

Introduction à RH-SSO



Centraliser la gestion d'identité et sécuriser vos applications

Michael Lessard
Architecte Principal
mlessard@redhat.com

 michaellessard

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'authentications externes - Gitlab, Twitter, etc.
 - Évolutivité avec Openshift

AMÉLIORE LA SÉCURITÉ EN LAISSANT LA GESTION DES IDENTIFIANTS AUX EXPERTS



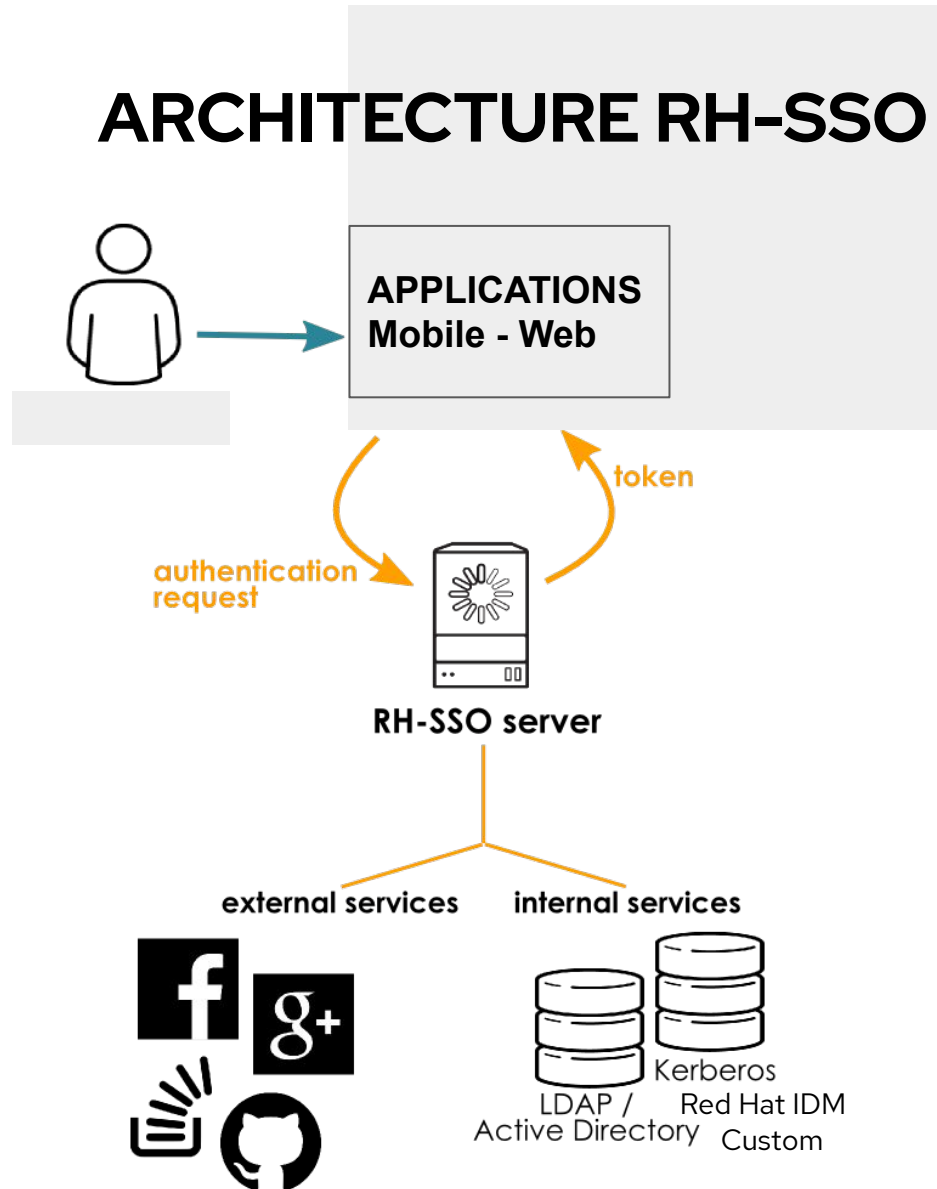
RED HAT®
SSO

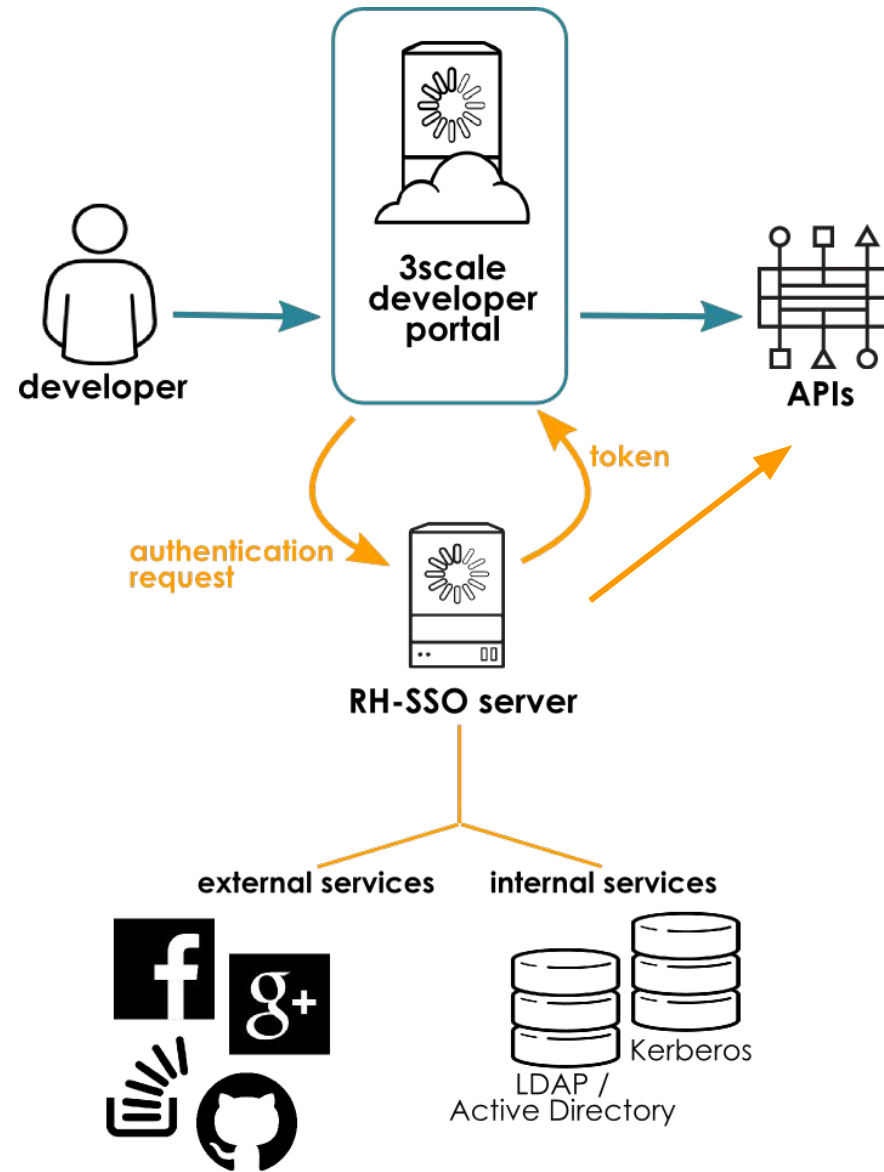
Sécuriser les applications web et fournir des capacités de connexion unique basées sur des standards populaires tels que SAML 2.0, OpenID Connect et OAuth 2.0.

