portainer.io

Portainer Extension Software

Implementation Guide

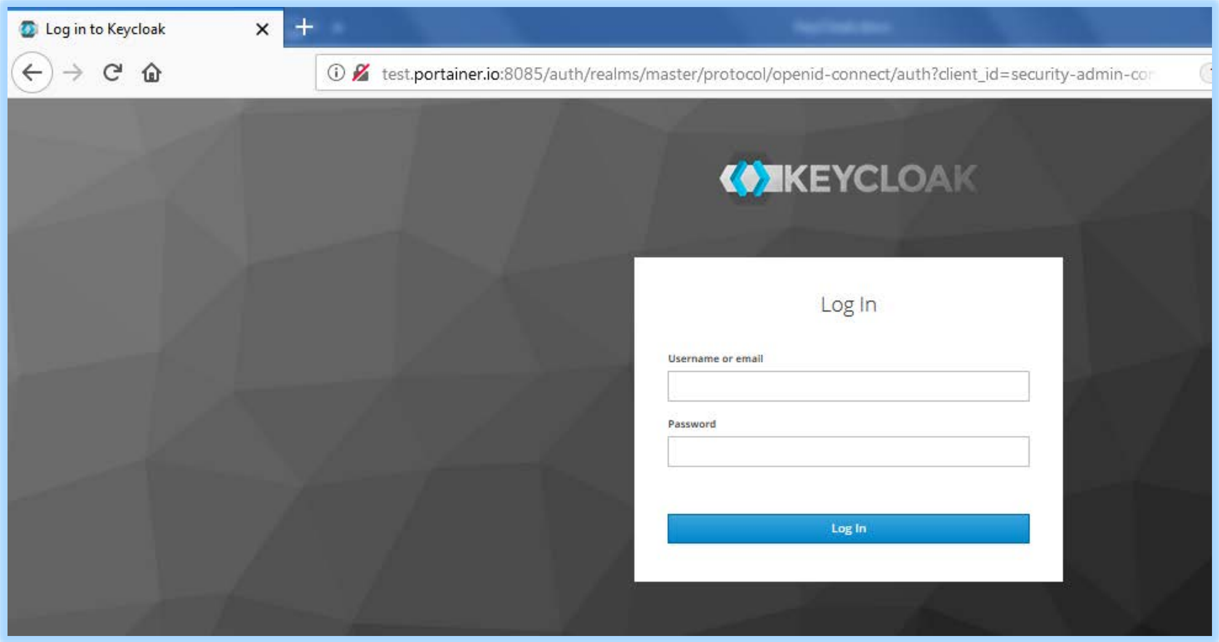
External Authentication for KeyCloak

March 2019

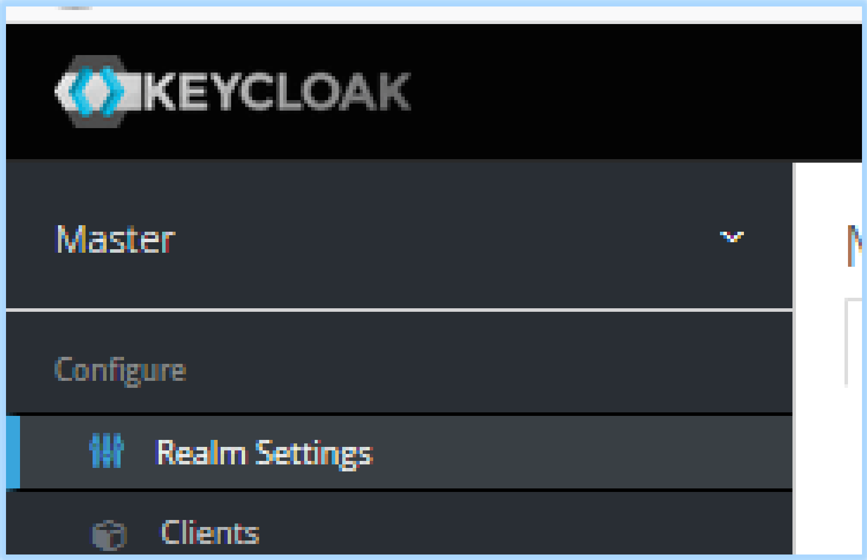
Implementation Guide

External Authentication for KeyCloak

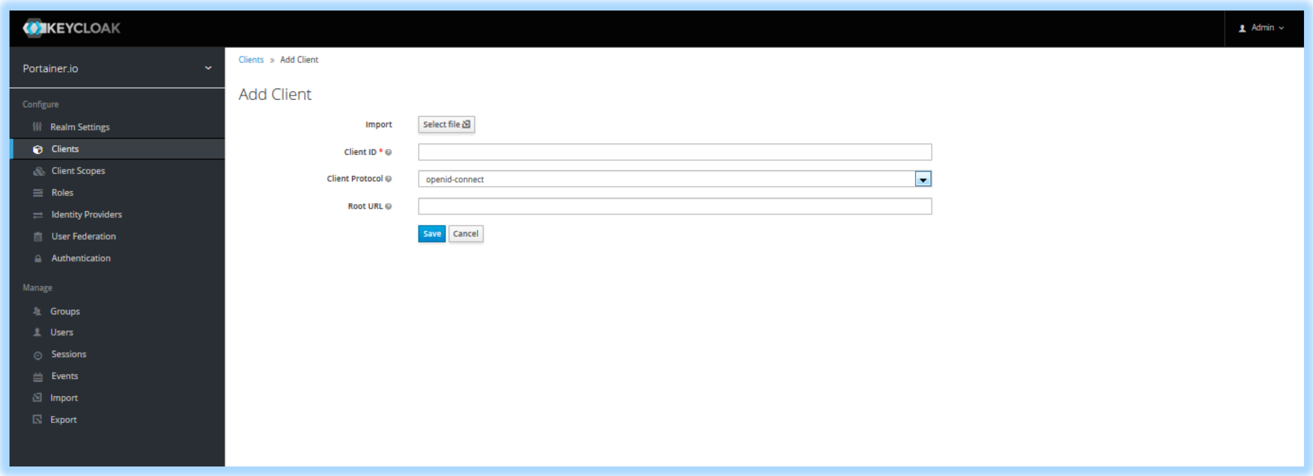
Step 1. Login to KeyCloak Administration Console as an Admin


Log in to Keycloak
test.portainer.io:8085/auth/realms/master/protocol/openid-connect/auth?client_id=security-admin-co
KEYCLOAK
Log In
Username or email
Password
Log In

