# Practical Exercise: Associating Roles with User Store Groups

## Training Objective

Learn how to associate user store groups to permissions inside WSO2 API Manager and transform them into roles.

## Business Scenario

PizzaShack wants to reuse existing user store groups to maintain access control on its API Management platform.

## High-Level Steps

- Associate permissions to existing groups

## Detailed Instructions

Create three new LDAP groups by creating new roles on WSO2 API Manager:

| **Role** | **Permissions** |
|---|---|
| Architect | Admin Permissions->Login<br>Admin Permissions->Manage->Search<br>Admin Permissions->Manage->Manage Tiers<br>Admin Permissions->Manage->API<br>Admin Permissions->Manage->Resources |
| Developer | Admin Permissions->Login<br>Admin Permissions->Manage->Search<br>Admin Permissions->Manage->API->Subscribe<br>Admin Permissions->Manage->API->Create<br>Admin Permissions->Manage->Resources |
| Admin | All Permissions |

## Managing Role Mappings

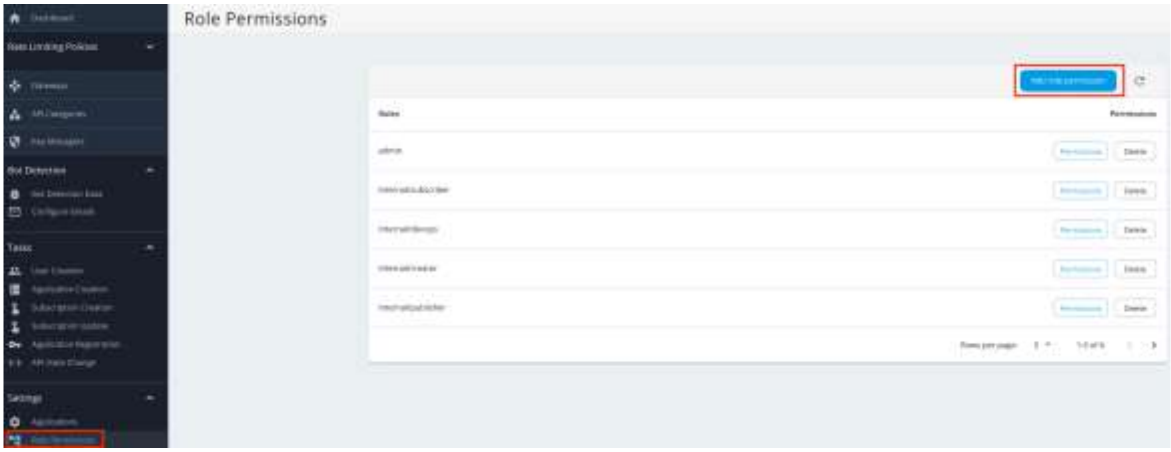
The new API Manager Publisher and Developer portals use Publisher and Developer portal REST APIs, with resources secured with scopes. Each action of the UI, login/ create api/ publish/ subscribe are governed by the scopes.
These scopes are mapped with the API Manager Internal/* roles. (Internal/creator, Internal/publisher etc)

Therefore, in order to provide access to the portals, the LDAP groups must be mapped with the respective Internal/* role in API Manager.

