Introduction à RH-SSO

Centraliser la gestion d'identité et sécuriser vos applications

Michael Lessard
Architecte Principal
mlessard@redhat.com
michaellessard

Red Hat

AGENDA

- Introduction à Red Hat SSO
    - Architecture
    - Fonctionnalités
    - SAML vs OIDC
    - Intégration avec Openshift
- Démonstrations
    - installation de RH-SSO avec Openshift
    - console administrateur, console client et connection d'une application en Java
    - fédération avec sources d'authentifications externes - Gitlab, Twitter, etc.
    - Évolutivité avec Openshift

AMÉLIORE LA SÉCURITÉ EN LAISSANT LA GESTION DES IDENTIFIANTS AUX EXPERTS

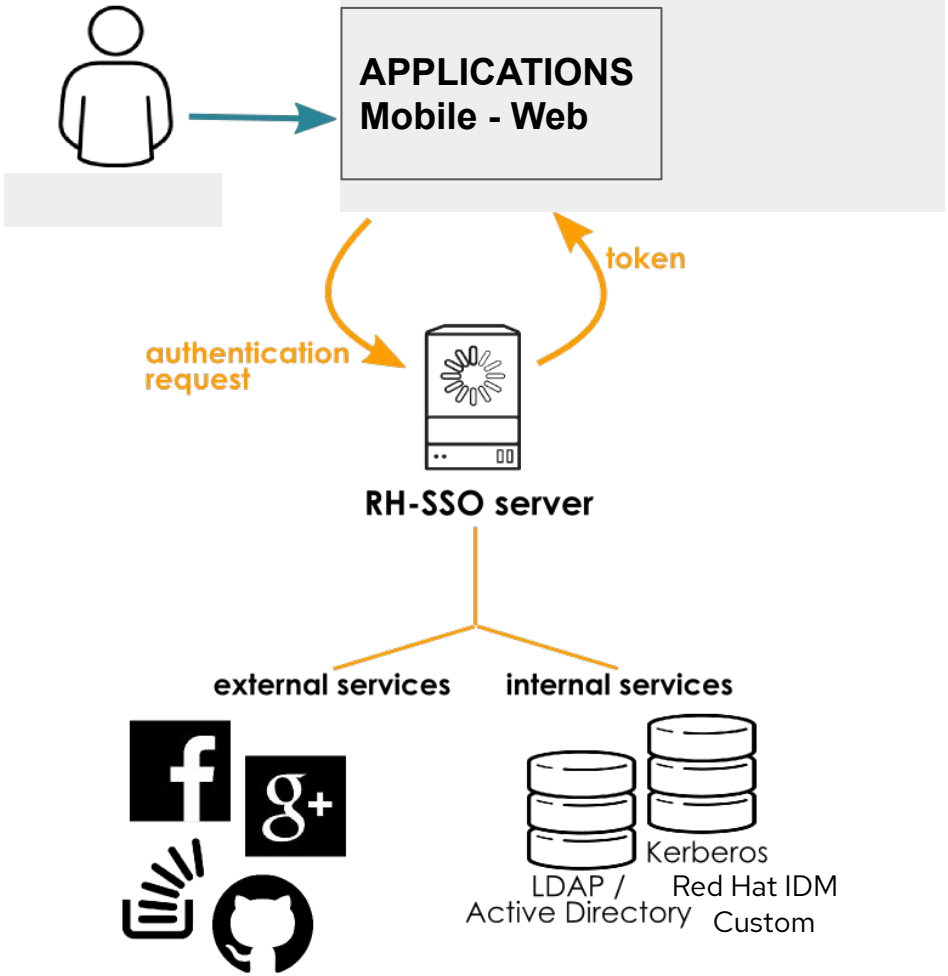
RED HAT®
SSO

Sécuriser les applications web et fournir des capacités de connexion unique basées sur des standards populaires tels que SAML 2.0, OpenID Connect et OAuth 2.0.

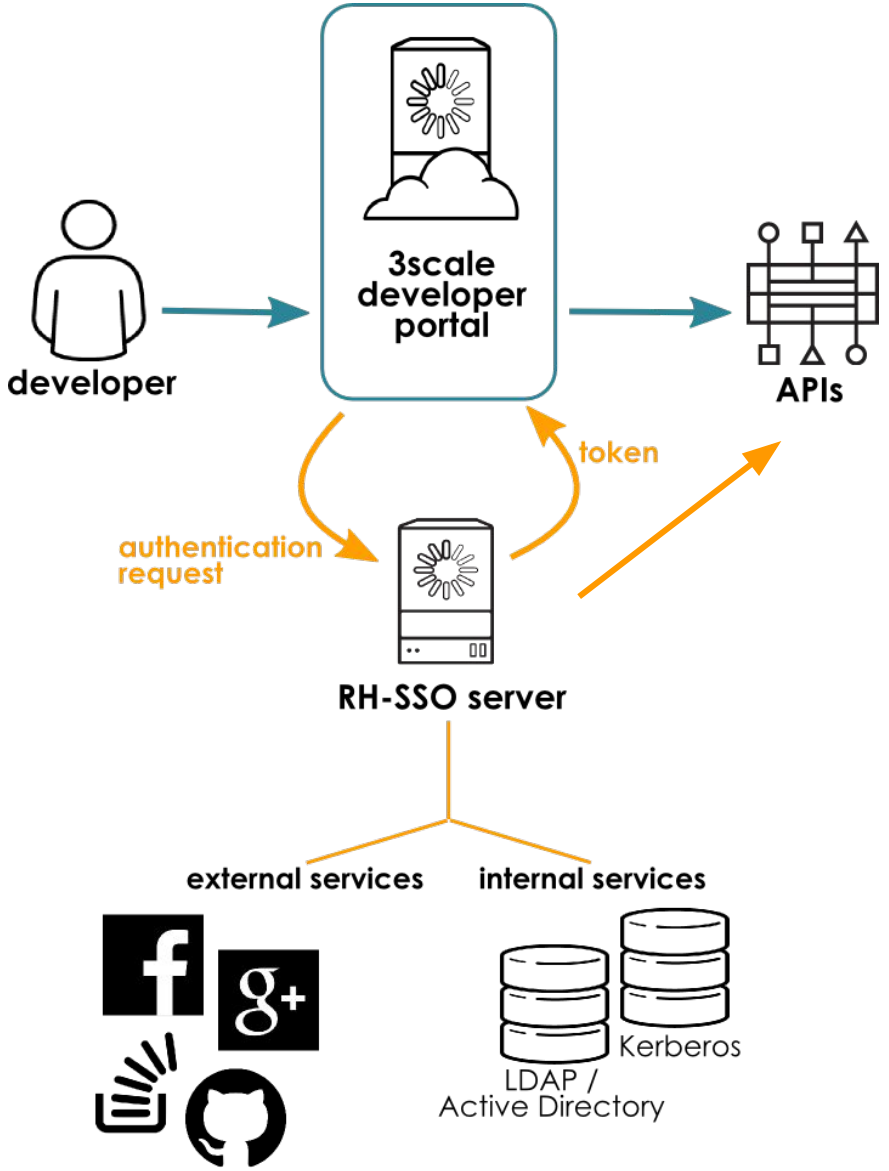
SIMPLE D'UTILISATION ET DÉPLOIEMENT RAPIDE

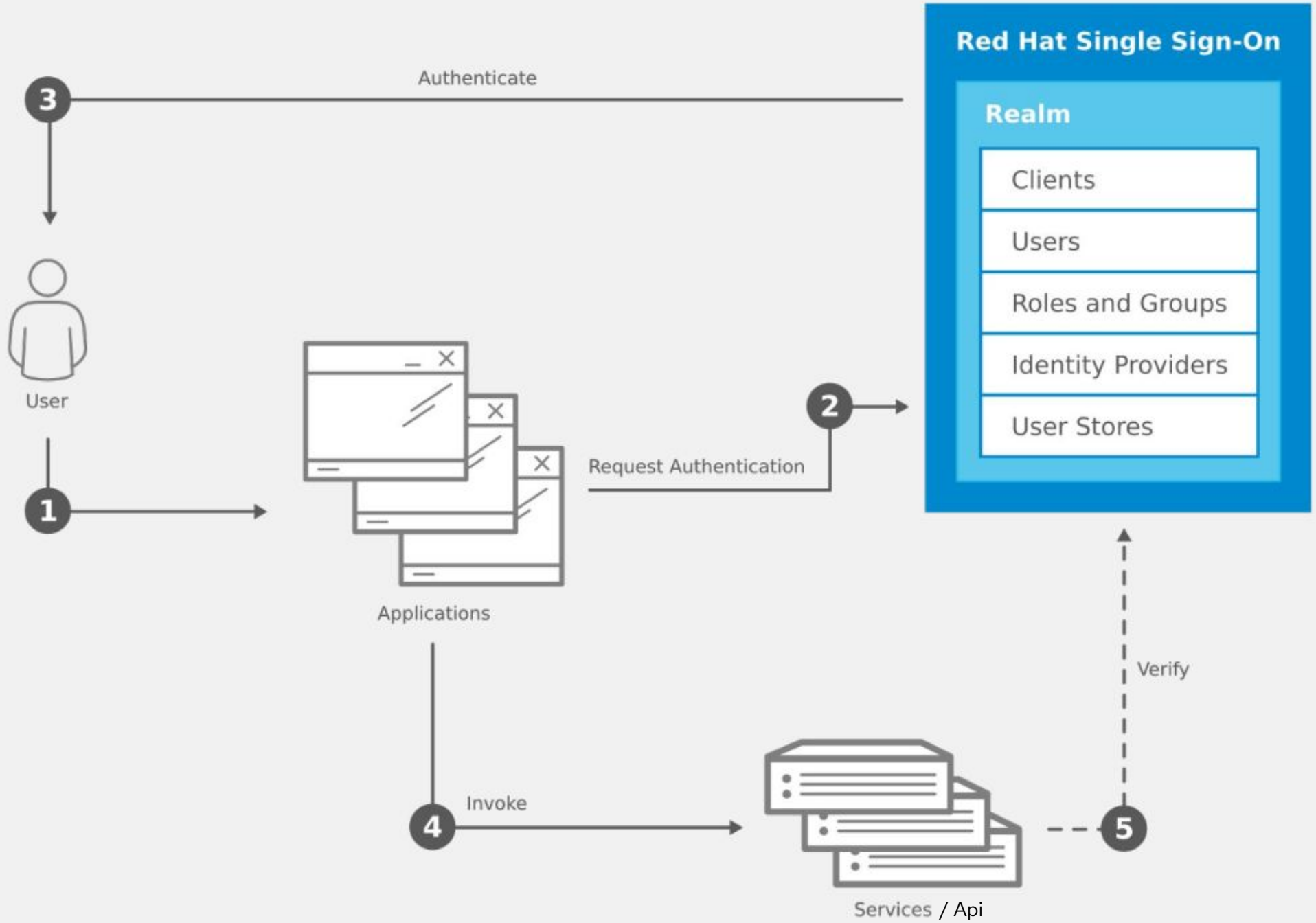
- Prend en charge les normes de sécurité modernes pour l'authentification et l'autorisation. (multi-facteur, captcha, etc ...)
- Fichier journaux des accès (audit).
- Intégré aux systèmes d'enregistrement existants (LDAP, AD, etc.) et réseaux sociaux.
- Intégration avec Data Grid pour la réplication inter-site, inter-cloud
- Application java disponible en conteneur. Inclus avec Openshift.
- Basé sur le projet communautaire Keycloak.

ARCHITECTURE RH-SSO



Red Hat

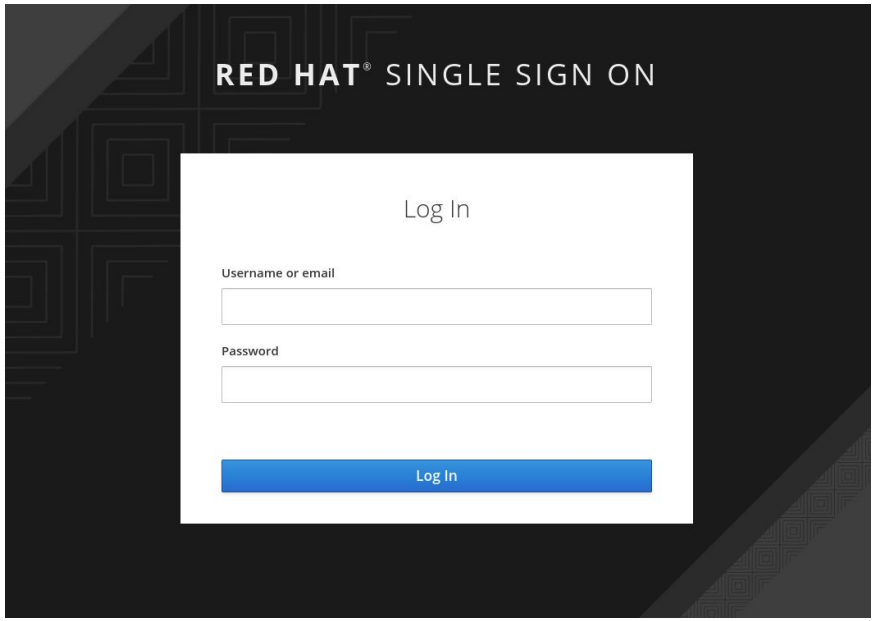

3scale
developer
portal
developer
APIs
token
authentication
request
RH-SSO server
external services
internal services
Kerberos
LDAP /
Active Directory


Red Hat Single Sign-On
Realm
Clients
Users
Roles and Groups
Identity Providers
User Stores
Authenticate
3
User
1
Applications
2
Request Authentication
Verify
4
Invoke
5
Services / Api

Authentification unique (Single-Sign On)

Les utilisateurs s'authentifient auprès de Red Hat SSO plutôt que par des applications individuelles. Cela signifie que vos applications n'ont pas à traiter de formulaires de connexion, d'authentification des utilisateurs et de stockage des utilisateurs. Une fois connectés au RH SSO, les utilisateurs n'ont pas besoin de se reconnecter pour accéder à une autre application.

