Authentification SSO via SAML entre Guacamole et Keycloak

Cette solution de sécurisation de Guacamole VPN est très fiable, que ce soit pour un usage en local ou lorsqu'on expose Guacamole Daudruy sur Internet. Keycloak gère les utilisateurs et leur authentification, offrant ainsi une gestion centralisée et sécurisée des accès.

1. Préparer l'environnement

- Serveur Keycloak : sur Ubuntu

Téléchargement Keycloak :

Dernier version

```
zafar@auth-srv:/opt$ sudo wget https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-26.1.0.tar.gz
--2025-01-31 09:09:40--  https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-26.1.0.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/11125589/ff752aee-0a36-473d-9168-
7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250131%2Fus-east-1%2Fs3%2Faws4
_request&X-Amz-Date=20250131T090931Z&X-Amz-Expires=300&X-Amz-Signature=ac5346f53a8f90dd419b7cf3e1b8701be73a221e61aada89a
98dd053fede2b57&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkeycloak-26.1.0.tar.gz&
response-content-type=application%2Foctet-stream [following]
--2025-01-31 09:09:40--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/11125589/ff752aee-
0a36-473d-9168-7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250131%2Fus-eas
t-1%2Fs3%2Faws4_request&X-Amz-Date=20250131T090931Z&X-Amz-Expires=300&X-Amz-Signature=ac5346f53a8f90dd419b7cf3e1b8701be7
3a221e61aada89a98dd053fede2b57&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkeycloak
-26.1.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.108
.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 148227819 (141M) [application/octet-stream]
Saving to: ‘keycloak-26.1.0.tar.gz’

keycloak-26.1.0.tar.gz    100%[===================================>] 141,36M  93,2MB/s    in 1,5s
```

Installation de java :

Keycloak est basé sur Quarkus, qui fonctionne sur la JVM (Java Virtual Machine). Il a besoin de Java 17 ou 21 pour s'exécuter

```
zafar@auth-srv:/opt$ sudo apt install -y openjdk-21-jre
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libllvm17t64
```

décompression des fichier tar keycloack

```
zafar@auth-srv:/opt$ sudo tar -xzvf keycloak-26.1.0.tar.gz
keycloak-26.1.0/version.txt
keycloak-26.1.0/conf/cache-ispn.xml
keycloak-26.1.0/README.md
keycloak-26.1.0/themes/README.md
```

```
zafar@auth-srv:/opt$ ls
keycloak-26.1.0  keycloak-26.1.0.tar.gz
zafar@auth-srv:/opt$
```

Les dossier de configuration et exécution :

```
zafar@auth-srv:/opt/keycloak-26.1.0$ tree bin/
bin/
├── client
│   ├── keycloak-admin-cli-26.1.0.jar
│   └── lib
│       ├── bcprov-jdk18on-1.78.1.jar
│       ├── keycloak-crypto-default-26.1.0.jar
│       └── keycloak-crypto-fips1402-26.1.0.jar
├── federation-sssd-setup.sh
├── kcadm.bat
├── kcadm.sh
├── kc.bat
├── kcreg.bat
├── kcreg.sh
└── kc.sh

3 directories, 11 files
zafar@auth-srv:/opt/keycloak-26.1.0$
```

```
root@auth-srv:/opt/keycloak# cd keycloak-26.1.0/
root@auth-srv:/opt/keycloak/keycloak-26.1.0# ls
bin  conf  lib  LICENSE.txt  providers  README.md  themes  version.txt
root@auth-srv:/opt/keycloak/keycloak-26.1.0# cd bin/
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls
client  federation-sssd-setup.sh  kcadm.bat  kcadm.sh  kc.bat  kcreg.bat  kcreg.sh  kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls -l ./kc.sh
-rwxr-xr-x 1 1001 118 6286 janv. 15 10:25 ./kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# chmod +x kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
```

Maintenant qu' on installé il faut créer un user Admin pour se connecter à l' interface Keycloack :

Premiere solution :