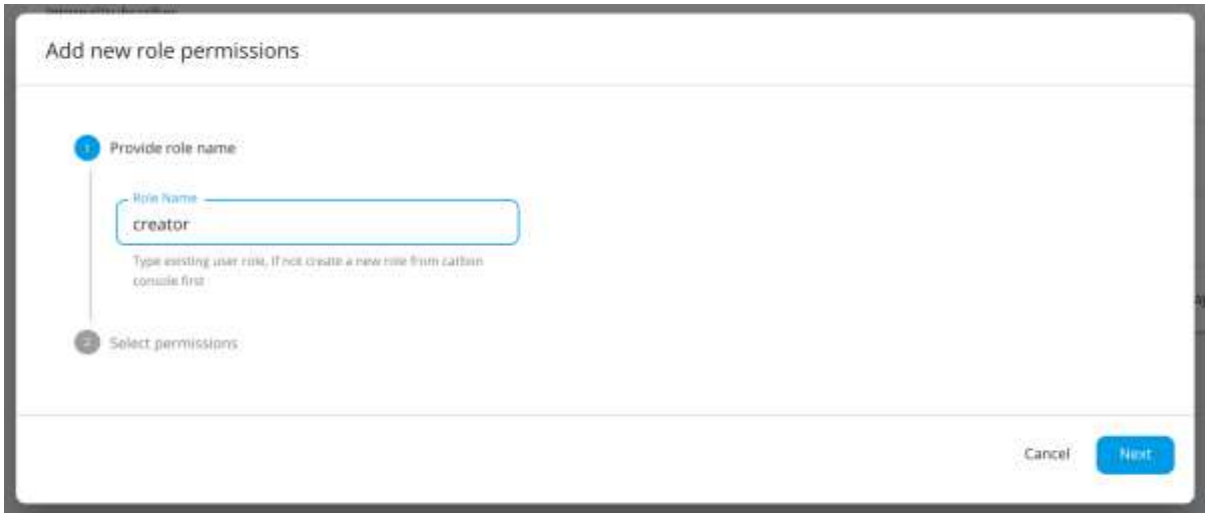
## Adding Role Mappings

You can use role mapping to map the above created roles to the existing default internal roles of API-M. This enables users with new roles to use REST API scopes of API-M Portals easily.

1. Sign in to the Admin Portal (`https://<APIM_Host>:<APIM_Port>/admin`) if you have not done so already.
2. Navigate to **Settings > Role Permissions** in Admin Portal and click on **Add Role Permission**.