Cela s'applique également à la déconnexion. Red Hat SSO fournit une déconnexion unique, ce qui signifie que les utilisateurs n'ont à se déconnecter qu'une seule fois pour être déconnectés de toutes les applications qui utilisent RH SSO.



Pont Kerberos

Si vos utilisateurs s'authentifient sur des postes de travail avec Kerberos (LDAP ou Active Directory), ils peuvent également être automatiquement authentifiés auprès du RH SSO sans avoir à fournir à nouveau leur nom d'utilisateur et leur mot de passe après s'être connectés au poste de travail.

SAML vs OIDC

Il existe deux normes populaires pour l'authentification fédérée. Le flux SAML (ou Security Assertion Markup Language) et OpenId Connect.

OpenId Connect est basé sur les processus d'OAuth 2.0 et utilise généralement le format JWT (JSON Web token) pour l'identifiant. Framework plus récent, plus moderne.

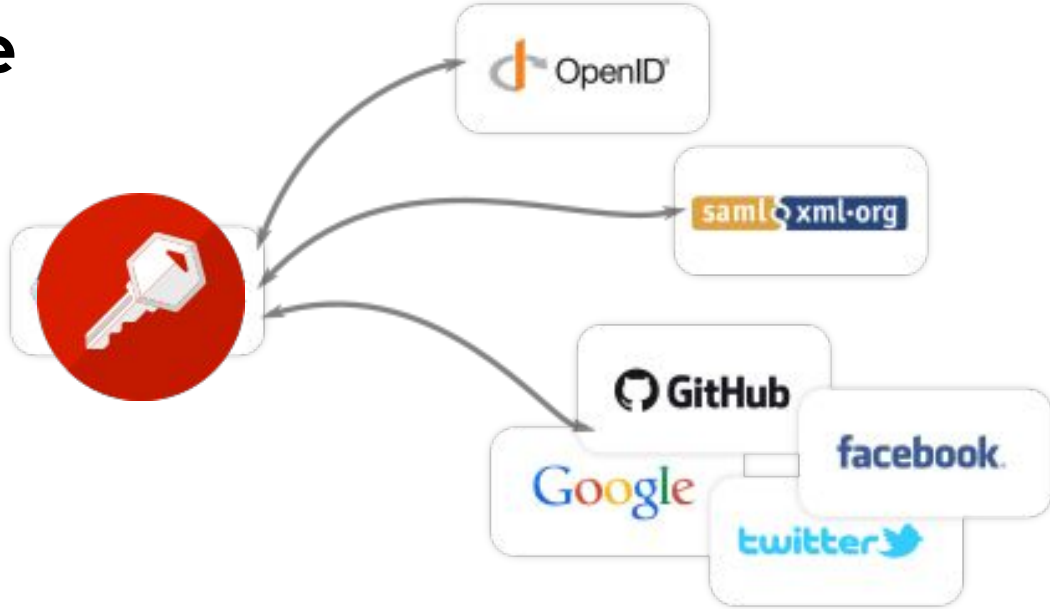
Le flux SAML est indépendant de OAuth 2.0, et repose sur l'échange de messages pour l'authentification au format XML SAML (au lieu du format JWT). Protocole mature.

Les deux flux permettent le SSO (Single Sign On), c'est-à-dire la possibilité de se connecter à un site web en utilisant ses identifiants de connexion à partir d'un site différent (par exemple, Facebook ou Google).

Courtage d'identité et connexion sociale

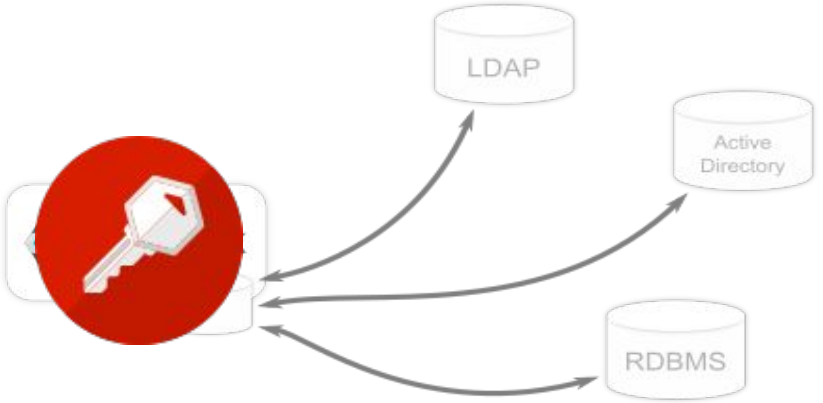
L'activation de la connexion aux réseaux sociaux est facile à ajouter via la console d'administration. Il suffit de sélectionner le réseau social que vous souhaitez ajouter. Aucun code ou modification de votre application n'est nécessaire.

Red Hat SSO peut également authentifier les utilisateurs avec les fournisseurs d'identité OpenID Connect ou SAML 2.0 existants. Encore une fois, il suffit de configurer le fournisseur d'identité via la console d'administration.

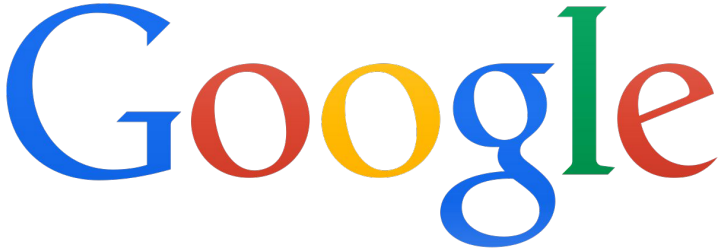


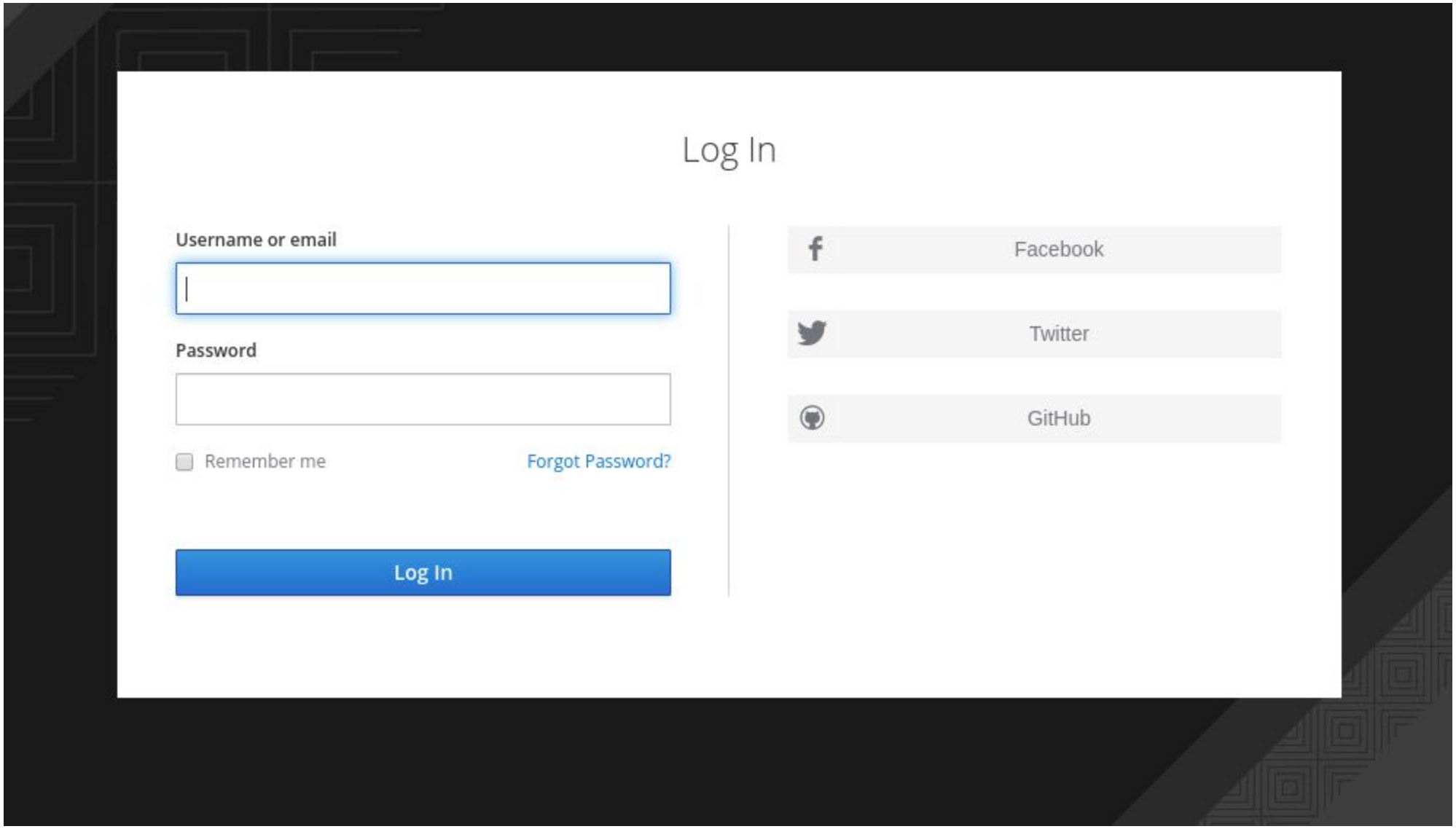
Fédération des utilisateurs

Red Hat SSO a un support intégré pour se connecter aux serveurs LDAP ou Active Directory existants. Vous pouvez également mettre en place votre propre fournisseur si vous avez des utilisateurs dans d'autres sources, par exemple une base de données relationnelle.



Fournisseurs d'identité externes




Log In
Username or email
Password
Remember me
Forgot Password?
Log In
Facebook
Twitter
GitHub

Red Hat

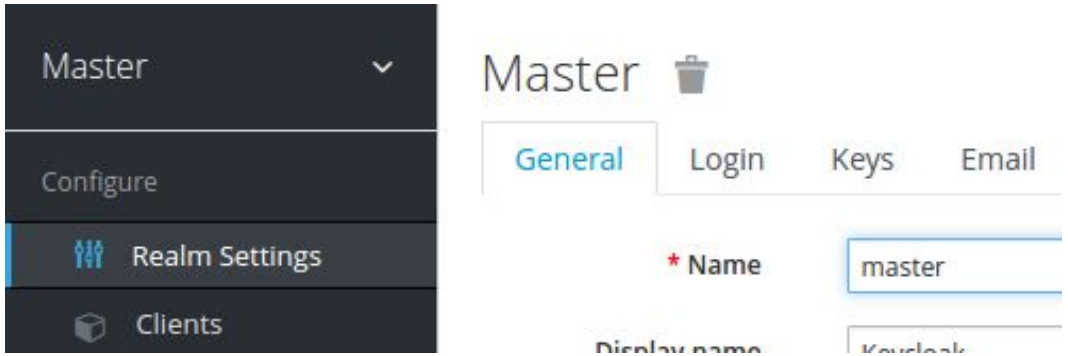
Connecteurs clients

Les connecteur clients Red Hat SSO permettent de sécuriser très facilement les applications et les services. Nous avons des adaptateurs disponibles pour un certain nombre de plateformes et de langages de programmation, mais s'il n'y en a pas un de disponible pour la plateforme de votre choix, ne vous inquiétez pas. RH SSO est construit sur des protocoles standards, vous pouvez donc utiliser n'importe quelle bibliothèque de ressources OpenID Connect SAML 2.0.

https://access.redhat.com/documentation/en-us/red_hat_single_sign-on/7.4/html/securing_applications_and_services_guide/index

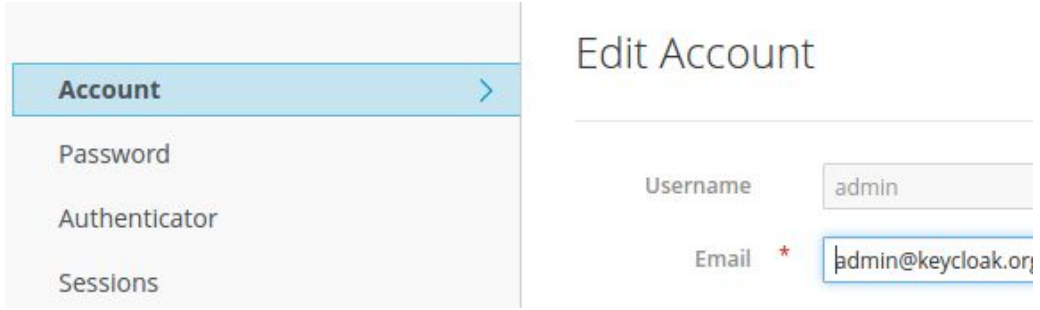
Console administrateur

- Grâce à la console d'administration, les administrateurs peuvent gérer de manière centralisée tous les aspects du serveur Red Hat SSO.
- ils peuvent activer et désactiver diverses fonctionnalités. Ils peuvent configurer le courtage d'identité et la fédération d'utilisateurs.
- Ils peuvent créer et gérer des applications et des services, et définir des politiques d'autorisation précises.
- Ils peuvent également gérer les utilisateurs, y compris les autorisations et les sessions.