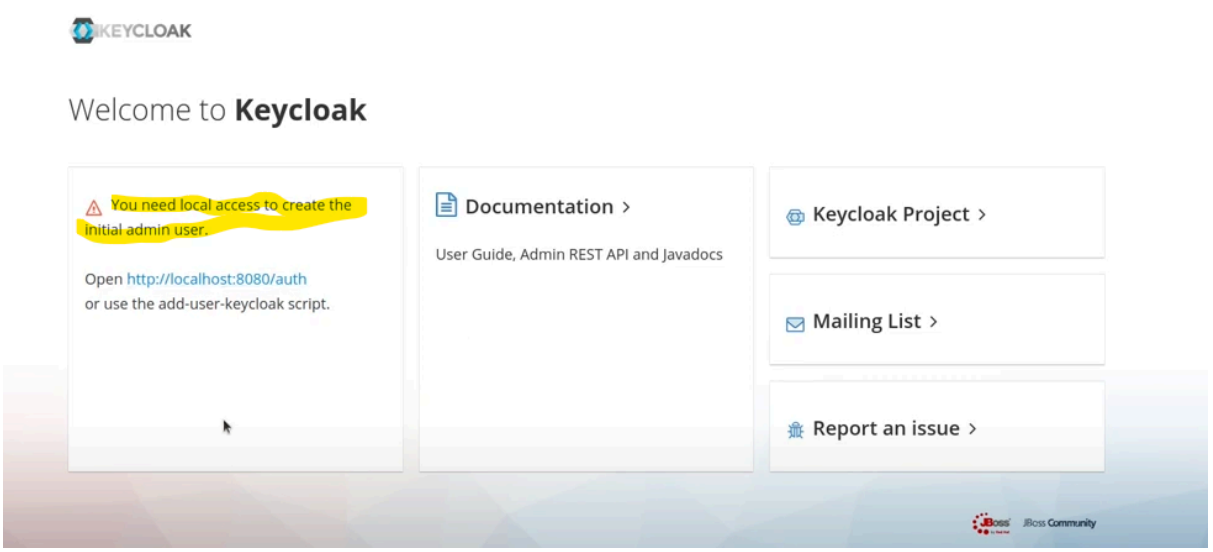
```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ export KEYCLOAK_ADMIN=admin
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ export KEYCLOAK_ADMIN_PASSWORD=admin
```

Puis on lance le scripte de keycloack :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev
Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-31 09:16:48,065 WARN  [io.qua.config] (build-9) Unrecognized configuration key "quarkus.smallrye-health.extensio
ns.enabled" was provided; it will be ignored; verify that the dependency extension for this configuration is set or that
 you did not make a typo
2025-01-31 09:16:49,629 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-13) Persistence unit 'keycloak-default':
 Enforcing Quarkus defaults for dialect 'org.hibernate.dialect.H2Dialect' by automatically setting 'jakarta.persistence.
database-product-version=2.3.230'.
2025-01-31 09:16:49,632 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-13) A legacy persistence.xml file is pre
```

Puis on tapes sur interface l' ip machine et le port :

http://10.10.10.x:80xx



Si cette méthode ne fonctionne pas on vas forcer Keycloack à configuier le user :

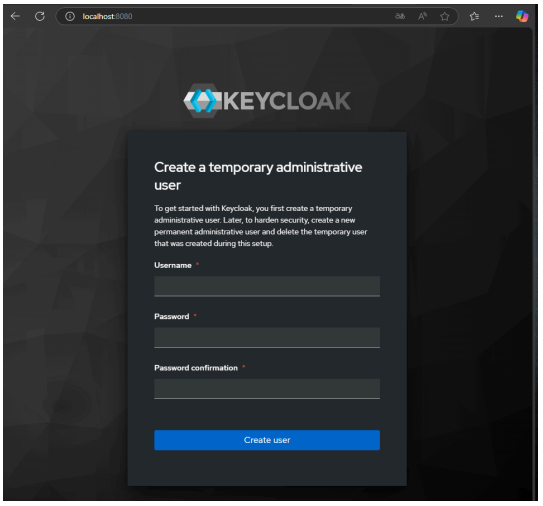
```
 GNU nano 7.2                          /opt/keycloak-26.1.0/conf/keycloak.conf
#metrics-enabled=true

# HTTP

# The file path to a server certificate or certificate chain in PEM format.
#https-certificate-file=${kc.home.dir}conf/server.crt.pem