Step 2. Select the applicable authentication “Realm” from the dropdown in the left sidebar



Step 3. Click on “Clients”, in the left sidebar, and then click the “Create” button to define a new app instance.

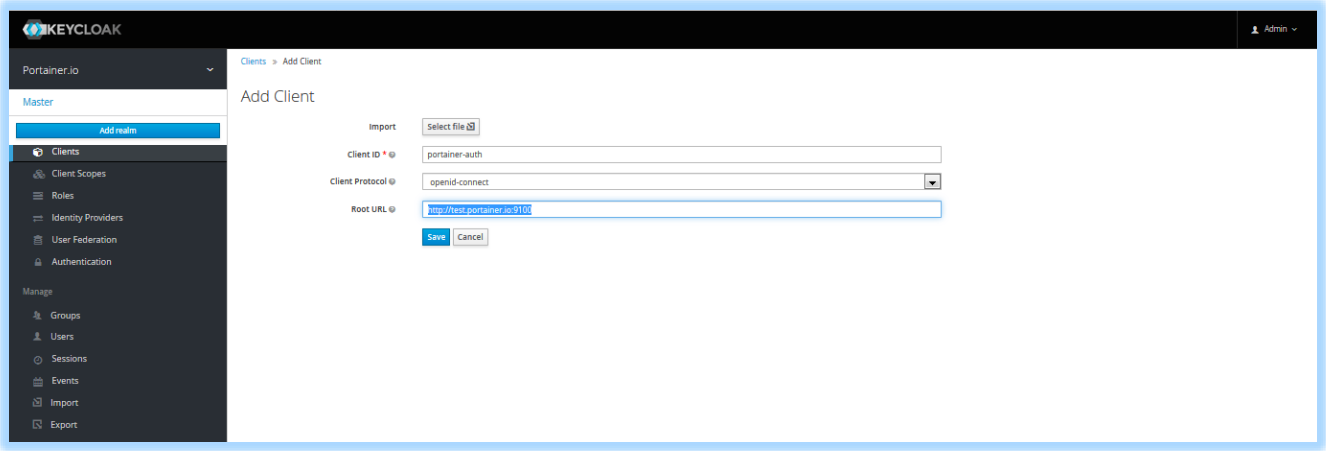


Client ID *

Client Protocol: openid-connect

Root URL

In the “Client ID” enter in (and record) a name for the Portainer App instance you are authorising. Something like portainer-auth. Keep the client protocol as openid-connect, and for the root URL enter in the FDQL of your Portainer instance, as below, and then click “Save”.