Console client

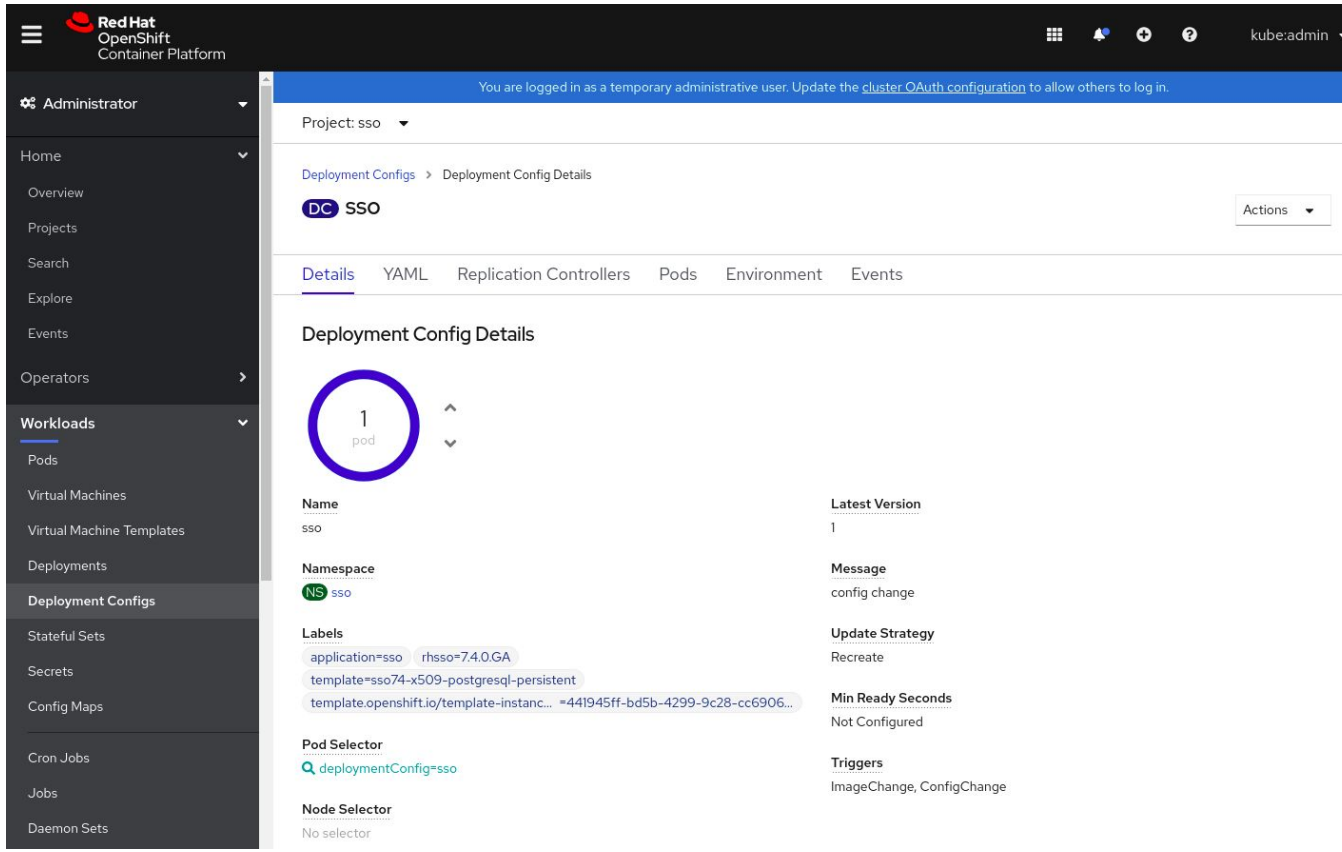
- Grâce à la console client, les utilisateurs peuvent gérer leurs propres comptes. Ils peuvent mettre à jour leur profil, modifier leurs mots de passe et mettre en place une authentification à deux facteurs.
- Les utilisateurs peuvent également gérer des sessions et consulter l'historique du compte.
- Si vous avez activé la connexion sociale ou le courtage d'identité, les utilisateurs peuvent également relier leurs comptes à des fournisseurs supplémentaires pour leur permettre de s'authentifier sur le même compte auprès de différents fournisseurs d'identité.



Red Hat

Pourquoi RH-SSO sur Openshift ?

- RH-SSO est inclus avec Openshift
- Procédure de mise à jour facile, transparent pour les usagers
  - Installe nouveau conteneur
  - Importe les realms
  - Modification de la route
- Évolutivité
  - D'un simple clic, j'augmente les capacités de mon cluster RH-SSO
- Déléguer la gestion des certificats SSL à Openshift



https://github.com/masauve-gitops/rhsso

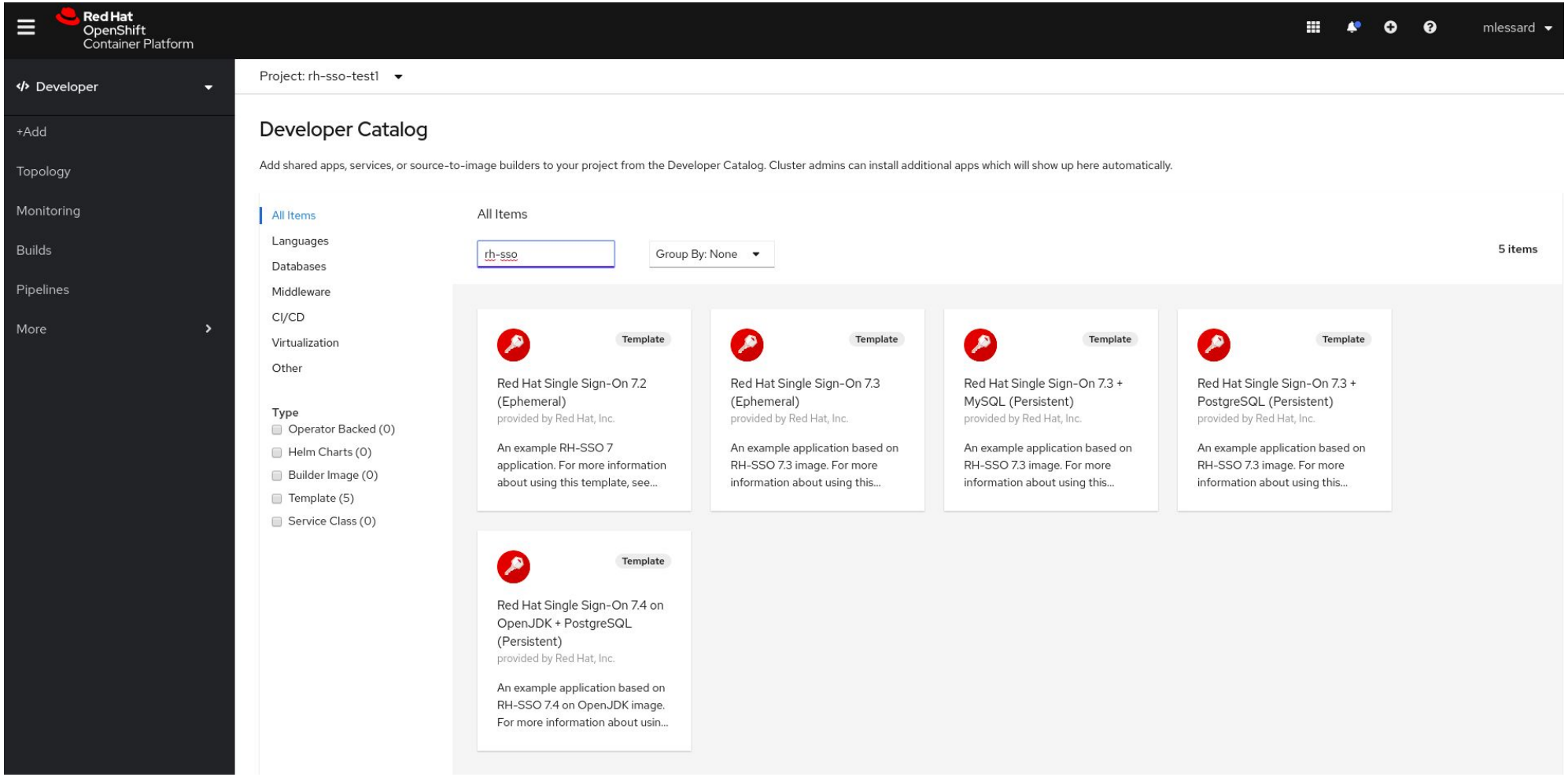
Red Hat

Démonstrations

Prerequis : Certifcate SSL !!

Red Hat

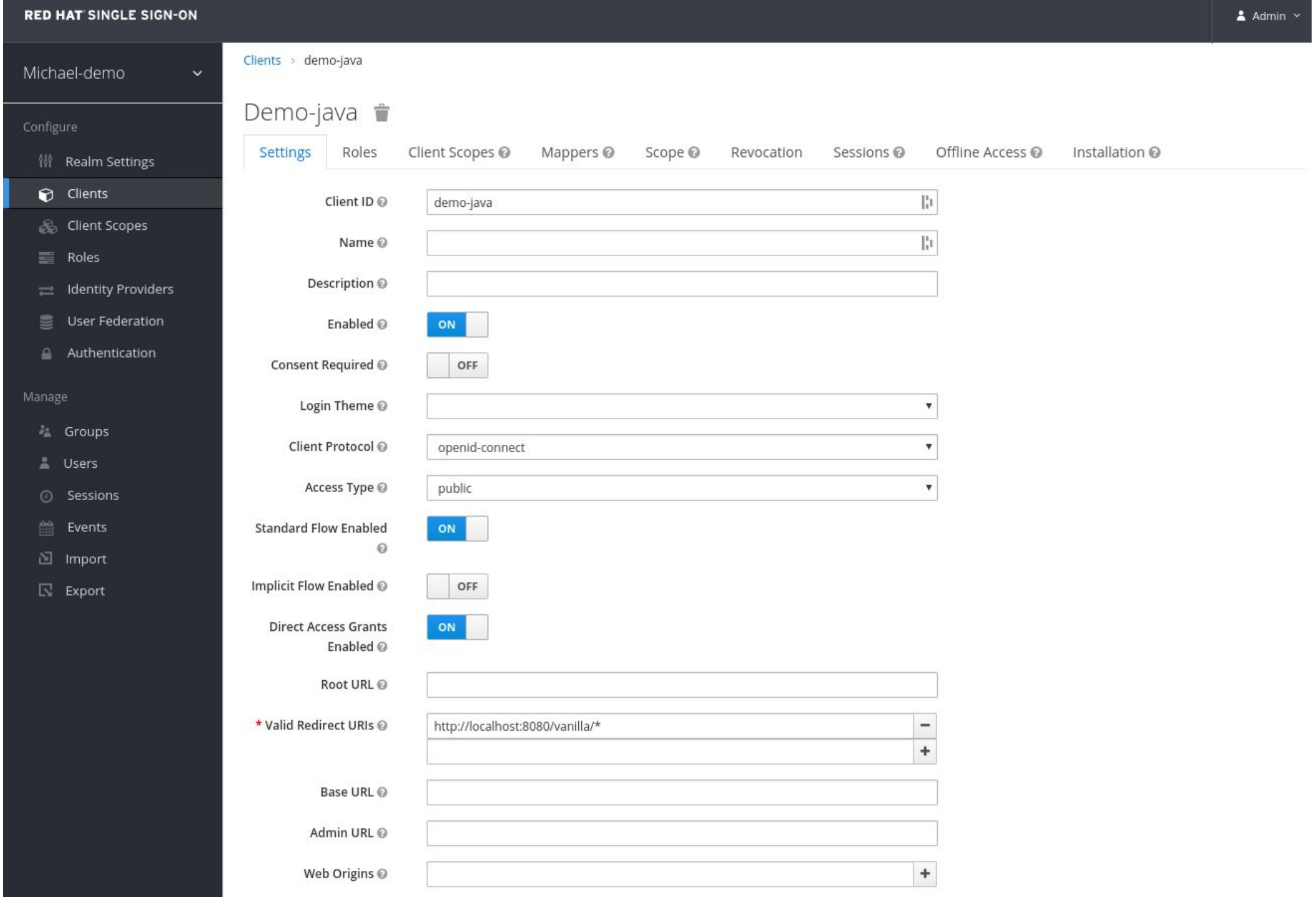
Démonstration #1 :: Installation RH-SSO sous Openshift