# The file path to a private key in PEM format.
#https-certificate-key-file=${kc.home.dir}conf/server.key.pem

# The proxy address forwarding mode if the server is behind a reverse proxy.
#proxy=reencrypt

# Do not attach route to cookies and rely on the session affinity capabilities from reverse proxy
#spi-sticky-session-encoder-infinispan-should-attach-route=false

# Hostname for the Keycloak server.
#hostname=myhostname

http-enabled=true
http-host=0.0.0.0
hostname=10.10.10.8
http-port=8080
admin=admin
admin-password=admin
```

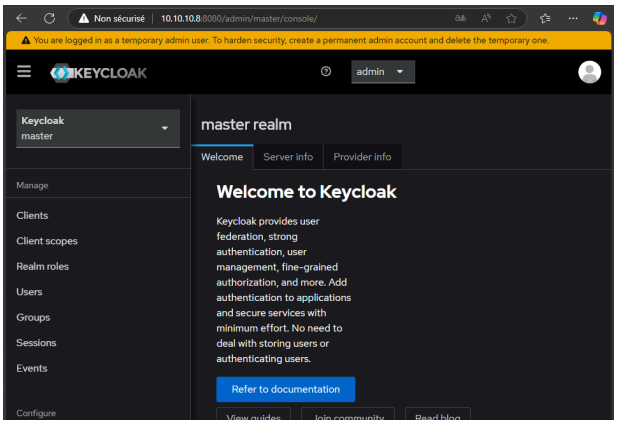
```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh build
WARNING: The following run time options were found, but will be ignored

Updating the configuration and installing your custom providers, if any
```

Puis on exécute à nouveau le script sur le serveur

```
zafar@auth-srv:/opt/keycloak-26.1.0$ sudo ./bin/kc.sh start-dev
Running the server in development mode. DO NOT use this configuration in production.
2025-01-31 10:06:29,469 WARN  [io.quarkus.config] (main) Unrecognized configuration key "quarkus.small
rye-health.extensions.enabled" was provided; it will be ignored; verify that the dependency extension
for this configuration is set or that you did not make a typo
2025-01-31 10:06:29,903 INFO  [org.keycloak.url.HostnameV2ProviderFactory] (main) If hostname is speci
fied, hostname-strict is effectively ignored
2025-01-31 10:06:31,763 INFO  [org.keycloak.quarkus.runtime.storage.infinispan.CacheManagerFactory] (T
hread-5) Starting Infinispan embedded cache manager
2025-01-31 10:06:31,845 INFO  [io.agroal.pool] (JPA Startup Thread) Datasource '<default>': Initial si
ze smaller than min. Connections will be created when necessary
2025-01-31 10:06:31,937 INFO  [org.infinispan.CONTAINER] (Thread-5) Virtual threads support enabled
2025-01-31 10:06:32,245 INFO  [org.infinispan.CONTAINER] (Thread-5) ISPN000556: Starting user marshall
```

On vérifie sur interface avec ip et port





Pour le moment le firewall est désactivé :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ufw status
Status: inactive
zafar@auth-srv:/opt/keycloak-26.1.0/bin$
```

Par défaut, on doit exécuter la commande suivante pour démarrer Keycloak à chaque fois que tu veux accéder à l'interface : /opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev