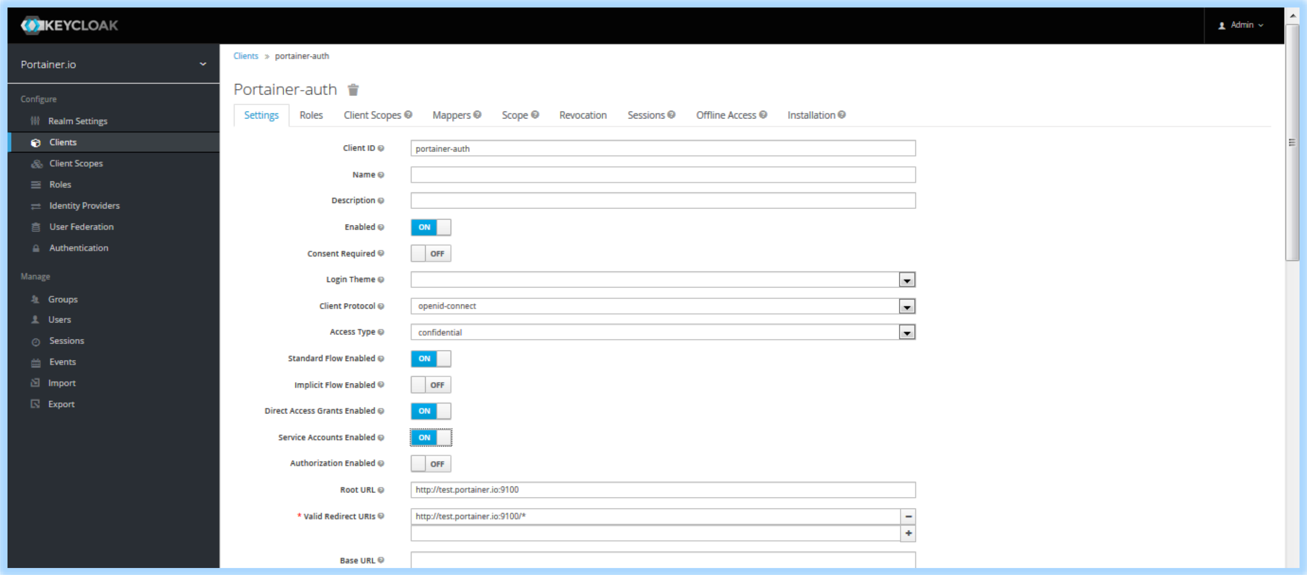


Client ID *: portainer-auth

Client Protocol: openid-connect

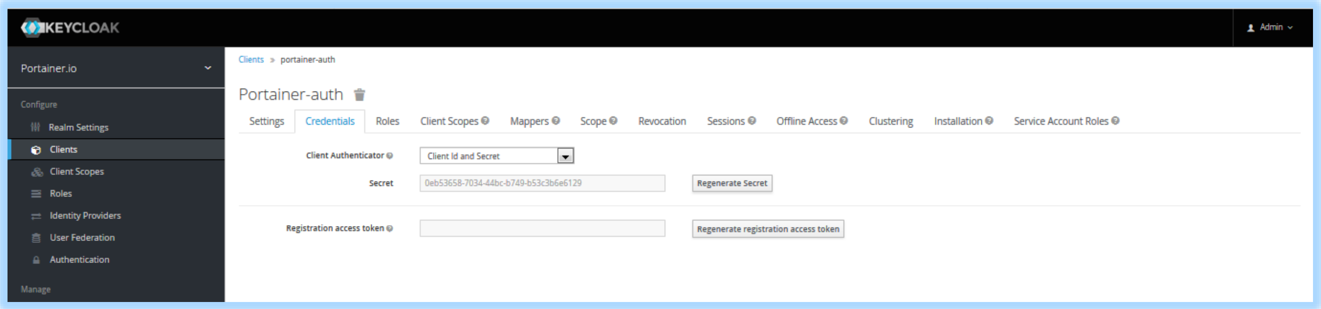
Root URL: http://test.portainer.io:9100

Now change the “Access Type” to “confidential”, and switch “Service Accounts Enabled” to “ON”, then click “Save”. Note that once you click “Save” a new header menu items appears, called “Credentials”. Click on that menu.



Client ID: portainer-auth
Name:
Description:
Enabled: ON
Consent Required: OFF
Login Theme:
Client Protocol: openid-connect
Access Type: confidential
Standard Flow Enabled: ON
Implicit Flow Enabled: OFF
Direct Access Grants Enabled: ON
Service Accounts Enabled: ON
Authorization Enabled: OFF
Root URL: http://test.portainer.io:9100
* Valid Redirect URIs: http://test.portainer.io:9100/*
Base URL:

Take a note of the secret; this will be required later.