Red Hat

Démonstration #2 :: RH-SSO avec Java

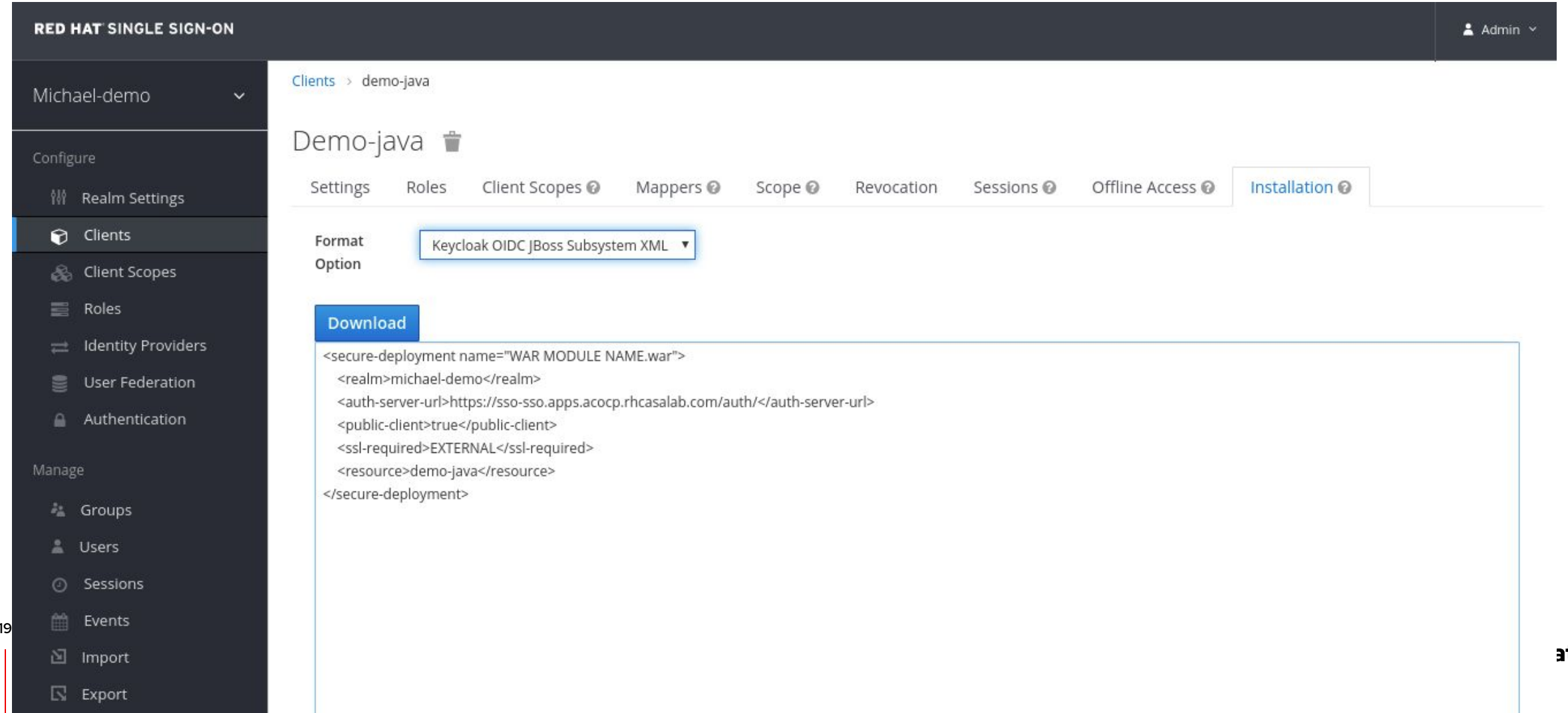
- Mise en place :
  - https://access.redhat.com/documentation/en-us/red_hat_single_sign-on/7.4/html/getting_started_guide/securing_a_jboss_servlet_application


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Clients > demo-java
Demo-java
Settings
Roles
Client Scopes
Mappers
Scope
Revocation
Sessions
Offline Access
Installation
Client ID
demo-java
Name
Description
Enabled
ON
Consent Required
OFF
Login Theme
Client Protocol
openid-connect
Access Type
public
Standard Flow Enabled
ON
Implicit Flow Enabled
OFF
Direct Access Grants Enabled
ON
Root URL
* Valid Redirect URIs
http://localhost:8080/vanilla/*
Base URL
Admin URL
Web Origins

Red Hat

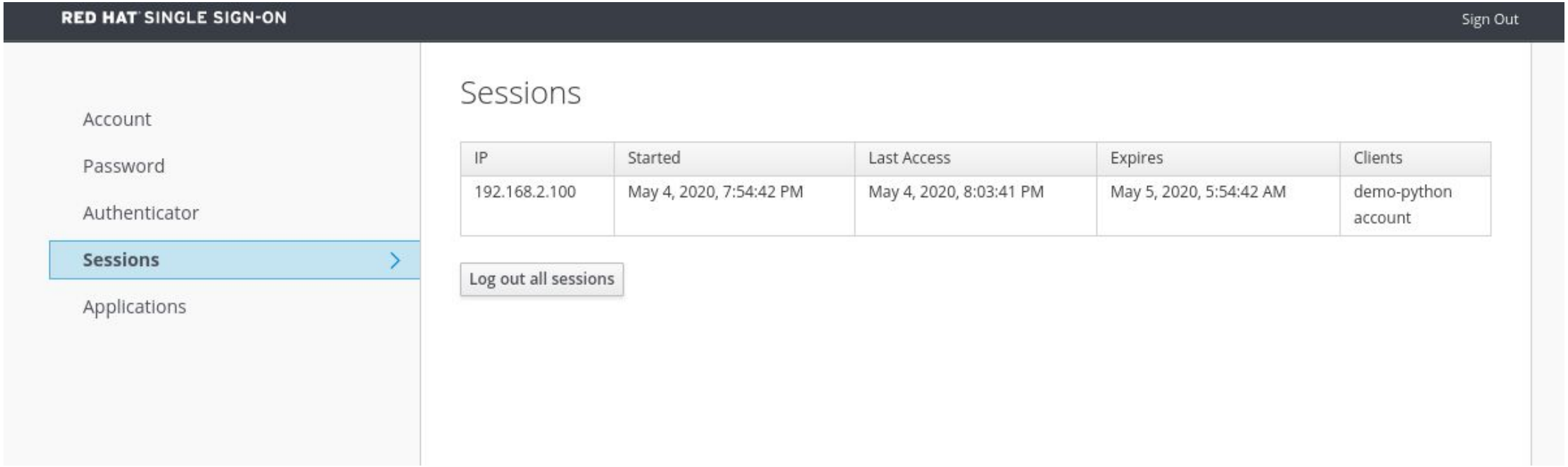
Configurer l'application Jboss

vim standalone/configuration/standalone.xml



Console client

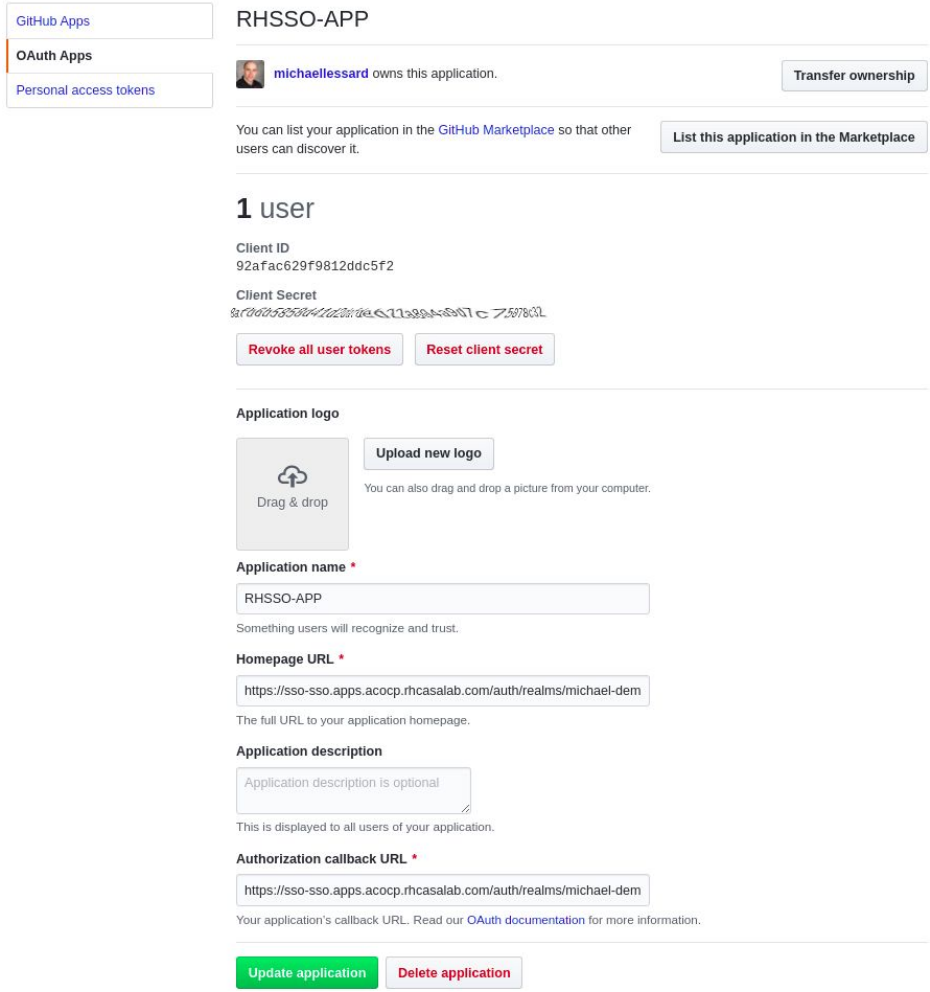
https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/account