3. Provide the name of the newly created role.



Add new role permissions

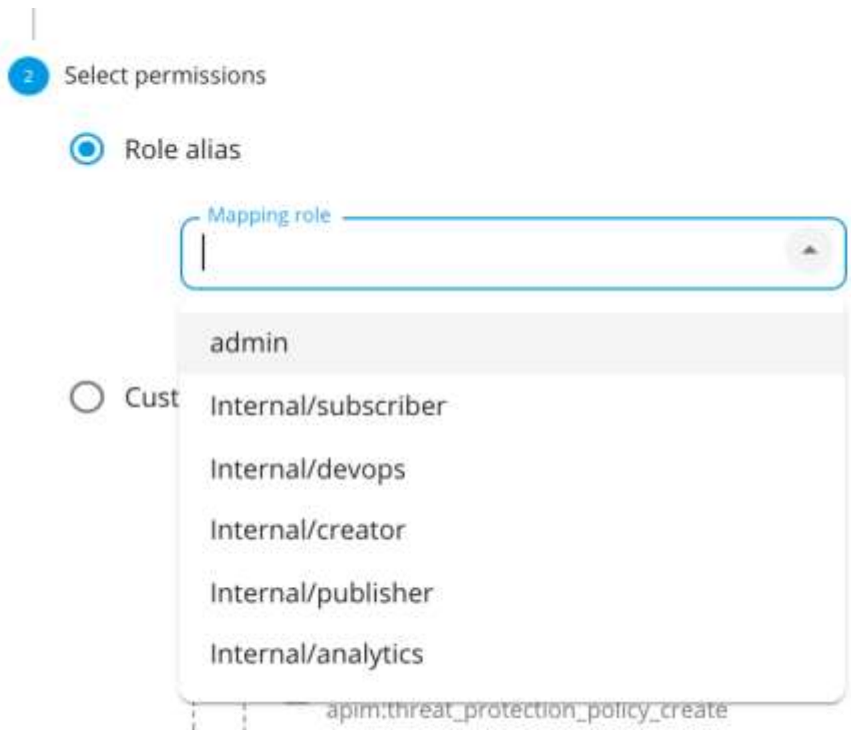
1 Provide role name

Role Name: creator

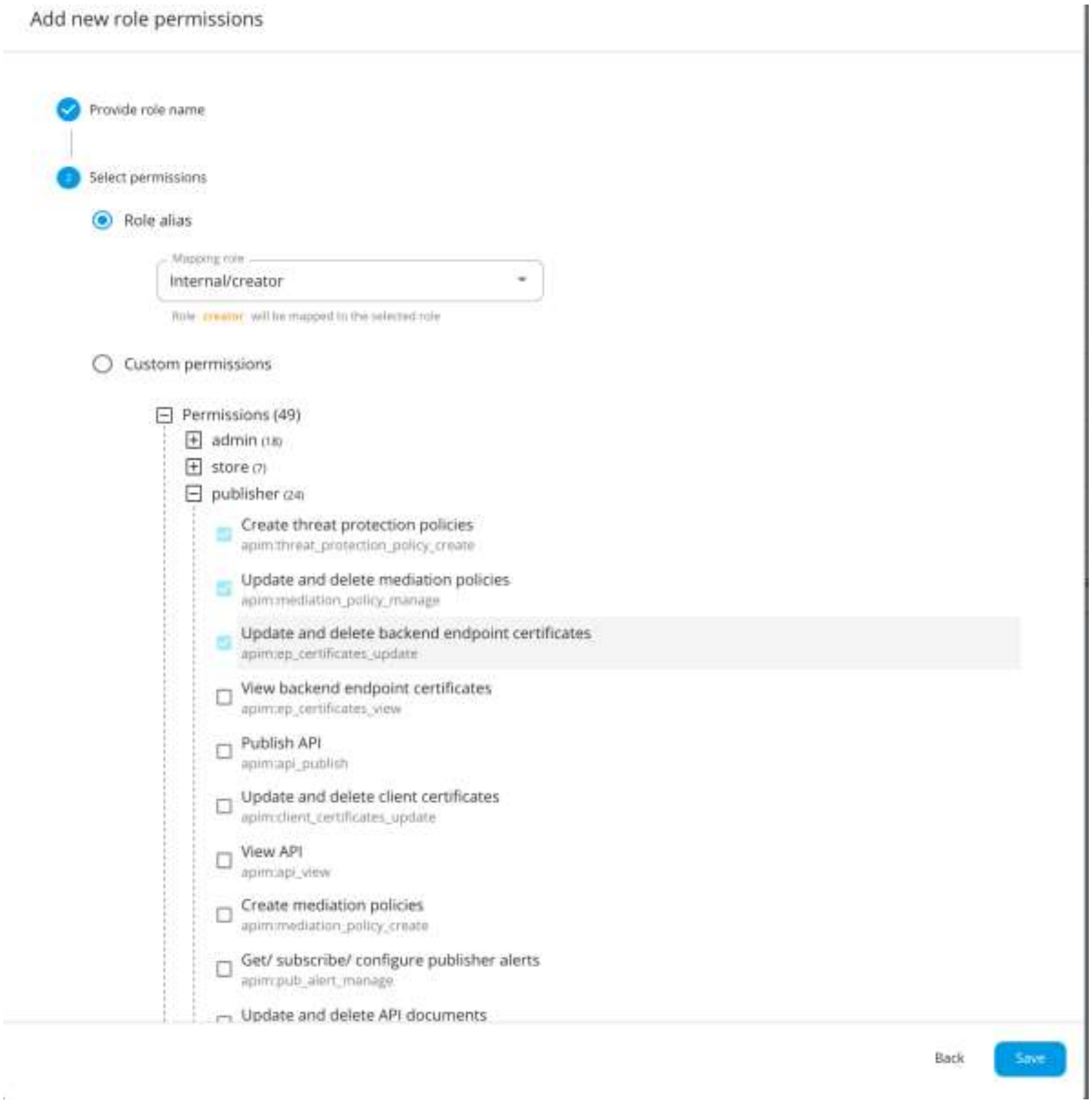
Type existing user role, If not create a new role from carbon console first