Client Authenticator: Client Id and Secret
Secret: 0eb53658-7034-44bc-b749-b53c3b6e6129
Registration access token:

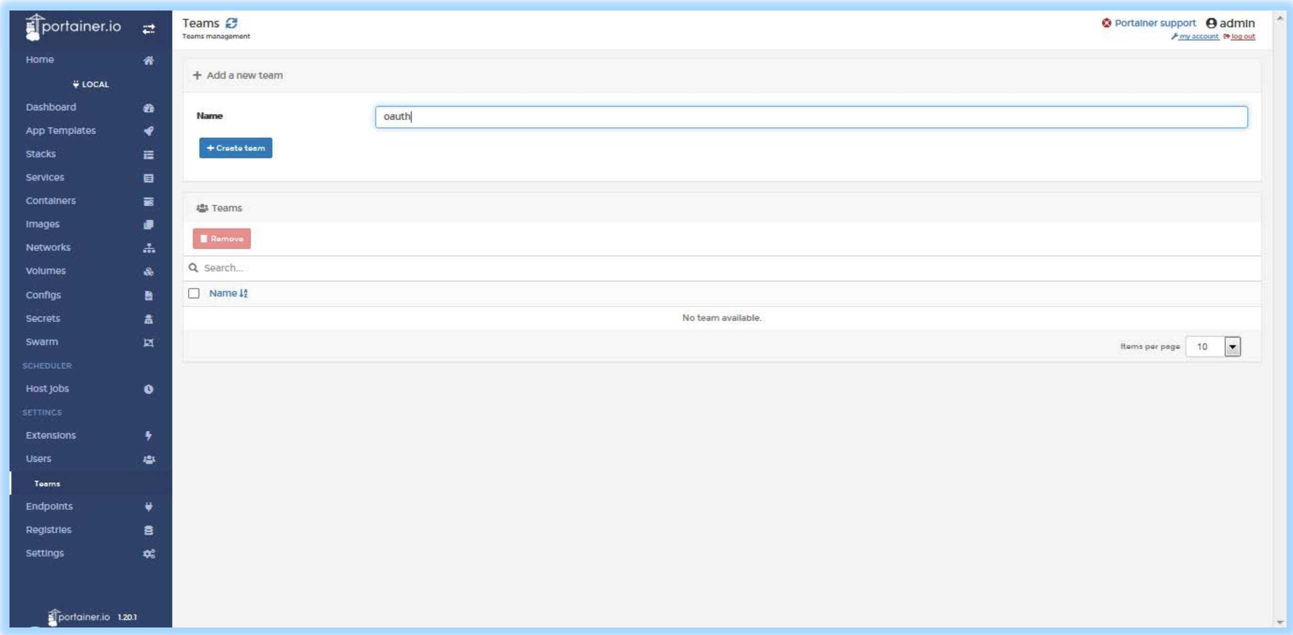
Now, we assume you already have users defined in your KeyCloak system, but if not, click on "Users" in the left sidebar and add users as required.

Step 4: Switch to your Portainer Instance and login as the local instance admin

Purchase the Portainer External Authentication extension and apply the license key (process not shown here).

Let's setup some basics, so that when user's login for the first time, they can actually access Portainer resources.

Click on "Users" and then "Teams", and create a team called "oauth" (or one of your choosing)