Red Hat

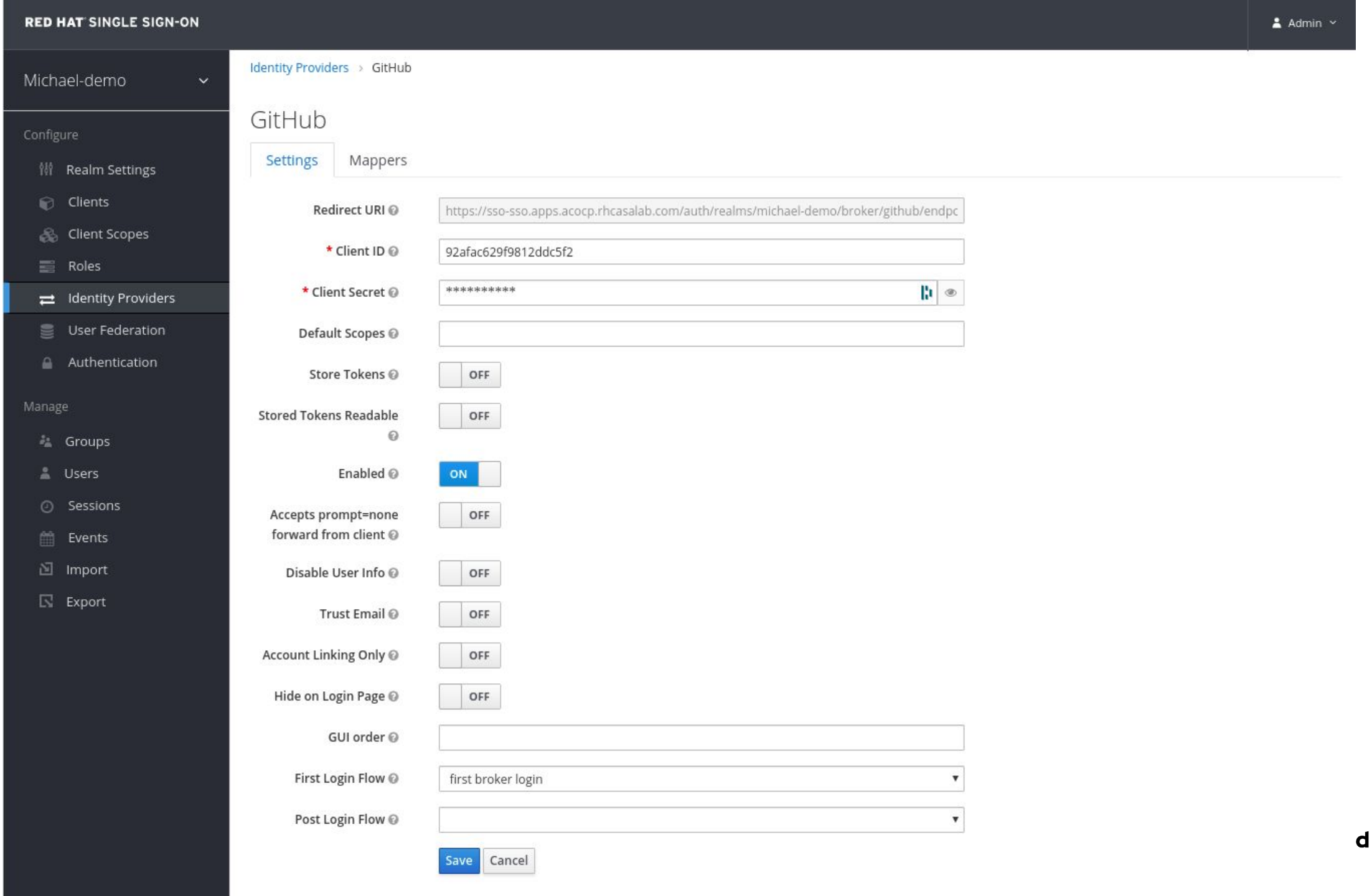
Démonstration #3 - Ajout du identité social - github

Sur Github
Settings → Developer settings → Register a new OAuth application.

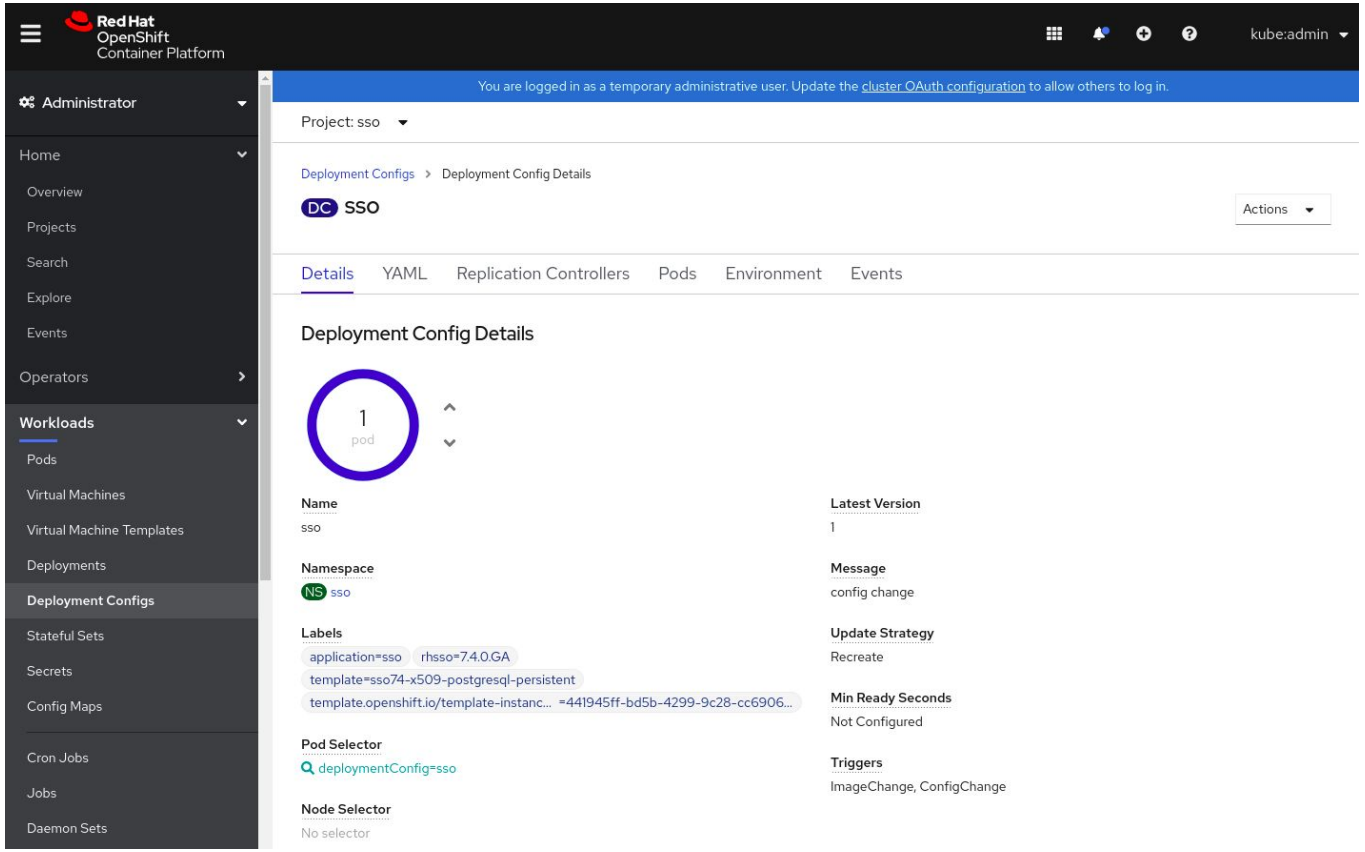


Source:
Insert source data here
Insert source data here
Insert source data here

Red Hat


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › GitHub
GitHub
Settings
Mappers
Redirect URI
https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/broker/github/endpc
* Client ID
92afac629f9812ddc5f2
* Client Secret
****
Default Scopes
Store Tokens
OFF
Stored Tokens Readable
OFF
Enabled
ON
Accepts prompt=none forward from client
OFF
Disable User Info
OFF
Trust Email
OFF
Account Linking Only
OFF
Hide on Login Page
OFF
GUI order
First Login Flow
first broker login
Post Login Flow
Save
Cancel

Démonstration #4:: RH-SSO sur Openshift



Red Hat

Merci !