2 Select permissions

4. The newly created role can be mapped to an existing internal or admin role if required.



5. Select the required existing scopes for the newly created role and save the changes.



Add new role permissions

Provide role name

Select permissions

● Role alias

Mapping role: Internal/creator

Role creator will be mapped to the selected role

○ Custom permissions

- Permissions (49)
  - admin (18)
  - store (7)
  - publisher (24)
    - ☑ Create threat protection policies  
      apim:threat_protection_policy_create
    - ☑ Update and delete mediation policies  
      apim:mediation_policy_manage
    - ☑ Update and delete backend endpoint certificates  
      apim:ep_certificates_update
    - □ View backend endpoint certificates  
      apim:ep_certificates_view
    - □ Publish API  
      apim:api_publish
    - □ Update and delete client certificates  
      apim:client_certificates_update
    - □ View API  
      apim:api_view
    - □ Create mediation policies  
      apim:mediation_policy_create
    - □ Get/ subscribe/ configure publisher alerts  
      apim:pub_alert_manage
    - □ Update and delete API documents

This will update all the scope mappings in the `tenant-conf.json` file with the `Internal/creator` role as an allowed role. As a result, the new creator role will also be allowed for all scopes that are allowed for the `Internal/creator` role.

### Info

The following are the scopes that are allowed for each default Internal role under the default configurations section.

| Scope | admin | Internal/publisher | Internal/creator | Internal/subscriber | Internal/analytics | Internal/everyone |
|---|---|---|---|---|---|---|
| apim:api_publish | ✓ | ✓ | | | | |

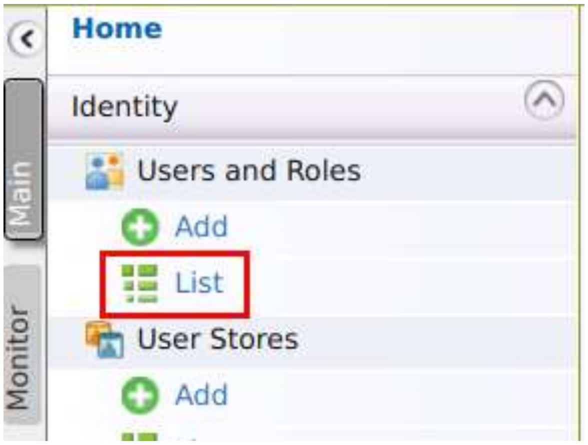
| Scope | admin | Internal/publisher | Internal/creator | Internal/subscriber | Internal/analytics | Internal/everyone |
|---|---|---|---|---|---|---|
| apim:api_create | ✓ |  | ✓ |  |  |  |
| apim:api_view | ✓ | ✓ | ✓ |  | ✓ |  |
| apim:api_delete | ✓ |  | ✓ |  |  |  |
| apim:subscribe | ✓ |  |  | ✓ |  |  |
| apim:tier_view | ✓ | ✓ | ✓ |  |  |  |
| apim:tier_manage | ✓ |  |  |  |  |  |
| apim:bl_view | ✓ |  |  |  |  |  |
| apim:subscription_view | ✓ | ✓ | ✓ |  |  |  |
| apim:subscription_block | ✓ | ✓ |  |  |  |  |
| apim:mediation_policy_view | ✓ |  | ✓ |  |  |  |
| apim:mediation_policy_create | ✓ |  | ✓ |  |  |  |
| apim:api_workflow | ✓ |  |  |  |  |  |
| apim:app_owner_change | ✓ |  |  |  |  |  |
| apim:app_import_export | ✓ |  |  |  |  |  |
| apim:api_import_export | ✓ |  |  |  |  |  |
| apim:label_manage | ✓ |  |  |  |  |  |
| apim:label_read | ✓ |  |  |  |  |  |
| apim:app_update | ✓ |  |  | ✓ |  |  |
| apim:app_manage | ✓ |  |  | ✓ |  |  |
| apim:sub_manage | ✓ |  |  | ✓ |  |  |
| apim:monetization_usage_publish | ✓ | ✓ |  |  |  |  |
| apim:document_create | ✓ | ✓ | ✓ |  |  |  |
| apim:ep_certificates_update | ✓ |  | ✓ |  |  |  |
| apim:client_certificates_update | ✓ |  | ✓ |  |  |  |
| apim:threat_protection_policy_manage | ✓ |  | ✓ |  |  |  |
| apim:document_manage | ✓ | ✓ | ✓ |  |  |  |