Dans le cas ou on veut automatiser le démarrage

1 Créer un service systemd pour Keycloak

1. Ouvre un fichier de service :

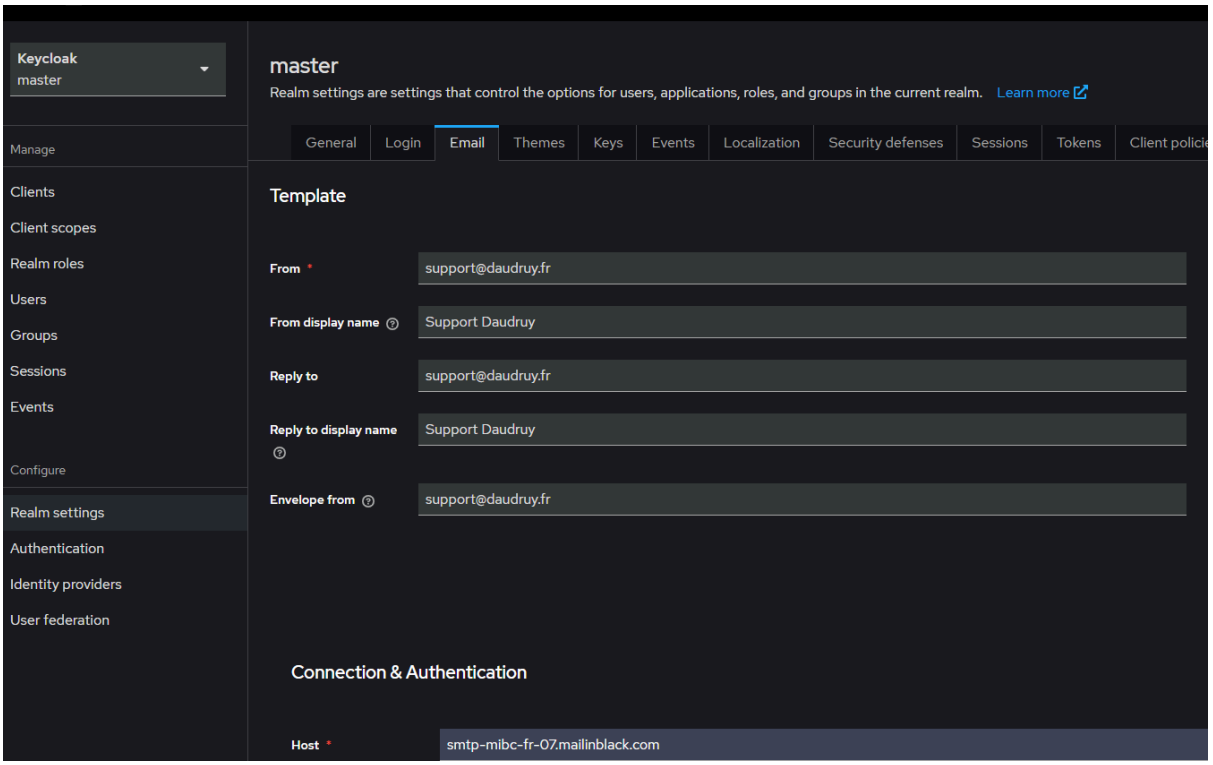
bash — Copier — Modifier

```
sudo nano /etc/systemd/system/keycloak.service
```

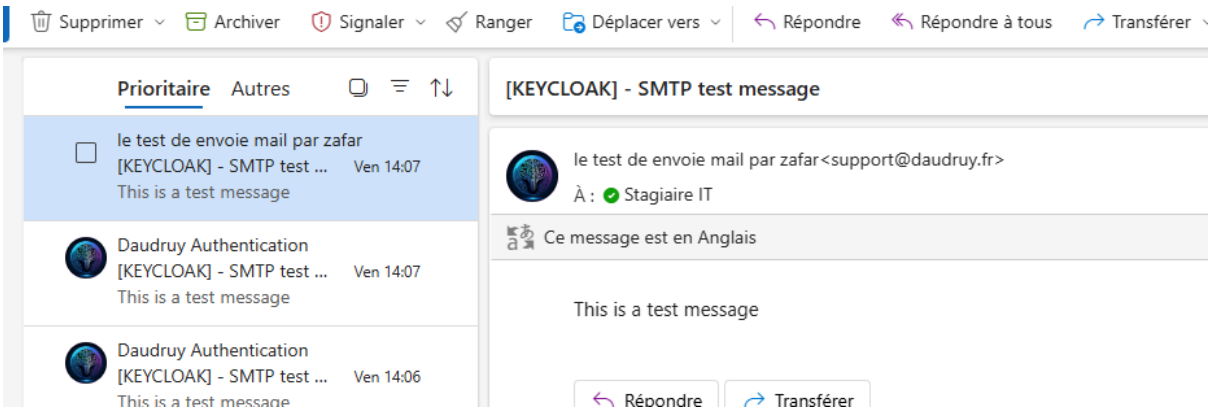
2. Ajoute cette configuration :

ini — Copier — Modifier

```
[Unit]
Description=Keycloak Server
After=network.target

[Service]
User=root
WorkingDirectory=/opt/keycloak
ExecStart=/opt/keycloak/bin/kc.sh start-dev
Restart=always
StandardOutput=journal
StandardError=journal
LimitNOFILE=1024

[Install]
WantedBy=multi-user.target
```