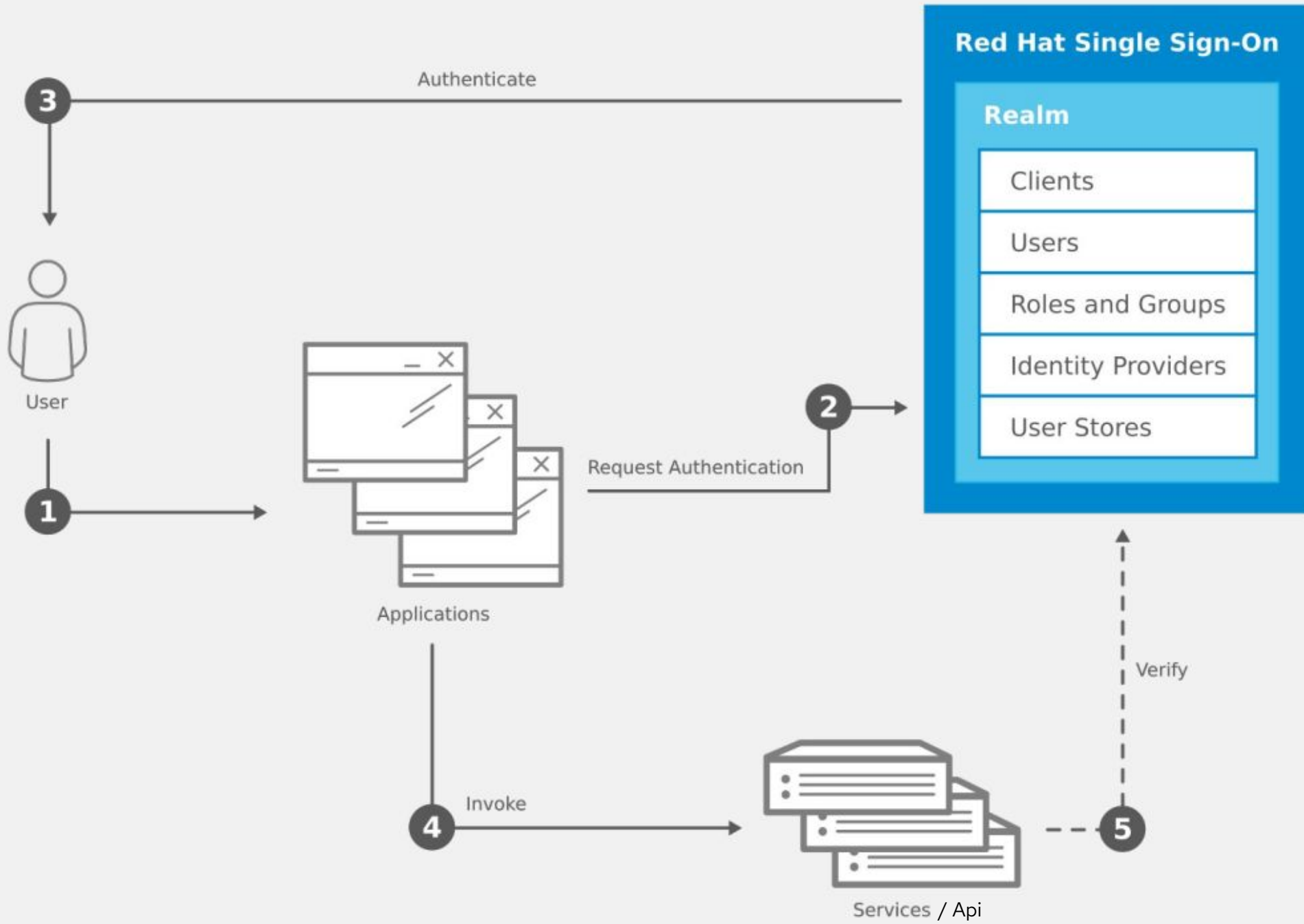
SIMPLE D'UTILISATION ET DÉPLOIEMENT RAPIDE

- ▶ 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







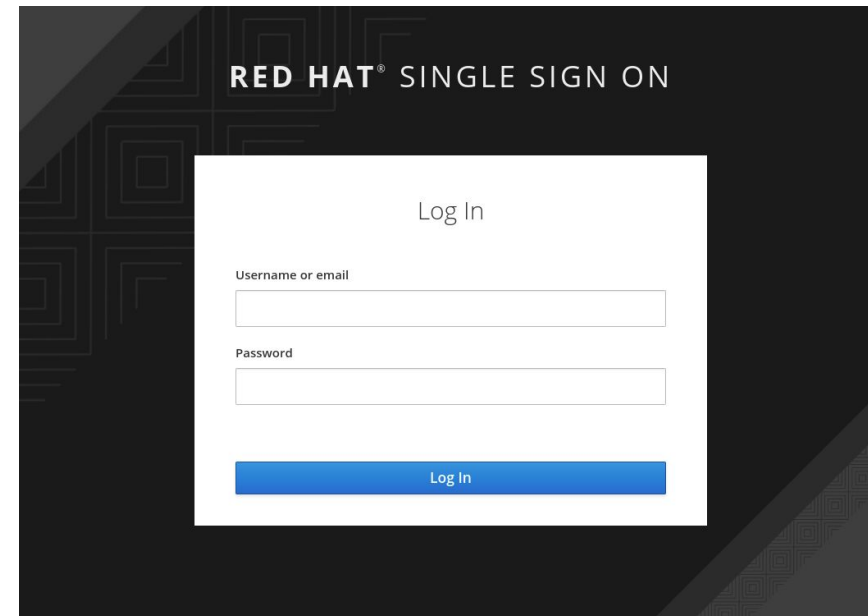
Authentification unique (Single-Sign On)

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.



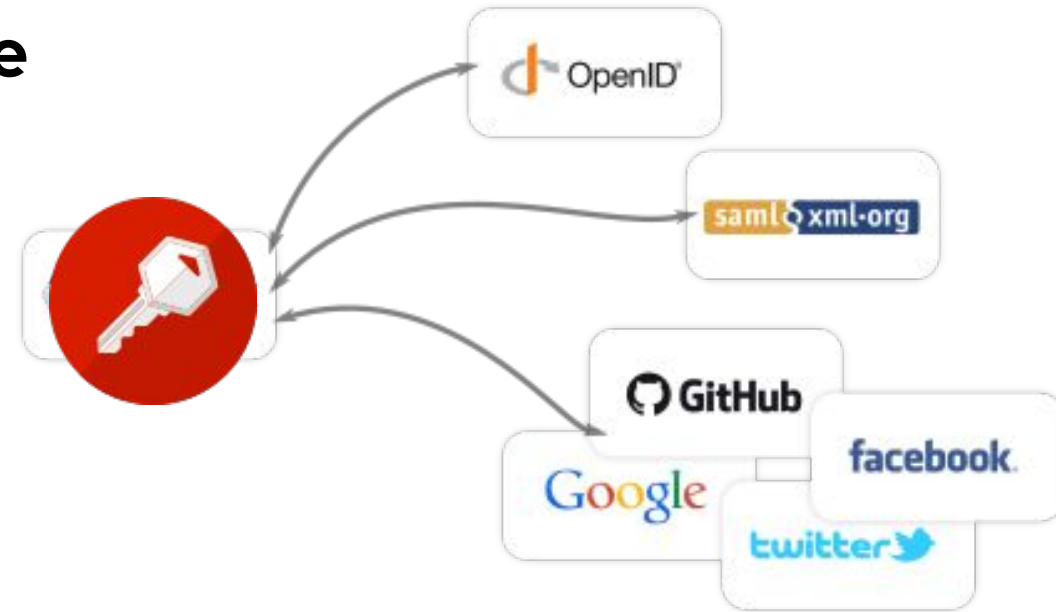
Le flux SAML est indépendant de OAuth 2.0, et repose sur l'échange de messages pour l'authentification au format XML SAML (au lieu du format JWT). Protocole mature.