Teams
Teams management

Add a new team

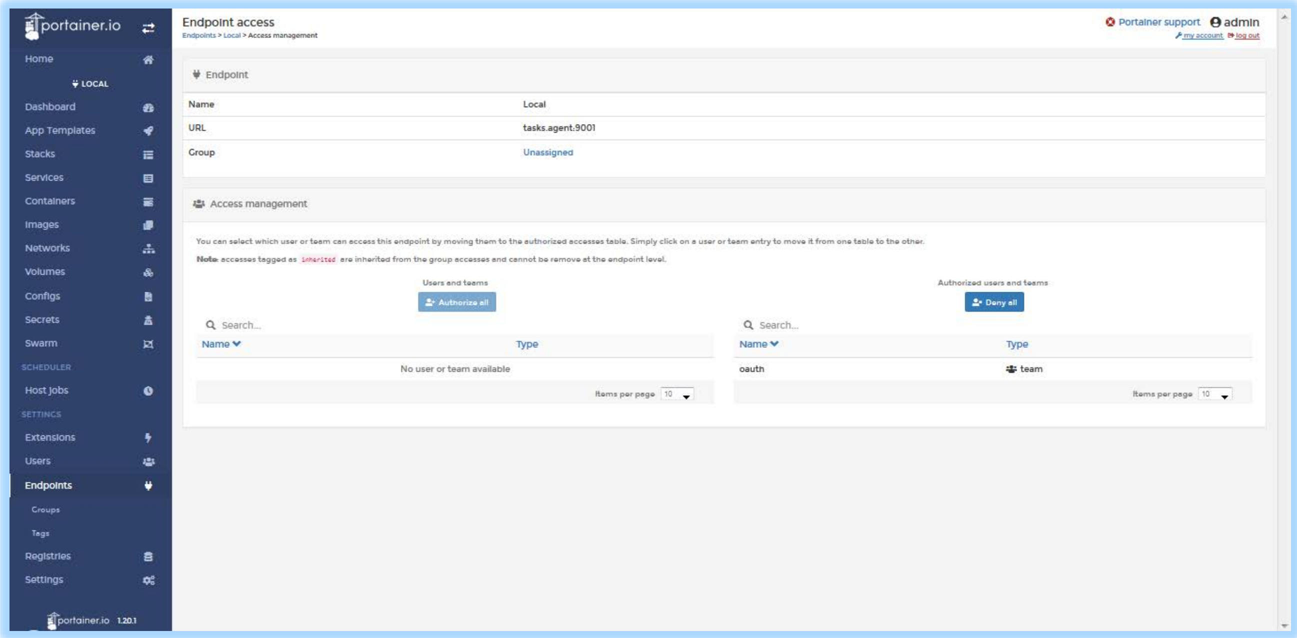
Name oauth

Teams

☐ Name
No team available.

Items per page 10

Click on “Endpoints” and then select the endpoints you would like to grant the oAUTH users access to manage, and then click “Manage Access”. Assign the oAUTH group you created to the authorized list.



Endpoint access
Endpoints > Local > Access management

Endpoint

Name	Local
URL	tasks.agent:9001
Group	Unassigned

Access management

You can select which user or team can access this endpoint by moving them to the authorized accesses table. Simply click on a user or team entry to move it from one table to the other.

Note: accesses tagged as `inherited` are inherited from the group accesses and cannot be remove at the endpoint level.

Users and teams

Name	Type
No user or team available	

Items per page 10

Authorized users and teams

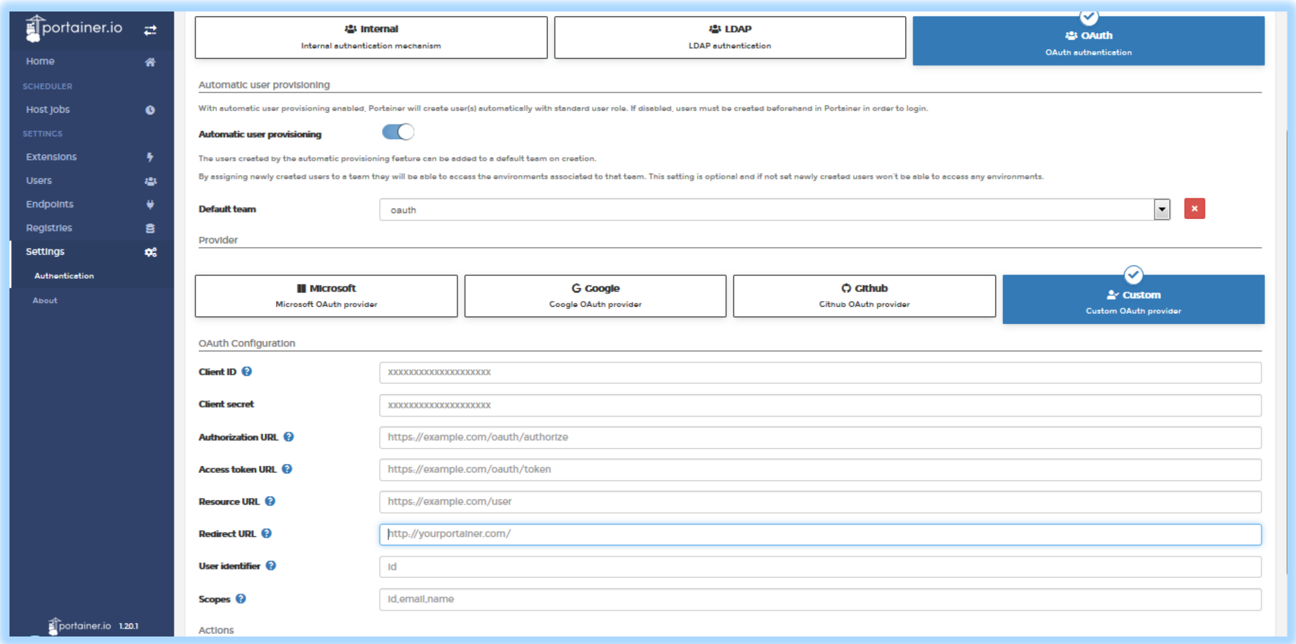
Name	Type
oauth	team

Items per page 10

Step 5. Let’s configure Portainer External Authenication.