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc

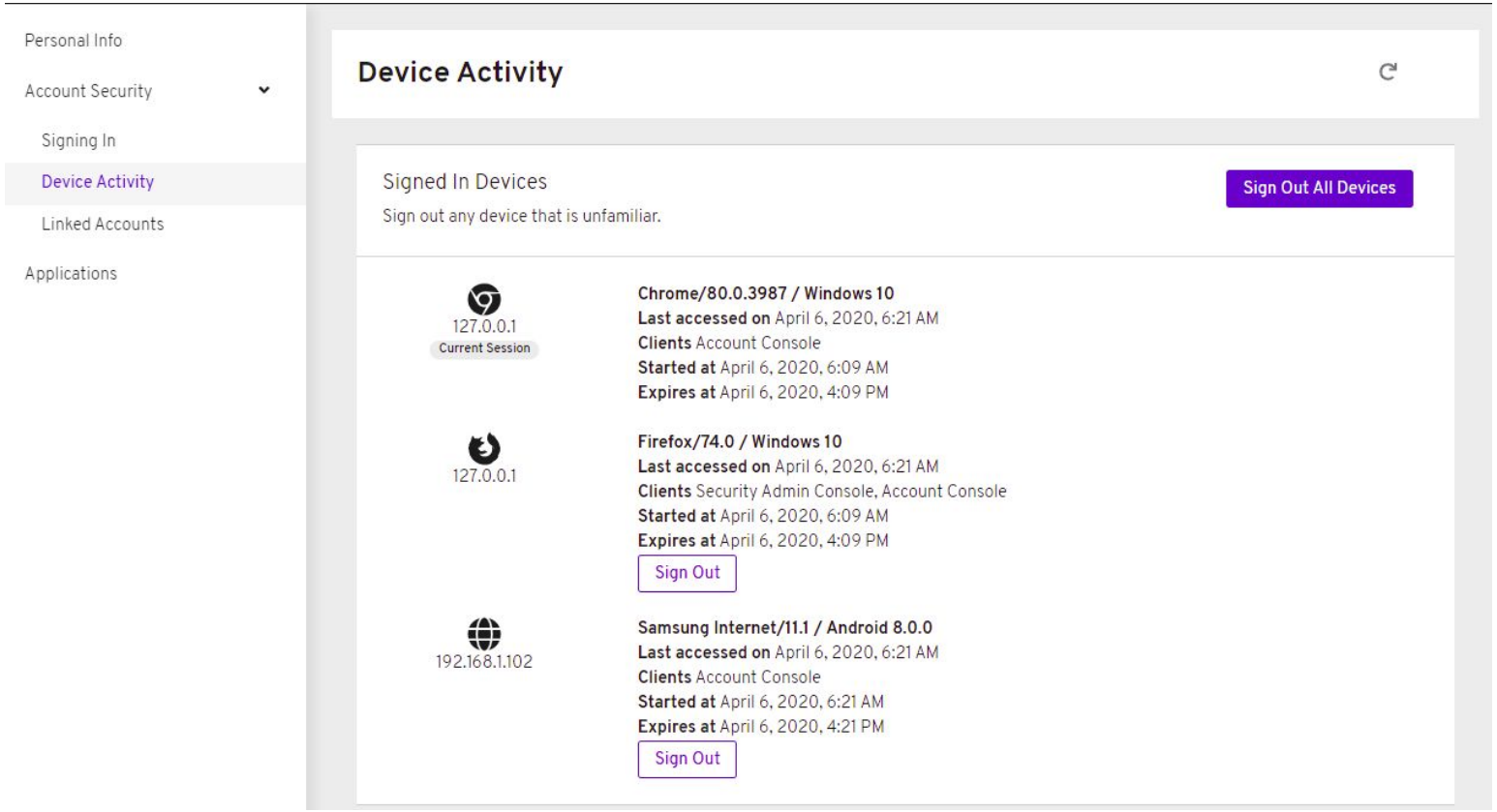

twitter.com/RedHat

Red Hat

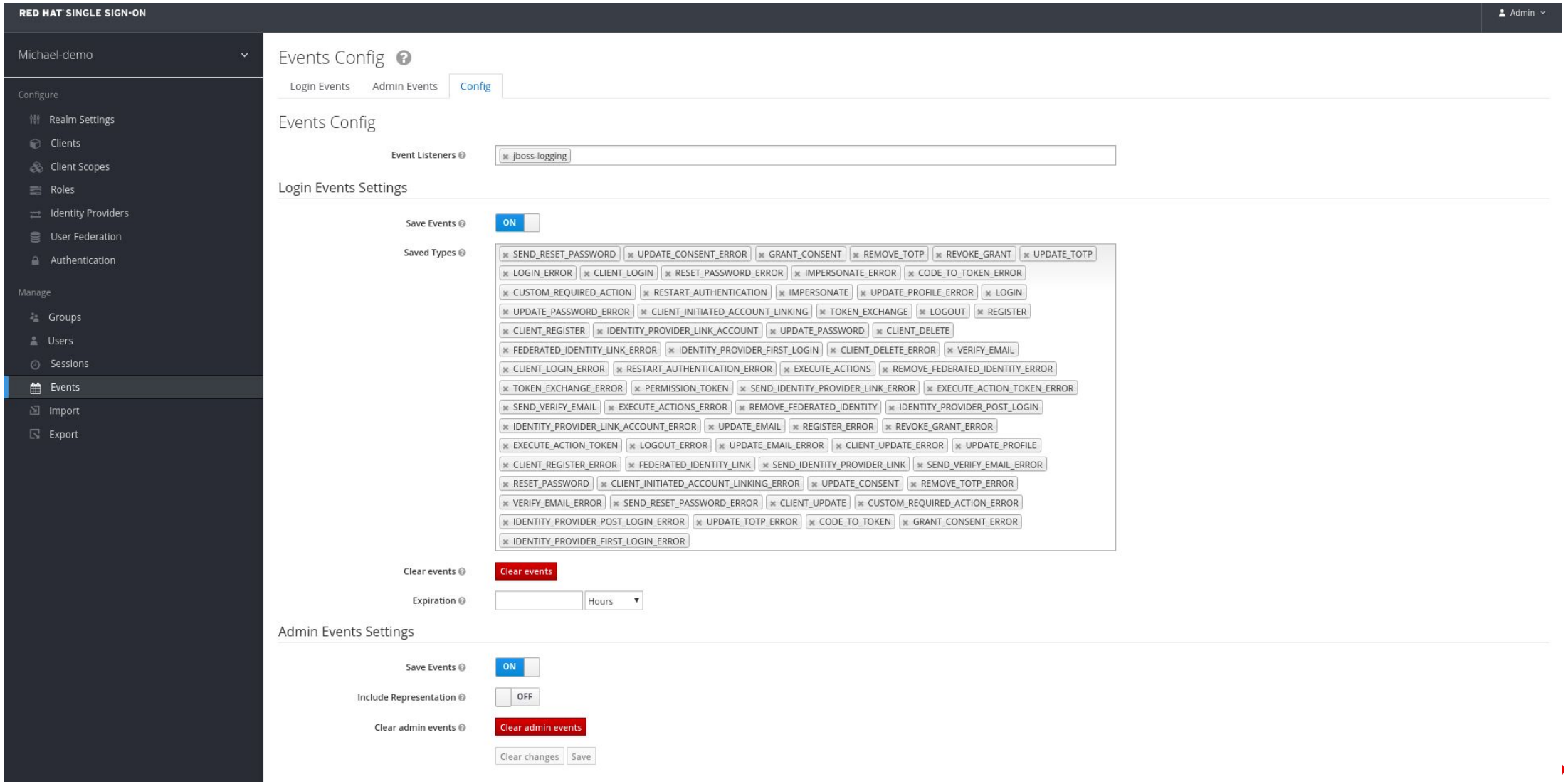
EXTRA

Red Hat

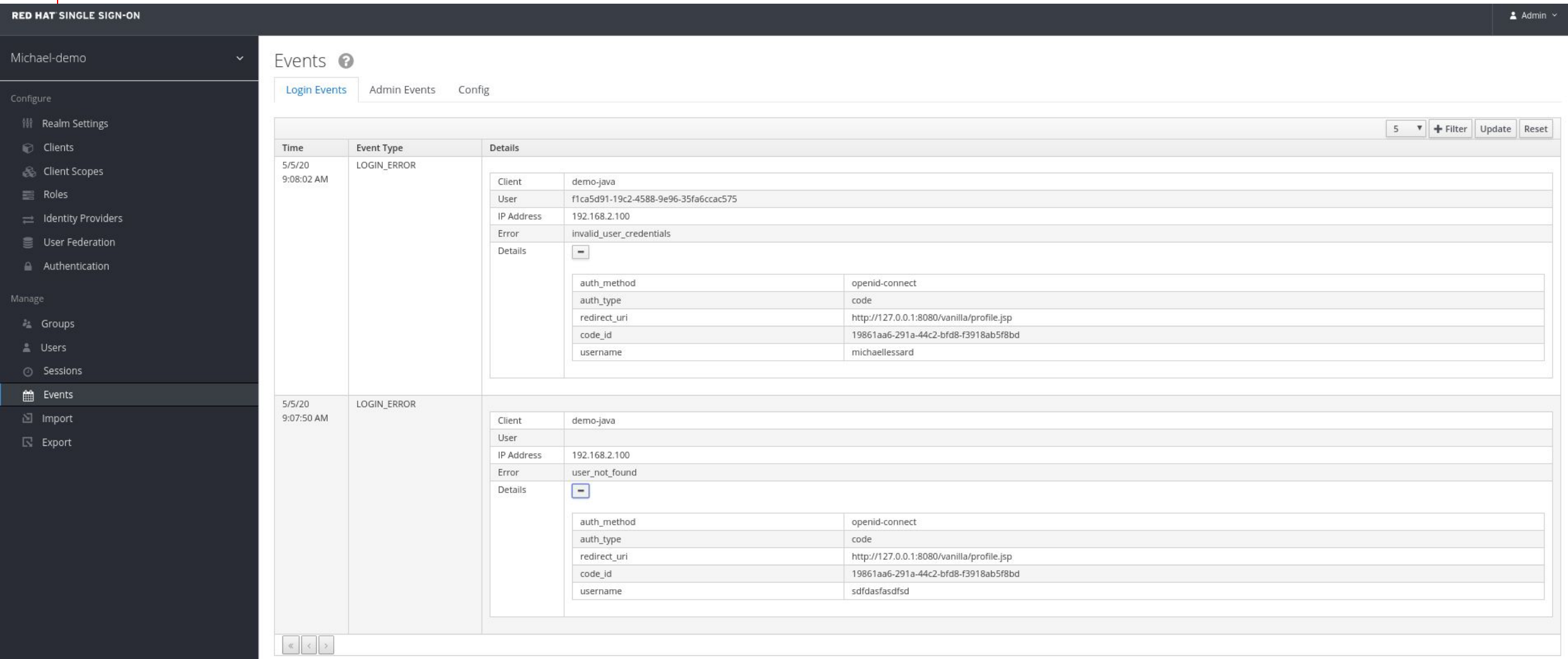
RH-SSO 7.4 – Nouvelle console client – Tech preview



LOG

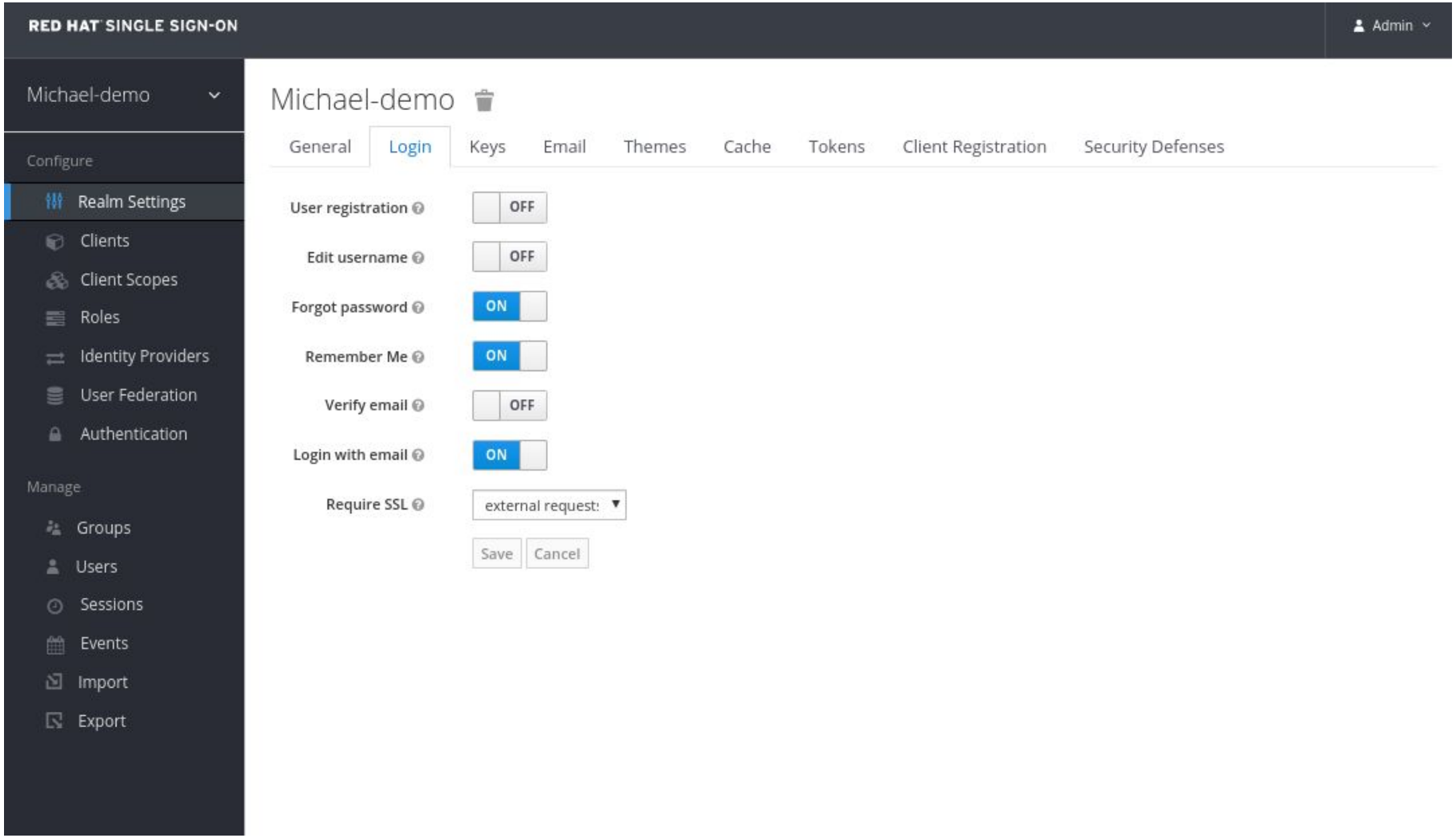

RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Events Config
Login Events
Admin Events
Config
Events Config
Event Listeners
jboss-logging
Login Events Settings
Save Events
ON
Saved Types
SEND_RESET_PASSWORD
UPDATE_CONSENT_ERROR
GRANT_CONSENT
REMOVE_TOTP
REVOKE_GRANT
UPDATE_TOTP
LOGIN_ERROR
CLIENT_LOGIN
RESET_PASSWORD_ERROR
IMPERSONATE_ERROR
CODE_TO_TOKEN_ERROR
CUSTOM_REQUIRED_ACTION
RESTART_AUTHENTICATION
IMPERSONATE
UPDATE_PROFILE_ERROR
LOGIN
UPDATE_PASSWORD_ERROR
CLIENT_INITIATED_ACCOUNT_LINKING
TOKEN_EXCHANGE
LOGOUT
REGISTER
CLIENT_REGISTER
IDENTITY_PROVIDER_LINK_ACCOUNT
UPDATE_PASSWORD
CLIENT_DELETE
FEDERATED_IDENTITY_LINK_ERROR
IDENTITY_PROVIDER_FIRST_LOGIN
CLIENT_DELETE_ERROR
VERIFY_EMAIL
CLIENT_LOGIN_ERROR
RESTART_AUTHENTICATION_ERROR
EXECUTE_ACTIONS
REMOVE_FEDERATED_IDENTITY_ERROR
TOKEN_EXCHANGE_ERROR
PERMISSION_TOKEN
SEND_IDENTITY_PROVIDER_LINK_ERROR
EXECUTE_ACTION_TOKEN_ERROR
SEND_VERIFY_EMAIL
EXECUTE_ACTIONS_ERROR
REMOVE_FEDERATED_IDENTITY
IDENTITY_PROVIDER_POST_LOGIN
IDENTITY_PROVIDER_LINK_ACCOUNT_ERROR
UPDATE_EMAIL
REGISTER_ERROR
REVOKE_GRANT_ERROR
EXECUTE_ACTION_TOKEN
LOGOUT_ERROR
UPDATE_EMAIL_ERROR
CLIENT_UPDATE_ERROR
UPDATE_PROFILE
CLIENT_REGISTER_ERROR
FEDERATED_IDENTITY_LINK
SEND_IDENTITY_PROVIDER_LINK
SEND_VERIFY_EMAIL_ERROR
RESET_PASSWORD
CLIENT_INITIATED_ACCOUNT_LINKING_ERROR
UPDATE_CONSENT
REMOVE_TOTP_ERROR
VERIFY_EMAIL_ERROR
SEND_RESET_PASSWORD_ERROR
CLIENT_UPDATE
CUSTOM_REQUIRED_ACTION_ERROR
IDENTITY_PROVIDER_POST_LOGIN_ERROR
UPDATE_TOTP_ERROR
CODE_TO_TOKEN
GRANT_CONSENT_ERROR
IDENTITY_PROVIDER_FIRST_LOGIN_ERROR
Clear events
Clear events
Expiration
Hours
Admin Events Settings
Save Events
ON
Include Representation
OFF
Clear admin events
Clear admin events
Clear changes
Save

Red Hat


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Events
Login Events
Admin Events
Config
5
Filter
Update
Reset
Time
Event Type
Details
5/5/20 9:08:02 AM
LOGIN_ERROR
Client demo-java
User f1ca5d91-19c2-4588-9e96-35fa6ccac575
IP Address 192.168.2.100
Error invalid_user_credentials
Details
auth_method openid-connect
auth_type code
redirect_uri http://127.0.0.1:8080/vanilla/profile.jsp
code_id 19861aa6-291a-44c2-bfd8-f3918ab5f8bd
username michaellessard
5/5/20 9:07:50 AM
LOGIN_ERROR
Client demo-java
User
IP Address 192.168.2.100
Error user_not_found
Details
auth_method openid-connect
auth_type code
redirect_uri http://127.0.0.1:8080/vanilla/profile.jsp
code_id 19861aa6-291a-44c2-bfd8-f3918ab5f8bd
username sdfdasfasdfsd