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

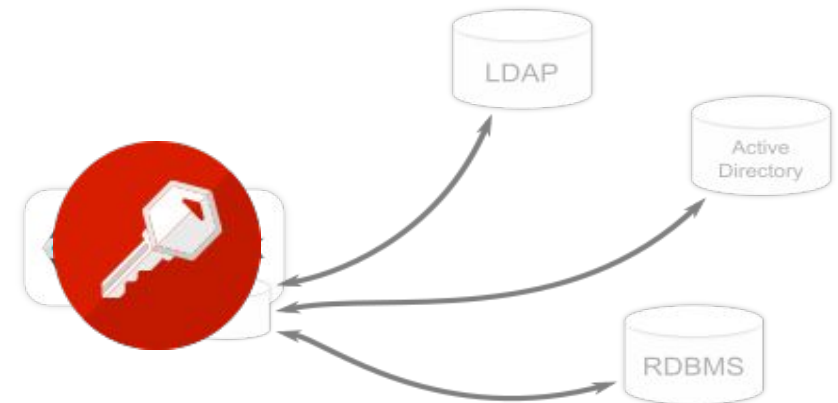
L'activation de la connexion aux réseaux sociaux est facile à ajouter via la console d'administration. Il suffit de sélectionner le réseau social que vous souhaitez ajouter. **Aucun code ou modification de votre application n'est nécessaire.**

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.

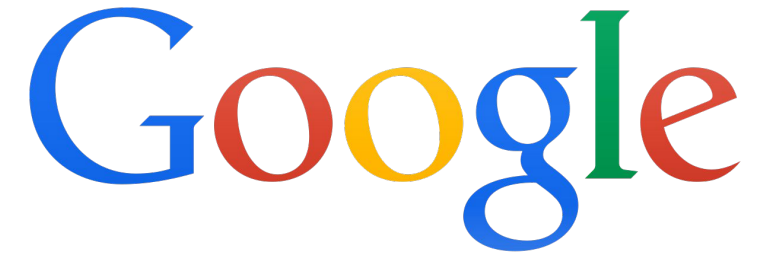
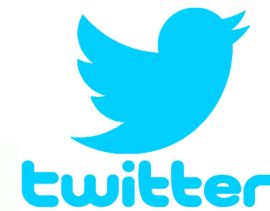


Fédération des utilisateurs

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

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

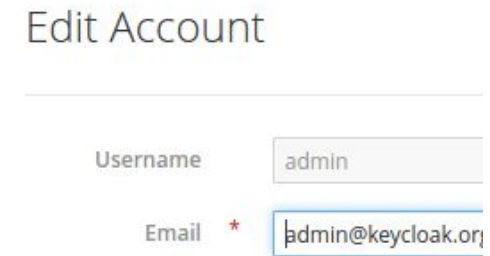
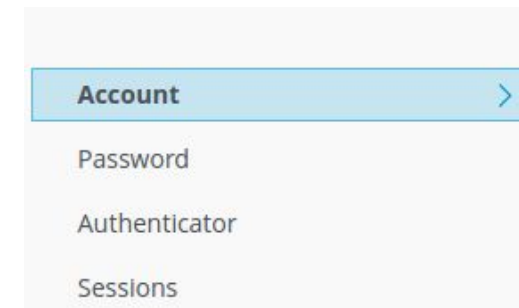
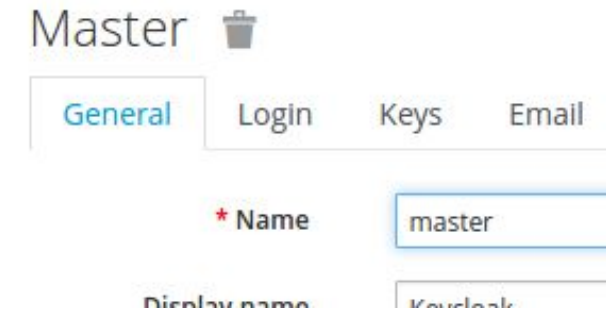
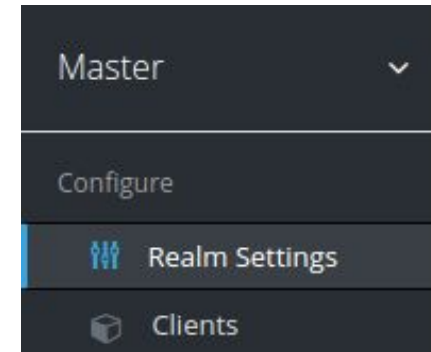


Console administrateur

- ▶ 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.