Click on “settings” and then “Authentication”

Select “oAUTH” and then select “Custom”



Automatic user provisioning

With automatic user provisioning enabled, Portainer will create user(s) automatically with standard user role. If disabled, users must be created beforehand in Portainer in order to login.

Automatic user provisioning ●

The users created by the automatic provisioning feature can be added to a default team on creation.

By assigning newly created users to a team they will be able to access the environments associated to that team. This setting is optional and if not set newly created users won't be able to access any environments.

Default team oauth

Provider

OAuth Configuration

Client ID xxxxxxxxxxxxxxxxxxxx

Client secret xxxxxxxxxxxxxxxxxxxx

Authorization URL https://example.com/oauth/authorize

Access token URL https://example.com/oauth/token

Resource URL https://example.com/user

Redirect URL http://yourportainer.com/

User identifier id

Scopes id,email,name

Actions

Enable “Automatic User Provisioning”, and select the default team(oAUTH or similar) that you created previoulsly.

In “Client ID” enter in the name you defined the Portainer App as in KeyCloak (eg portainer-auth).

In the “Client Secret” enter in the secret that was shown on the “Credentials” page when you defined the Portainer App in KeyCloak.

In the “Authorisation URL” field, enter http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/auth but make sure to change to your server name and port, and change portainer.io to the name of your actual realm.

In the “Access token URL” field, enter http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/token but make sure to change to your server name and port, and change portainer.io to the name of your actual realm.

In the “Resource URL” field, enter

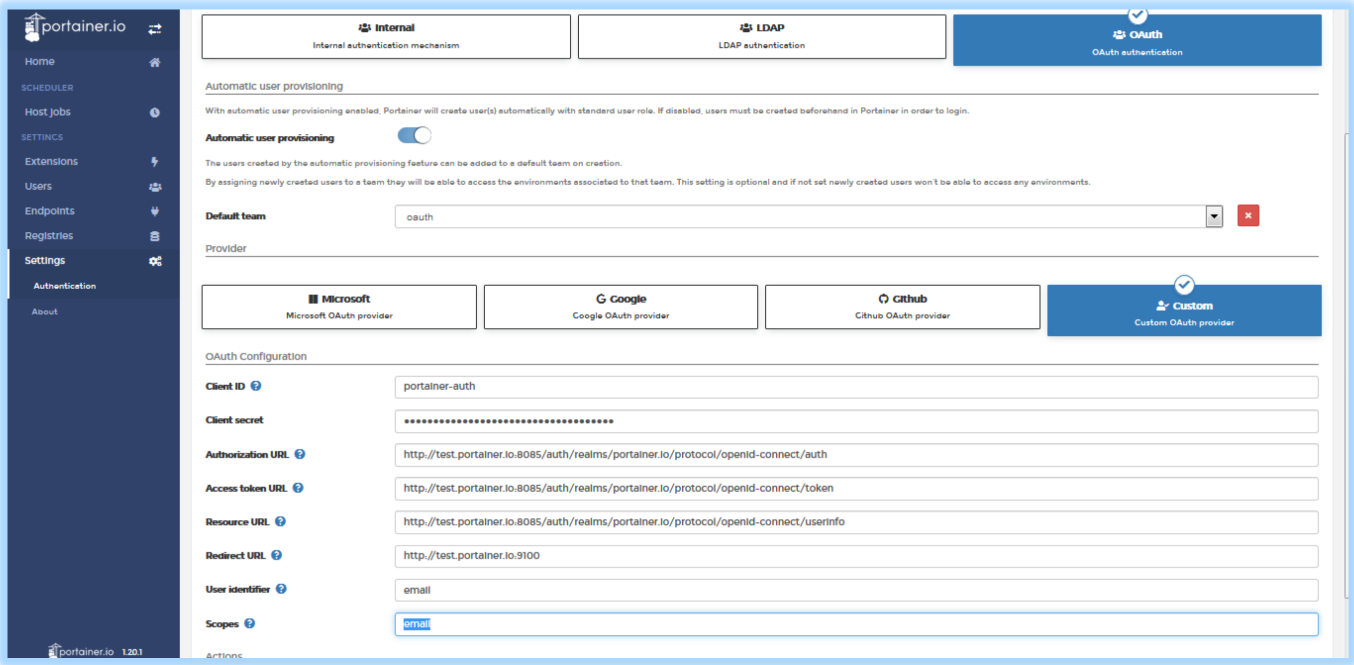
http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/userinfo but make sure to change to your server name and port, and change portainer.io to the name of your actual realm.