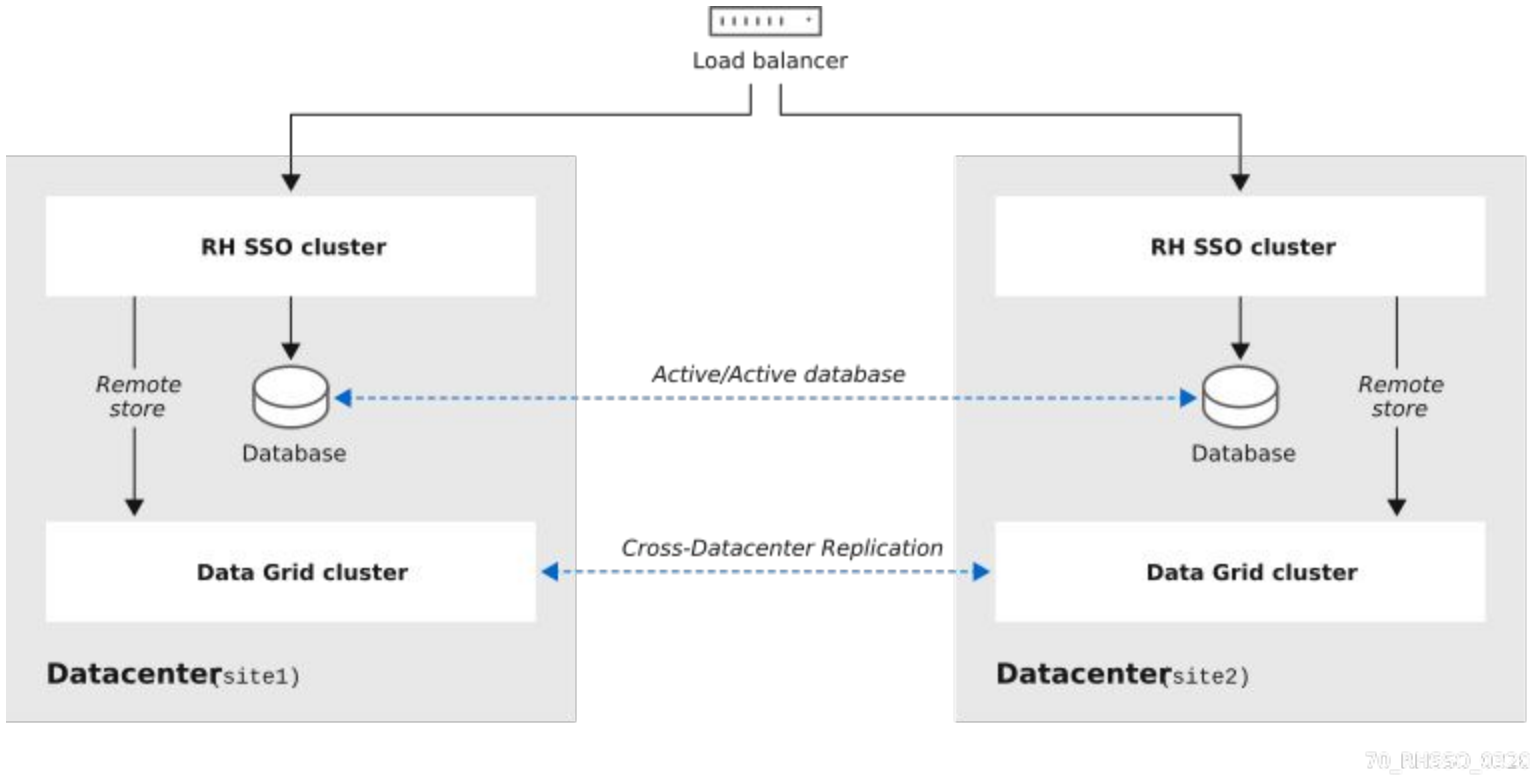
Red Hat

Forget password / remember me

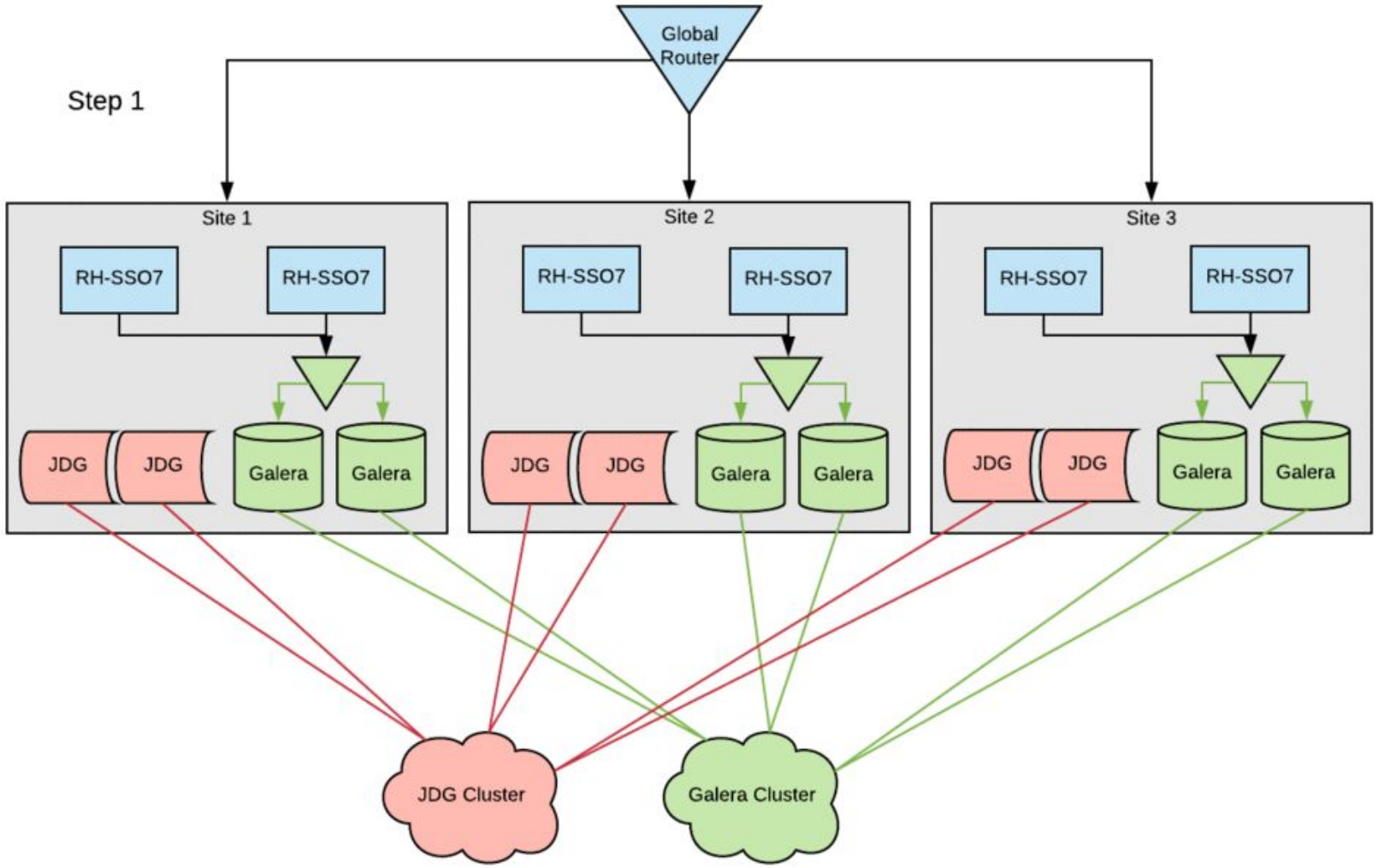


Red Hat

RH-SSO architecture Multi-site



Red Hat



https://developers.redhat.com/blog/2019/02/14/red-hat-sso-high-availability-hybrid-cloud/