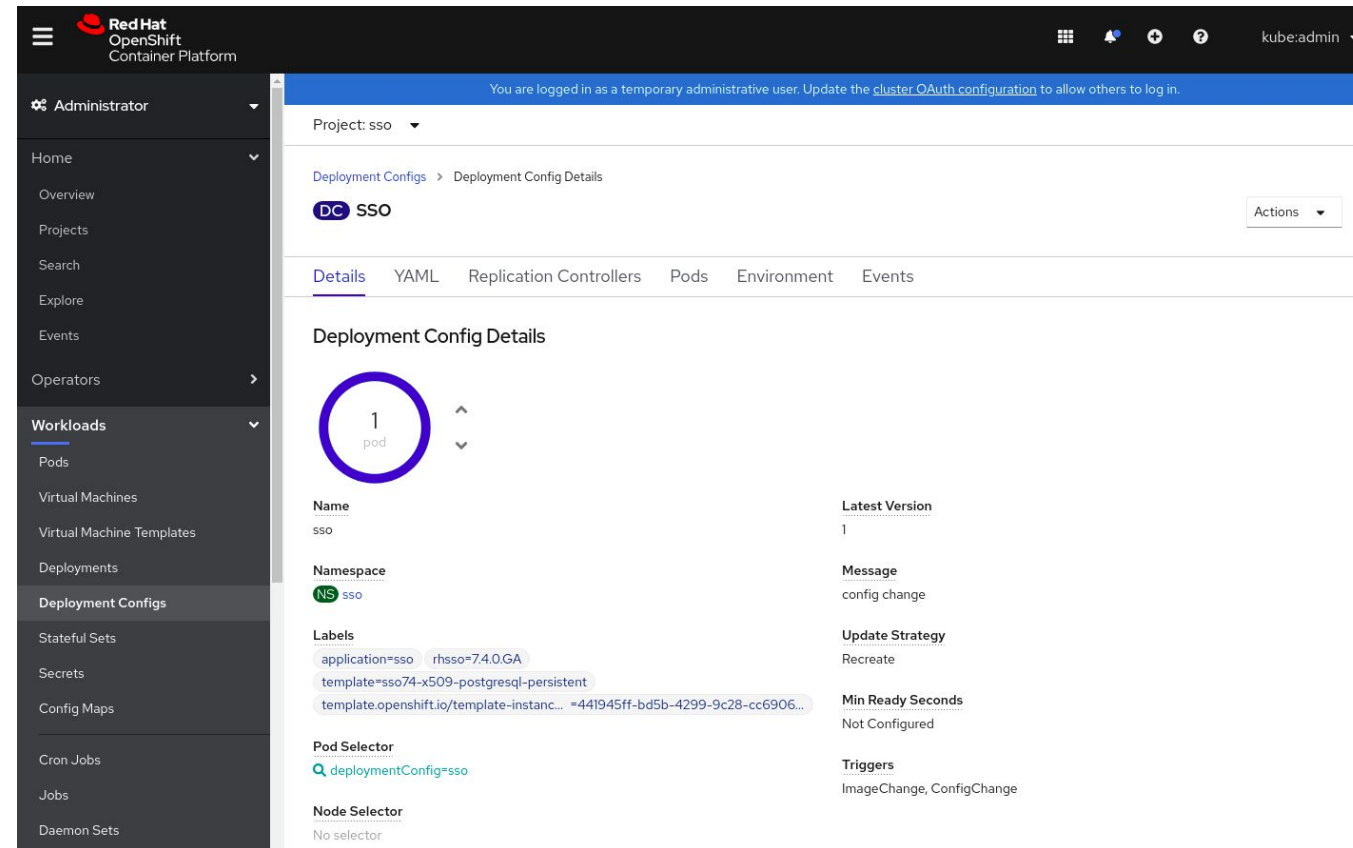
Console client

- ▶ 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.



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



Démonstrations



Prerequis : Certifcate SSL !!

Démonstration #1 :: Installation RH-SSO sous Openshift

The screenshot displays the Red Hat OpenShift Developer Catalog interface. The top navigation bar includes the Red Hat OpenShift logo and the text 'Container Platform'. The left sidebar contains navigation options: Developer, +Add, Topology, Monitoring, Builds, Pipelines, and More. The main content area shows the 'Developer Catalog' for project 'rh-ssso-test1'. A search filter 'rh-ssso' is applied, resulting in 5 items. The items are categorized as 'Template' and include:

- Red Hat Single Sign-On 7.2 (Ephemeral) provided by Red Hat, Inc. An example RH-SSO 7 application. For more information about using this template, see...
- Red Hat Single Sign-On 7.3 (Ephemeral) provided by Red Hat, Inc. An example application based on RH-SSO 7.3 image. For more information about using this...
- Red Hat Single Sign-On 7.3 + MySQL (Persistent) provided by Red Hat, Inc. An example application based on RH-SSO 7.3 image. For more information about using this...
- Red Hat Single Sign-On 7.3 + PostgreSQL (Persistent) provided by Red Hat, Inc. An example application based on RH-SSO 7.3 image. For more information about using this...
- Red Hat Single Sign-On 7.4 on OpenJDK + PostgreSQL (Persistent) provided by Red Hat, Inc. An example application based on RH-SSO 7.4 on OpenJDK image. For more information about usin...

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

Michael-demo

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 ?

Configure

Realm Settings

Clients

Client Scopes

Roles

Identity Providers

User Federation

Authentication

Manage

Groups

Users

Sessions

Events

Import

Export

Configurer l'application Jboss

vim standalone/configuration/standalone.xml

The screenshot displays the Red Hat Single Sign-On administration interface. The top navigation bar shows 'RED HAT SINGLE SIGN-ON' and a user profile 'Admin'. The left sidebar contains a navigation menu with sections: 'Michael-demo', 'Configure' (with sub-items: Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication), and 'Manage' (with sub-items: Groups, Users, Sessions, Events, Import, Export). The main content area is titled 'Clients > demo-java' and shows the configuration for 'Demo-java'. A breadcrumb trail 'Clients > demo-java' is visible. Below the title, there are tabs for 'Settings', 'Roles', 'Client Scopes', 'Mappers', 'Scope', 'Revocation', 'Sessions', 'Offline Access', and 'Installation'. The 'Installation' tab is active. Under the 'Format Option' dropdown, 'Keycloak OIDC JBoss Subsystem XML' is selected. A blue 'Download' button is present above a text area containing the following XML code:

```
<secure-deployment name="WAR MODULE NAME.war">
  <realm>michael-demo</realm>
  <auth-server-url>https://sso-sso.apps.acocp.rhcasalab.com/auth/</auth-server-url>
  <public-client>true</public-client>
  <ssl-required>EXTERNAL</ssl-required>
  <resource>demo-java</resource>
</secure-deployment>
```