In the “Redirect URL” field, enter the FDQN/URL of your Portainer instance, eg http://test.portainer.io:9100

In the “User Identifier” field, enter “email”

In the “Scopes” field, enter “email”

Click Save



Automatic user provisioning

With automatic user provisioning enabled, Portainer will create user(s) automatically with standard user role. If disabled, users must be created beforehand in Portainer in order to login.

Automatic user provisioning

The users created by the automatic provisioning feature can be added to a default team on creation.

By assigning newly created users to a team they will be able to access the environments associated to that team. This setting is optional and if not set newly created users won't be able to access any environments.

Default team: oauth

Provider

OAuth Configuration

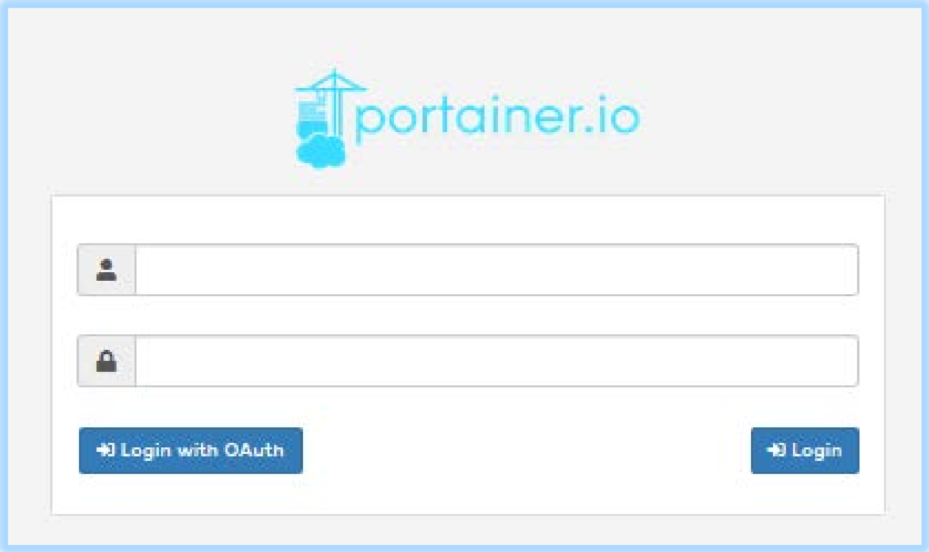
Client ID: portainer-auth
Client secret: ••••••••••••••••••••••••••••••••••••
Authorization URL: http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/auth
Access token URL: http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/token
Resource URL: http://test.portainer.io:8085/auth/realms/portainer.io/protocol/openid-connect/userinfo
Redirect URL: http://test.portainer.io:9100
User identifier: email
Scopes: email

Click Save.

