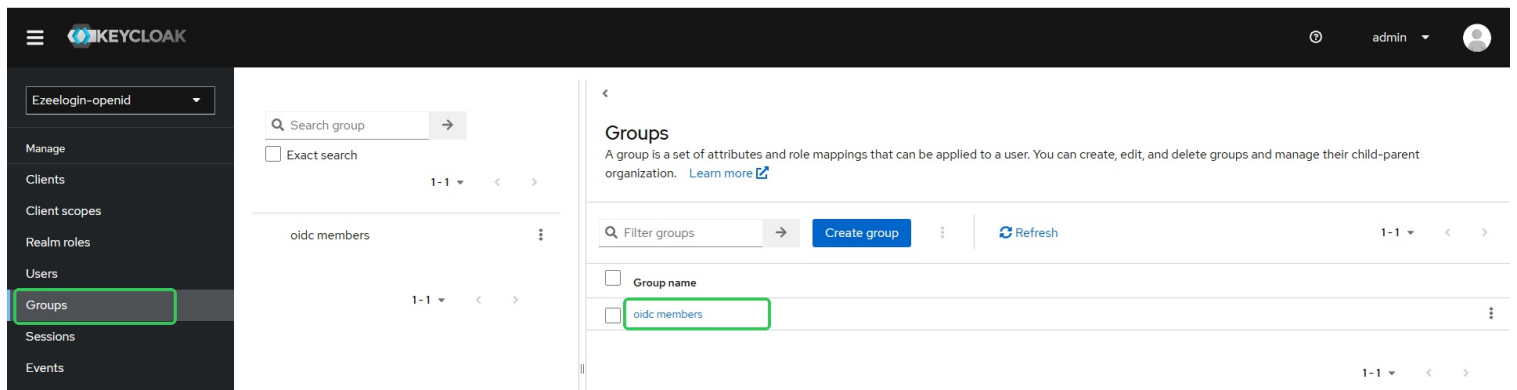
Step 2: Add mapper > Click on By configuration > select By Configuration option(Group Memebership)

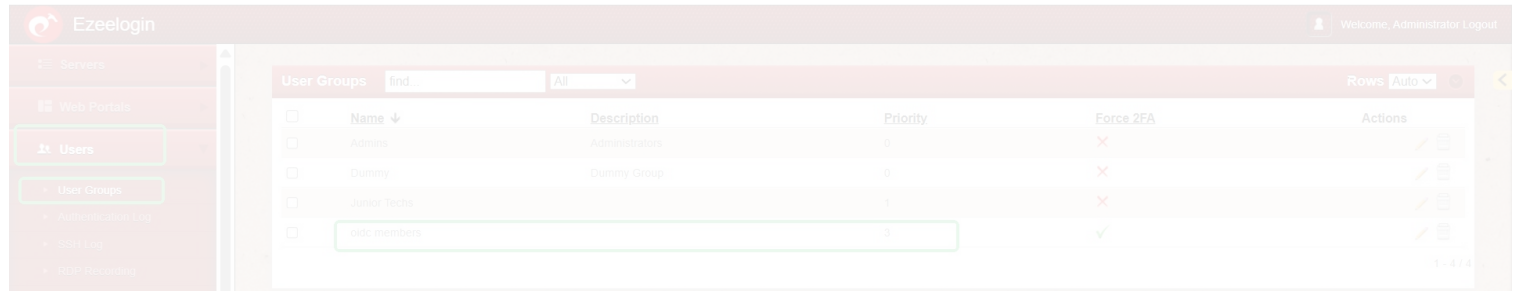
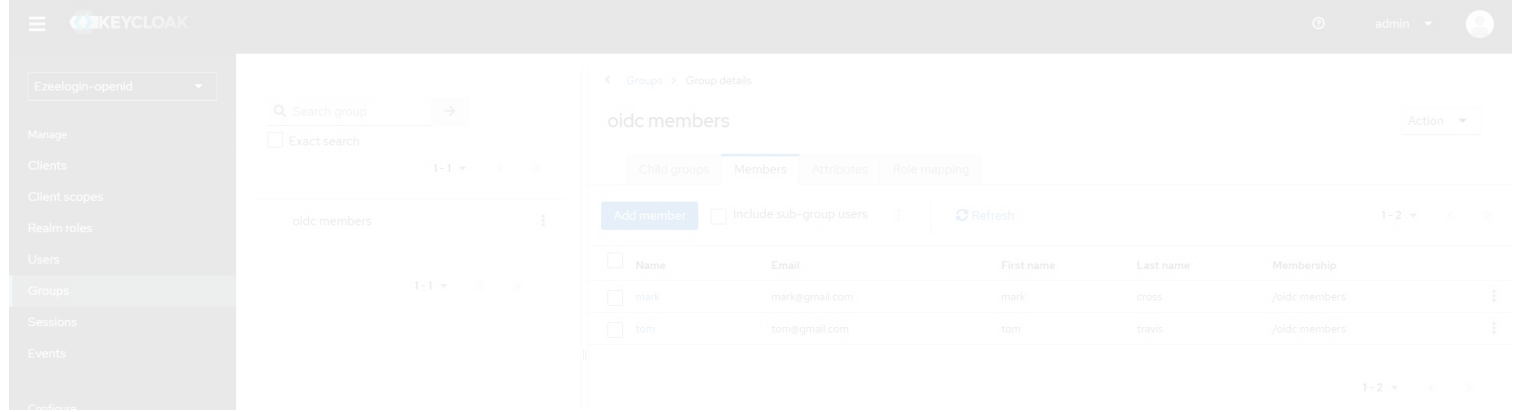


Step 2(A): Add the Name and Token Claim Name and disable Full group path and Save it.

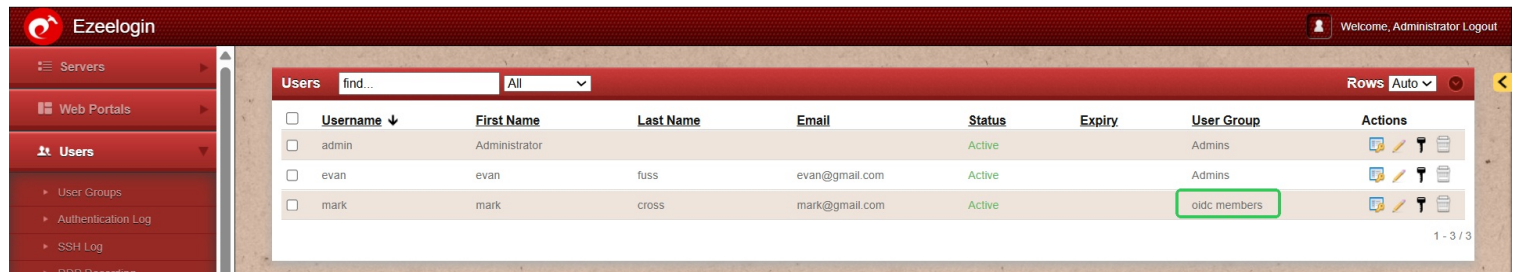


Step 3: Create the same user group in Ezeelogin as in Keycloak by setting the group priority. And add those keycloak users in same group.





Step 4: Now try to rellogin and confirm the group map.



Related Articles:

[Authentication with OpenID Connect.](#)

[Integrate Okta OpenID Connect.](#)

[Integrate OneLogin OpenID Connect.](#)

[Integrate Azure OpenID Connect.](#)

[Integrate Jumpcloud OpenID Connect.](#)

Online URL: <https://www.ezeelogin.com/kb/article/integrating-keycloak-oidc-777.html>