Console client

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

The screenshot shows the 'RED HAT SINGLE SIGN-ON' console client interface. The top navigation bar includes the text 'RED HAT SINGLE SIGN-ON' on the left and 'Sign Out' on the right. A left-hand sidebar contains a menu with the following items: 'Account', 'Password', 'Authenticator', 'Sessions' (which is highlighted with a blue bar and a right-pointing arrow), and 'Applications'. The main content area is titled 'Sessions' and contains a table with the following data:

IP	Started	Last Access	Expires	Clients
192.168.2.100	May 4, 2020, 7:54:42 PM	May 4, 2020, 8:03:41 PM	May 5, 2020, 5:54:42 AM	demo-python account

Below the table is a button labeled 'Log out all sessions'.

Démonstration #3 – Ajout du identité social – github

Sur Github

Settings → Developer settings → Register a new OAuth application.

The screenshot shows the GitHub OAuth application settings for an application named 'RHSSO-APP'. On the left, there is a sidebar with three tabs: 'GitHub Apps', 'OAuth Apps' (which is selected), and 'Personal access tokens'. The main content area shows the application details:

- Application name:** RHSSO-APP
- Owner:** michaellessard (with a profile picture icon) owns this application. A 'Transfer ownership' button is visible.
- Marketplace:** A note states 'You can list your application in the GitHub Marketplace so that other users can discover it.' with a 'List this application in the Marketplace' button.
- Client ID:** 92afac629f9812ddc5f2
- Client Secret:** A masked secret with a 'Reset client secret' button.
- Buttons:** 'Revoke all user tokens' and 'Reset client secret'.
- Application logo:** A section with a 'Drag & drop' area and an 'Upload new logo' button. A note says 'You can also drag and drop a picture from your computer.'
- Application name:** A text input field containing 'RHSSO-APP'.
- Homepage URL:** A text input field containing 'https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-dem'.
- Application description:** A text area with the placeholder text 'Application description is optional'.
- Authorization callback URL:** A text input field containing 'https://sso-sso.apps.acocp.rhcasalab.com/auth/realms/michael-dem'.

At the bottom, there are two buttons: 'Update application' (green) and 'Delete application' (red).

Michael-demo

Identity Providers > GitHub

GitHub

Settings

Mappers

Redirect URI * Client ID * Client Secret Default Scopes Store Tokens OFFStored Tokens Readable OFFEnabled ONAccepts prompt=none forward from client OFFDisable User Info OFFTrust Email OFFAccount Linking Only OFFHide on Login Page OFFGUI order First Login Flow Post Login Flow

Configure

 Realm Settings Clients Client Scopes Roles Identity Providers User Federation Authentication

Manage

 Groups Users Sessions Events Import Export

Démonstration #4:: RH-SSO sur Openshift

The screenshot displays the Red Hat OpenShift Container Platform console interface. The left sidebar shows the navigation menu with 'Workloads' selected. The main content area shows the 'Deployment Config Details' for a Deployment Config named 'SSO' in the 'sso' namespace. A notification at the top indicates the user is logged in as a temporary administrative user. The 'SSO' Deployment Config is shown with 1 pod. The details include the following fields:

Field	Value
Name	sso
Latest Version	1
Namespace	sso
Message	config change
Update Strategy	Recreate
Min Ready Seconds	Not Configured
Triggers	ImageChange, ConfigChange
Pod Selector	deploymentConfig=sso
Node Selector	No selector

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](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat

EXTRA



RH-SSO 7.4 – Nouvelle console client – Tech preview

The screenshot displays the 'Device Activity' section of the RH-SSO 7.4 console. On the left is a navigation menu with options: Personal Info, Account Security, Signing In, Device Activity (highlighted), Linked Accounts, and Applications. The main content area is titled 'Device Activity' and includes a 'Sign Out All Devices' button. Below this, a section titled 'Signed In Devices' contains a list of three active sessions:

Device Information	Session Details
127.0.0.1 Current Session	Chrome/80.0.3987 / Windows 10 Last accessed on April 6, 2020, 6:21 AM Clients Account Console Started at April 6, 2020, 6:09 AM Expires at April 6, 2020, 4:09 PM
127.0.0.1	Firefox/74.0 / Windows 10 Last accessed on April 6, 2020, 6:21 AM Clients Security Admin Console, Account Console Started at April 6, 2020, 6:09 AM Expires at April 6, 2020, 4:09 PM Sign Out
192.168.1.102	Samsung Internet/11.1 / Android 8.0.0 Last accessed on April 6, 2020, 6:21 AM Clients Account Console Started at April 6, 2020, 6:21 AM Expires at April 6, 2020, 4:21 PM Sign Out

LOG

RED HAT SINGLE SIGN-ON Admin

Michael-demo

Events Config ?