| Scope | admin | Internal/publisher | Internal/creator | Internal/subscriber | Internal/analytics | Internal/everyone |
|---|---|---|---|---|---|---|
| apim:client_certificates_add | √ | | √ | | | |
| apim:publisher_settings | √ | √ | √ | | | |
| apim:store_settings | √ | | | √ | | |
| apim:client_certificates_view | √ | | √ | | | |
| apim:mediation_policy_manage | √ | | √ | | | |
| apim:threat_protection_policy_create | √ | | √ | | | |
| apim:ep_certificates_add | √ | | √ | | | |
| apim:ep_certificates_view | √ | | √ | | | |
| apim:api_key | √ | | √ | | | |
| apim_analytics:admin | √ | | | | | |
| apim_analytics:api_analytics:own | √ | | | | | |
| apim_analytics:api_analytics:edit | √ | | | | | |
| apim_analytics:api_analytics:view | √ | √ | √ | | | |
| apim_analytics:application_analytics:own | √ | | | | | |
| apim_analytics:application_analytics:edit | √ | | | | | |
| apim_analytics:application_analytics:view | √ | | | √ | | |
| apim_analytics:monitoring_dashboard:own | √ | | | | | |
| apim_analytics:monitoring_dashboard:edit | √ | | | | | |
| apim_analytics:monitoring_dashboard:view | √ | | | | √ | |
| apim_analytics:business_analytics:own | √ | | | | | |
| apim_analytics:business_analytics:edit | √ | | | | | |

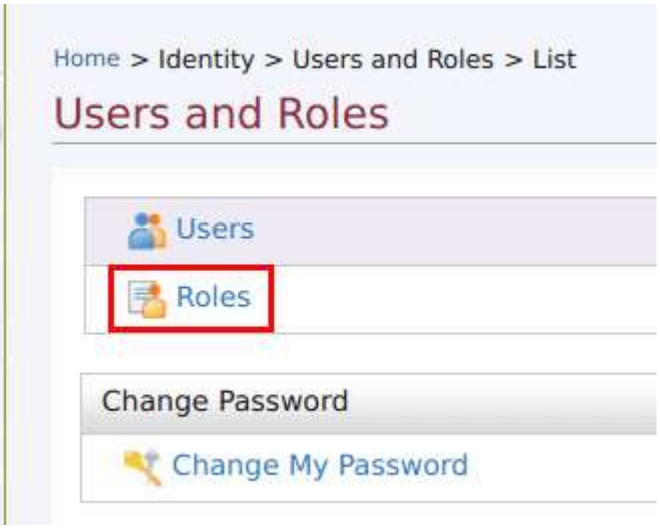
| Scope | admin | Internal/publisher | Internal/creator | Internal/subscriber | Internal/analytics | Internal/everyone |
|---|---|---|---|---|---|---|
| apim_analytics:business_analytics:view | ✓ | | | | ✓ | |
| apim:pub_alert_manage | ✓ | | ✓ | | | |
| apim:sub_alert_manage | ✓ | | | ✓ | | |
| apim:tenantInfo | ✓ | | | | | |
| apim:admin_operations | ✓ | | | | | |
| apim:shared_scope_manage | ✓ | | | | | |