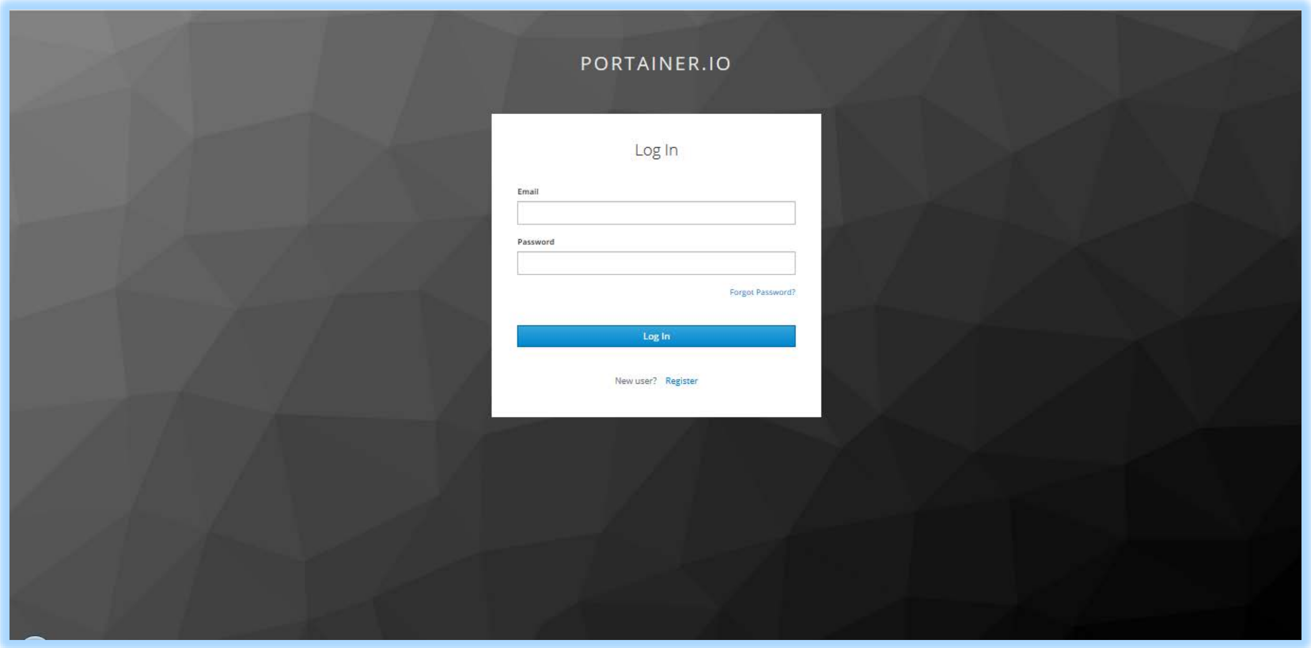
Logout as the Admin

Step 6. Login using oAUTH

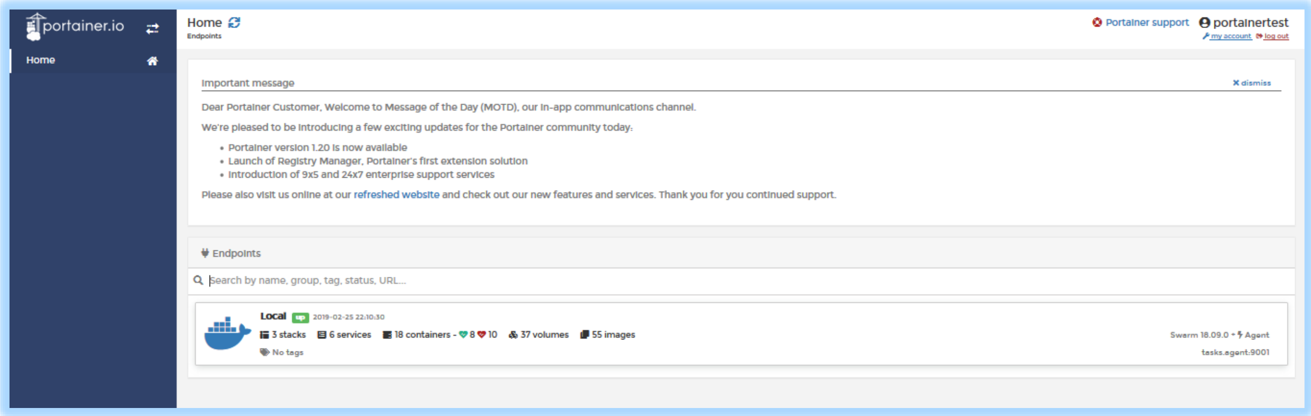
At the Login Page, click the "Login with oAUTH" box for oAUTH login.



Enter your KeyCloak username and password where prompted (note you are redirected to your KeyCloak instance for Auth).



You are now logged into Portainer using oAUTH from KeyCloak.



Important message

Dear Portainer Customer, Welcome to Message of the Day (MOTD), our in-app communications channel.

We're pleased to be introducing a few exciting updates for the Portainer community today:

- Portainer version 1.20 is now available
- Launch of Registry Manager, Portainer's first extension solution
- Introduction of 9x5 and 24x7 enterprise support services

Please also visit us online at our refreshed website and check out our new features and services. Thank you for you continued support.

Local 2019-02-25 22:10:30
3 stacks 6 services 18 containers - 8 10 37 volumes 55 images
No tags
Swarm 18.09.0 + Agent
tasks.agent:9001

Optionally, if you want your oAUTH user to be a Portainer Admin, first login/logout as the oAUTH user to create the Portainer record, then login as the Portainer local admin (or as another admin), and then edit the user to elevate them to an admin.

.end.