Login Events Admin Events **Config**

Events Config

Event Listeners

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

Expiration Hours

Admin Events Settings

Save Events ON

Include Representation OFF

Clear admin events

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	<table border="1"> <tr><td>Client</td><td>demo-java</td></tr> <tr><td>User</td><td>f1ca5d91-19c2-4588-9e96-35fa6ccac575</td></tr> <tr><td>IP Address</td><td>192.168.2.100</td></tr> <tr><td>Error</td><td>invalid_user_credentials</td></tr> <tr><td>Details</td><td> <table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>michaellessard</td></tr> </table> </td></tr> </table>	Client	demo-java	User	f1ca5d91-19c2-4588-9e96-35fa6ccac575	IP Address	192.168.2.100	Error	invalid_user_credentials	Details	<table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>michaellessard</td></tr> </table>	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
Client	demo-java																					
User	f1ca5d91-19c2-4588-9e96-35fa6ccac575																					
IP Address	192.168.2.100																					
Error	invalid_user_credentials																					
Details	<table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>michaellessard</td></tr> </table>	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											
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	<table border="1"> <tr><td>Client</td><td>demo-java</td></tr> <tr><td>User</td><td></td></tr> <tr><td>IP Address</td><td>192.168.2.100</td></tr> <tr><td>Error</td><td>user_not_found</td></tr> <tr><td>Details</td><td> <table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>sdfdasfasdfsd</td></tr> </table> </td></tr> </table>	Client	demo-java	User		IP Address	192.168.2.100	Error	user_not_found	Details	<table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>sdfdasfasdfsd</td></tr> </table>	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
Client	demo-java																					
User																						
IP Address	192.168.2.100																					
Error	user_not_found																					
Details	<table border="1"> <tr><td>auth_method</td><td>openid-connect</td></tr> <tr><td>auth_type</td><td>code</td></tr> <tr><td>redirect_uri</td><td>http://127.0.0.1:8080/vanilla/profile.jsp</td></tr> <tr><td>code_id</td><td>19861aa6-291a-44c2-bfd8-f3918ab5f8bd</td></tr> <tr><td>username</td><td>sdfdasfasdfsd</td></tr> </table>	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											
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																					