## Editing or deleting a role

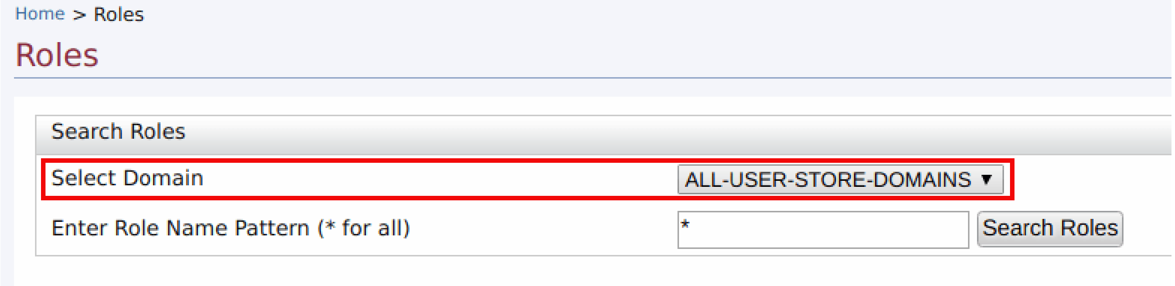
1. Sign in to the management console (`https://<APIM_Host>:<APIM_Port>/carbon`) if you have not done so already.
2. In the **Main** menu, click **List** under **Users and Roles**.



3. Click **Roles**.



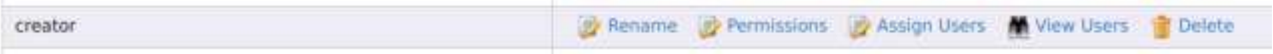
4. If you need to modify to a role, select the domain (user store) under **Search Roles** > **Select Domain** where the role resides.



Select Domain: ALL-USER-STORE-DOMAINS ▼

Enter Role Name Pattern (* for all): *

Then use the relevant links in the **Actions** column that corresponds to the role listing to perform the following:



- Rename the role
- Change the default permissions associated with this role
- Assign this role to users
- View the users who are assigned this role
- Delete the role if you no longer need it

**Info**

If the role is in an external user store to which you are connected in read-only mode, you will be able to view the existing roles but not edit or delete them. However, you can still create new editable roles.

## Updating before the first startup (recommended)

The default role name of the Administrator, (`admin`) can be changed before starting WSO2 API Manager by editing `<API-M_HOME>/repository/conf/deployment.toml` file. For more information, see Change the super admin credentials.
Configure the property `admin_role` with your custom role (`administrator`) in the `deployment.toml` file as follows and start the server.

```
[super_admin]
admin_role = "administrator"
username = "admin"
password = "admin"
create_admin_account = true
```

## Updating the role name after the product is used for some time (advanced configuration)

**Tip**

These steps are not necessary if you have already updated the role names before the first startup of the product.

The following steps guide you through updating the role names after you have used the product for some time.

1. Make the configuration changes indicated in the above section.
2. Do the following user store level changes for existing users:

   If you are connected to the `JDBCUserStoreManager`, update the `UM_ROLE` table with the new role name that you defined in place of the `admin` role.

   **Info**

   The schema can be located by referring to the data source `[database.shared_db]` defined in the `deployment.toml` file. The data source definition can also be found in the same file.

   - If you are connected to the `ReadWriteLdapUserStoreManager`, populate the members of the previous `admin` role to the new role under **Groups**.
3. Restart the server.

| **Original Role** | **Mapped Role** |
|---|---|
| Internal/publisher | Architect |

| | |
|---|---|
| Internal/creator | Developer |
| Internal/subscriber | Developer |
| admin | Admin |

## Expected Outcome

Understand how to define roles in your organization based on user store groups already created or by verifying the need to create new groups to manage such permissions and providing access to the roles by mapping with Internal/* roles.