Démonstration #2 :: RH-SSO avec Python

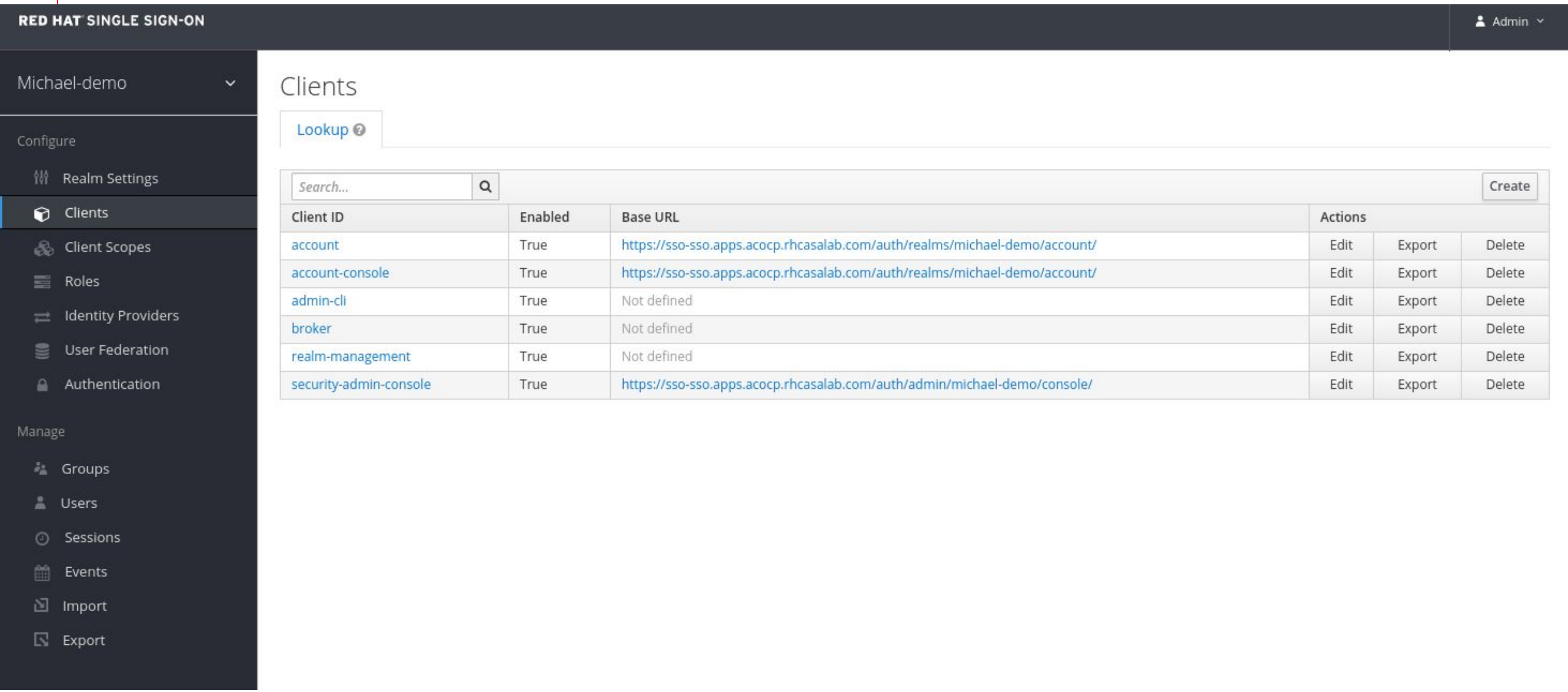
Étapes

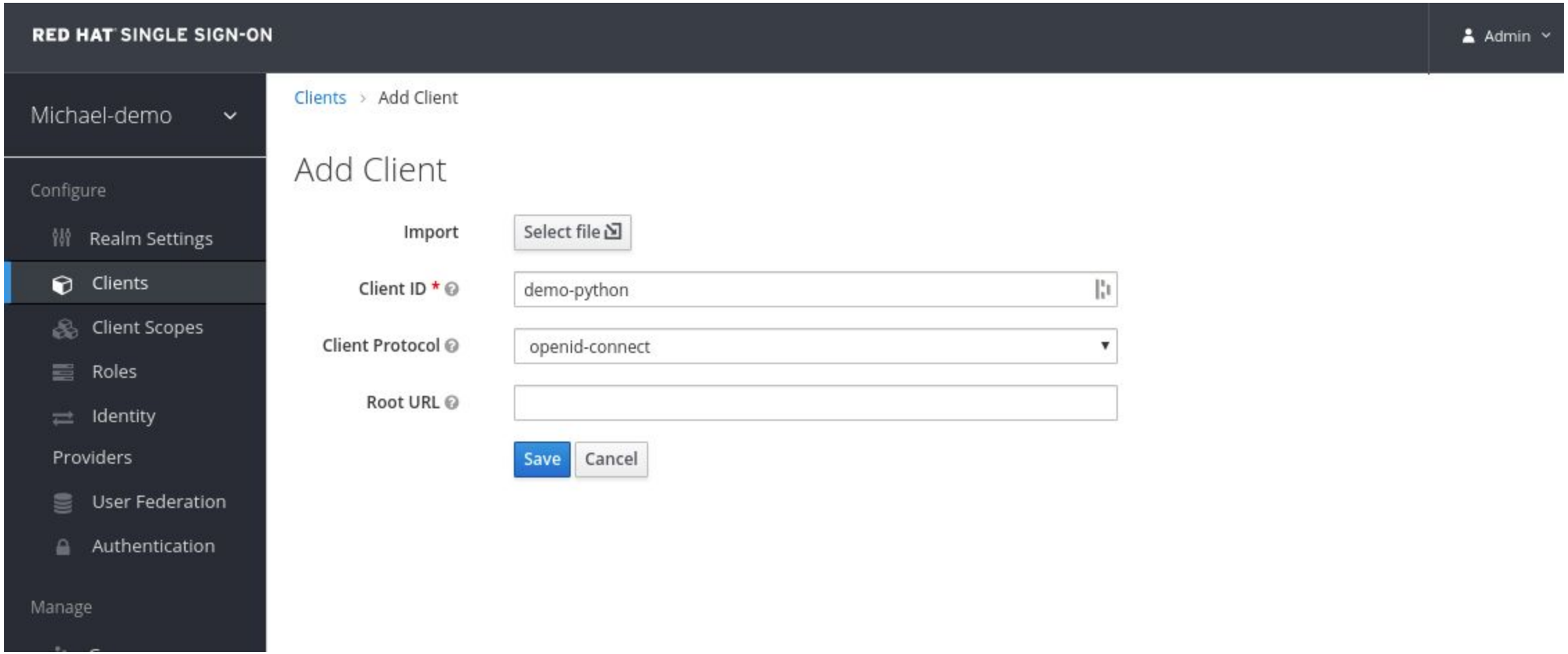
Requis côté Python : librairies oidc

Example : flask-oidc

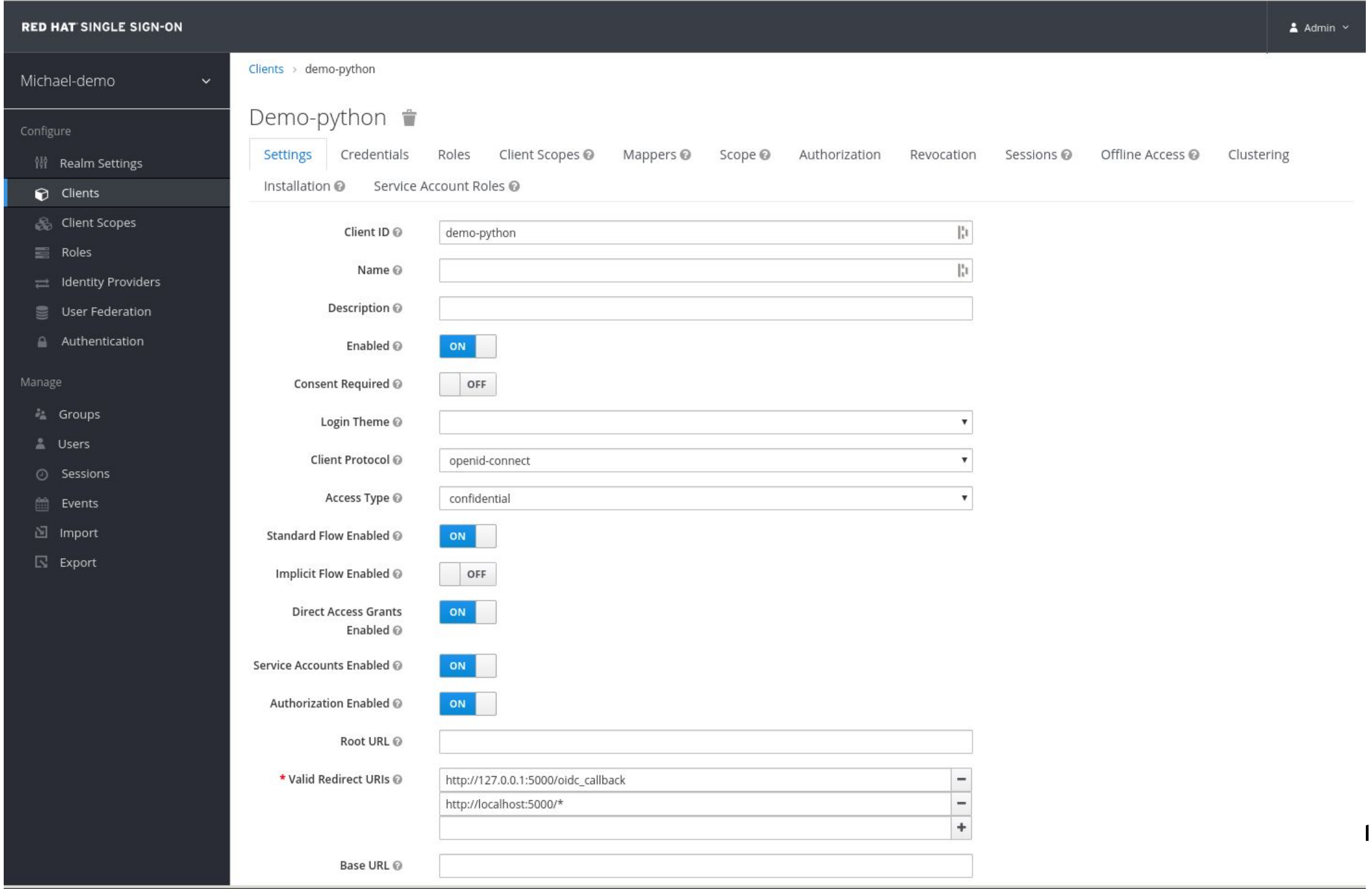
1. Créer un REALM
    a. un realm est pour un ensemble d'applications partagent un domaine commun et une sécurité commune
2. Ajouter un client
    a. Access type
3. Créer un usager

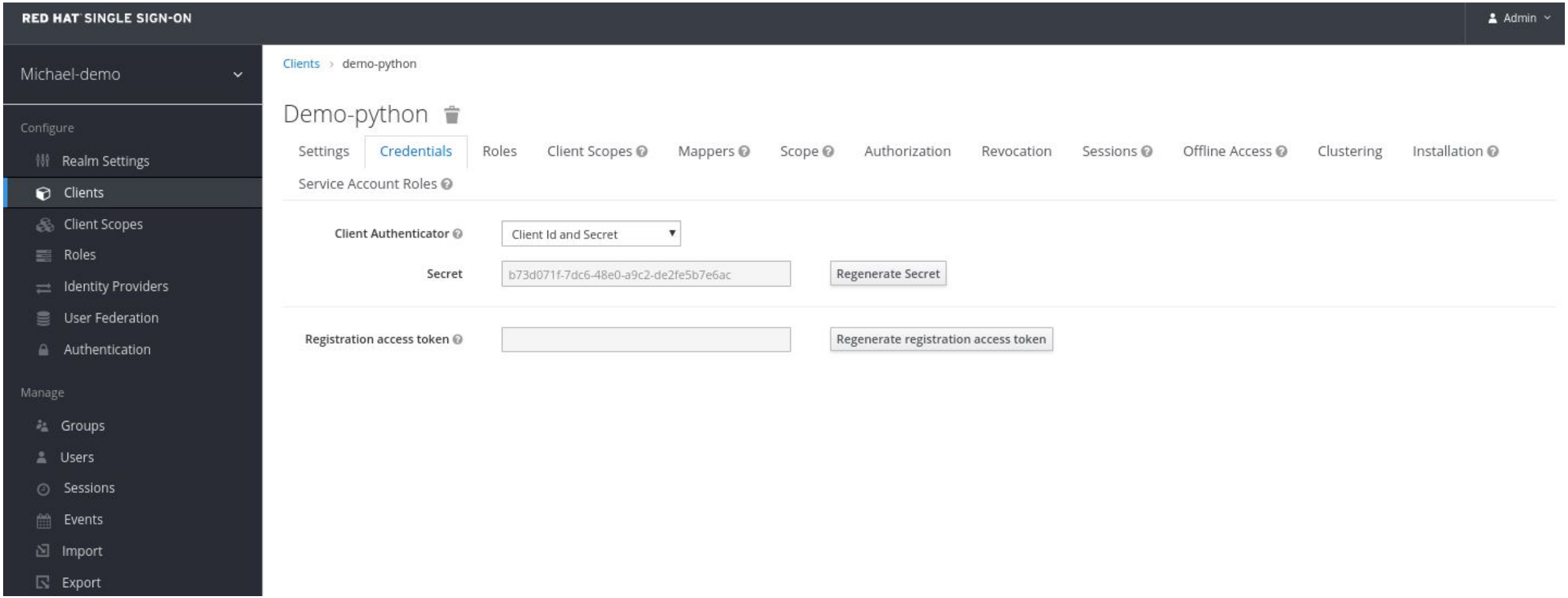
** Attention flask enregistre des informations sur les sessions dans la cache du navigateur (truc - démarrer une session incognito)


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Clients
Lookup
Search...
Create
Client ID
Enabled
Base URL
Actions
account
True
https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/account/
Edit
Export
Delete
account-console
True
https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/account/
Edit
Export
Delete
admin-cli
True
Not defined
Edit
Export
Delete
broker
True
Not defined
Edit
Export
Delete
realm-management
True
Not defined
Edit
Export
Delete
security-admin-console
True
https://sso-sso.apps.acocp.rhcasalab.com/auth/admin/michael-demo/console/
Edit
Export
Delete


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity
Providers
User Federation
Authentication
Manage
Clients > Add Client
Add Client
Import
Select file
Client ID *
demo-python
Client Protocol
openid-connect
Root URL
Save
Cancel

Red Hat


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Clients › demo-python
Demo-python
Settings
Credentials
Roles
Client Scopes
Mappers
Scope
Authorization
Revocation
Sessions
Offline Access
Clustering
Installation
Service Account Roles
Client ID
demo-python
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
ON
Root URL
* Valid Redirect URIs
http://127.0.0.1:5000/oidc_callback
http://localhost:5000/*
Base URL


RED HAT SINGLE SIGN-ON
Admin
Michael-demo
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Clients
demo-python
Demo-python
Settings
Credentials
Roles
Client Scopes
Mappers
Scope
Authorization
Revocation
Sessions
Offline Access
Clustering
Installation
Service Account Roles
Client Authenticator
Client Id and Secret
Secret
b73d071f-7dc6-48e0-a9c2-de2fe5b7e6ac
Regenerate Secret
Registration access token
Regenerate registration access token

Exemple code Python

```
import json
import logging
from flask import Flask, g, redirect
from flask_oidc import OpenIDConnect
import requests
from base64 import b64encode, b64decode, urlsafe_b64encode, urlsafe_b64decode
logging.basicConfig(level=logging.DEBUG)

app = Flask(__name__)
app.config.update({
    'SECRET_KEY': 'b73d071f-7dc6-48e0-a9c2-de2fe5b7e6zz',
    'TESTING': True,
    'DEBUG': True,
    'OIDC_CLIENT_SECRETS': 'oid.json',
    'OIDC_ID_TOKEN_COOKIE_SECURE': False,
    'OIDC_REQUIRE_VERIFIED_EMAIL': False,
    'OIDC_USER_INFO_ENABLED': True,
    'OIDC_OPENID_REALM': 'michael-demo',
    'OIDC_SCOPES': ['openid', 'email', 'profile'],
   'OIDC_INTROSPECTION_AUTH_METHOD': 'client_secret_post'
})
```

```
oidc = OpenIDConnect(app)

@app.route('/')
def index():
    if oidc.user_loggedin:
        return 'Welcome %s' % oidc.user_getfield('preferred_username')
    Else:
        return '<a href="/login">Loggin here</a>'

@app.route('/login')
@oidc.require_login
def login():
    return 'Welcome %s' % oidc.user_getfield('preferred_username') + '<br><a href="/logout">Logout</a>'

@app.route('/logout')
#@oidc.require_login
def logout():
    oidc.logout()
    return redirect("/")

if __name__ == '__main__':
    app.run('localhost', port=5000)
```

oid.json

```
{
  "web": {
      "auth_uri": "https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/protocol/openid-connect/auth",
      "client_id": "demo-python",
      "client_secret": "b73d071f-7dc6-48e0-a9c2-de2fe5b7e6zz",
      "redirect_uris": [
          "http://localhost:5000/oidc_callback"
      ],
      "userinfo_uri": "https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/protocol/openid-connect/userinfo",
      "token_uri": "https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/protocol/openid-connect/token",
      "token_introspection_uri": "https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo/protocol/openid-connect/token/introspect",
      "issuer": "https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-demo"
  }
}
```