Navigation icons: back, forward, refresh

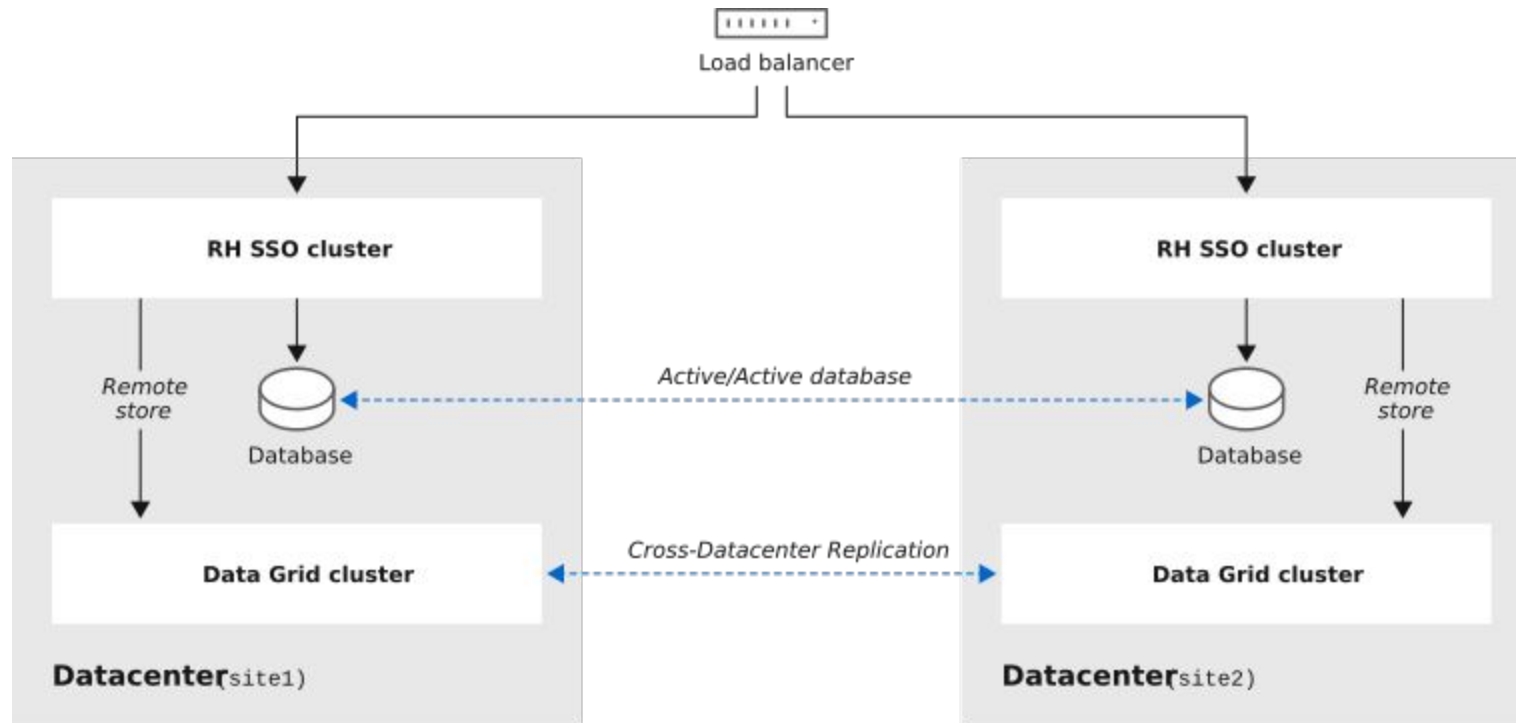
Forget password / remember me

The screenshot displays the Red Hat Single Sign-On administration interface. At the top, the header reads "RED HAT SINGLE SIGN-ON" and "Admin". The left sidebar shows the navigation menu with "Configure" selected, and "Realm Settings" highlighted. The main content area is titled "Michael-demo" and contains several tabs: "General", "Login", "Keys", "Email", "Themes", "Cache", "Tokens", "Client Registration", and "Security Defenses". The "Login" tab is active, showing the following settings:

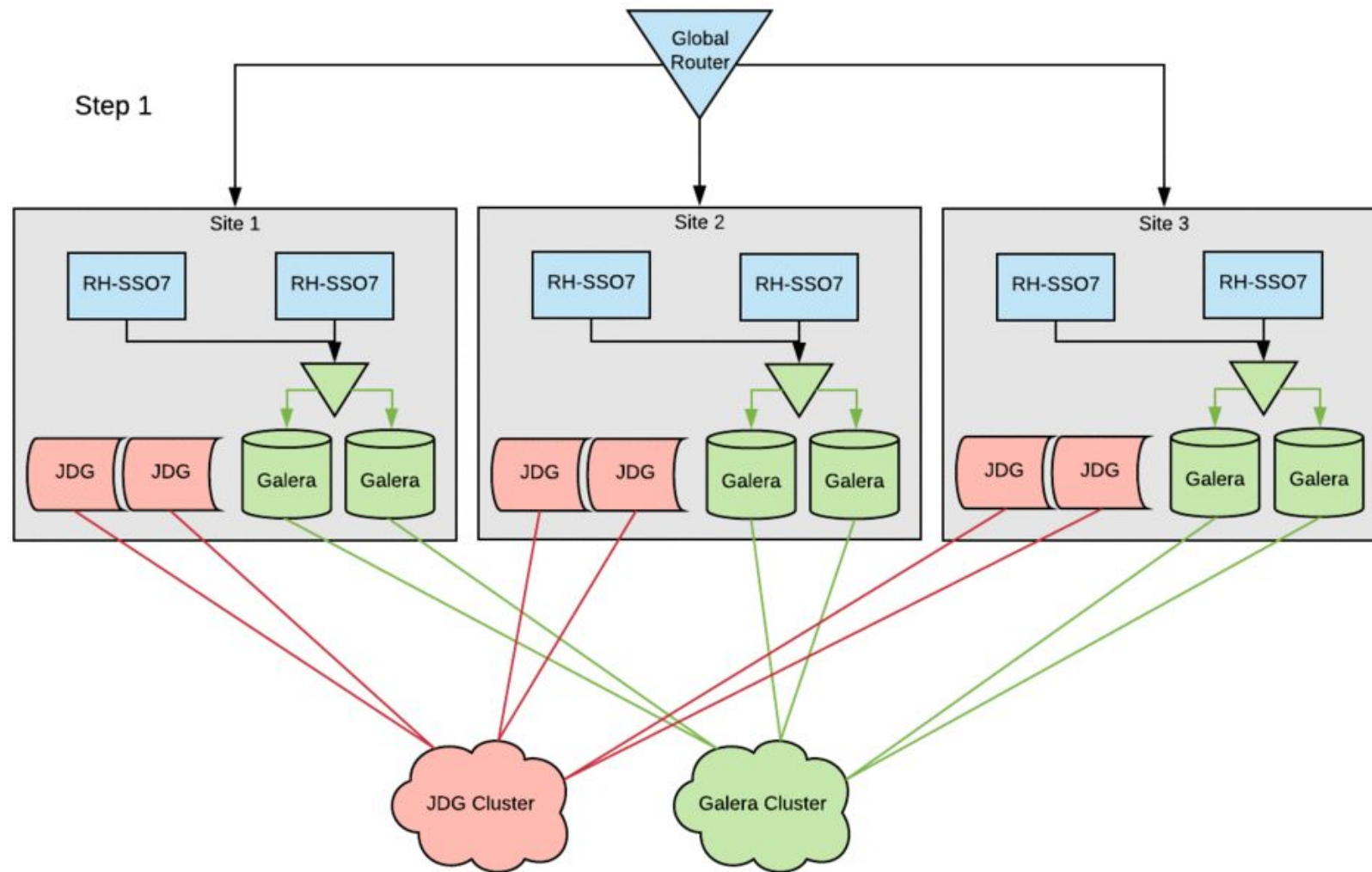
- User registration: OFF
- Edit username: OFF
- Forgot password: ON
- Remember Me: ON
- Verify email: OFF
- Login with email: ON
- Require SSL: external request!

At the bottom of the settings, there are "Save" and "Cancel" buttons.

RH-SSO architecture Multi-site



70_RHSSO_0020



<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

Exemple : 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)

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

Client ID * ⓘ

Client Protocol ⓘ

Root URL ⓘ

Michael-demo

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"
  }
}
```