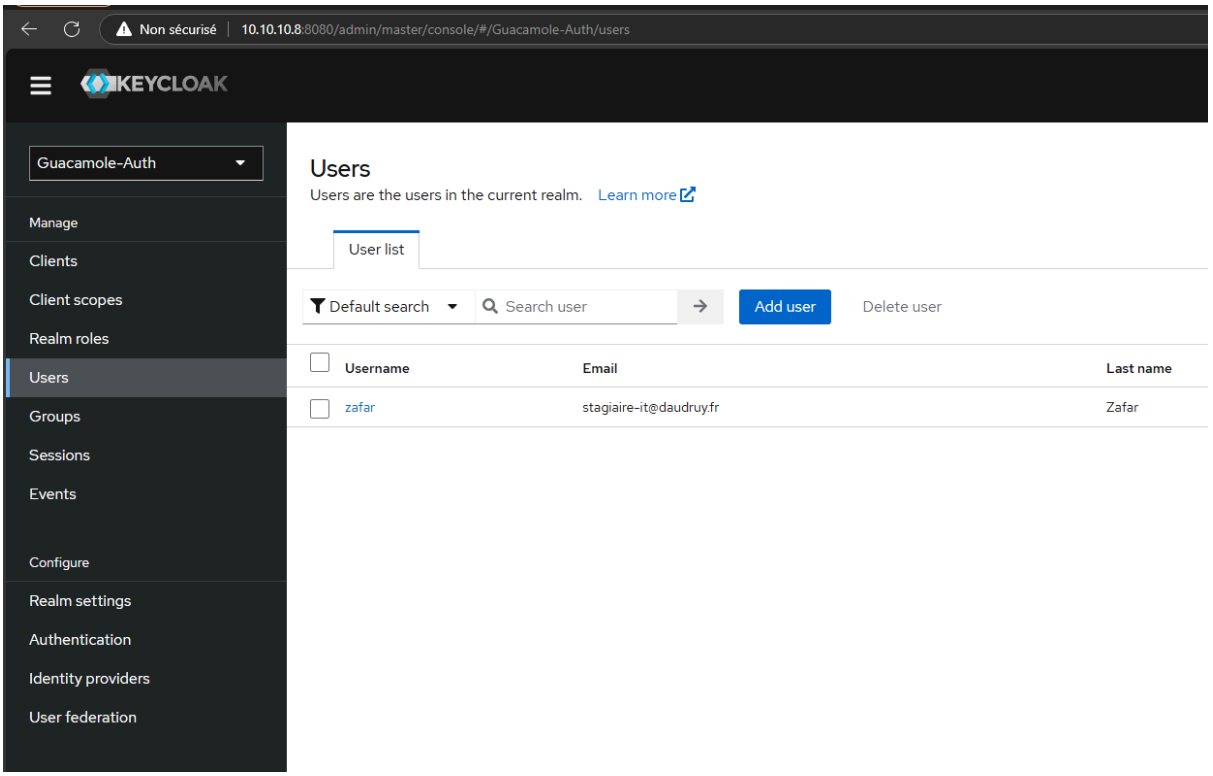


Test envoi mail :



Mode recovery si jamis on a un sousci de connexion

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ cd /opt/keycloak-26.1.0/bin
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev --spi-authenticator-required-action-verify-email-enabled=false
Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-31 11:35:07,451 WARN  [io.qua.config] (build-40) Unrecognized configuration key "quarkus.smallrye-health.extensions.enabled" was provide
his configuration is set or that you did not make a typo
2025-01-31 11:35:09,089 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-19) Persistence unit 'keycloak-default': Enforcing Quarkus defau
y setting 'jakarta.persistence.database-product-version=2.3.230'.
2025-01-31 11:35:09,091 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-19) A legacy persistence.xml file is present in the classpath. T
units, and any configuration of the Hibernate ORM extension will be ignored. To ignore persistence.xml files instead, set the configuration prop
2025-01-31 11:35:12,892 INFO  [io.qua.dep.QuarkusAugmentor] (main) Quarkus augmentation completed in 6679ms
Running the server in development mode. DO NOT use this configuration in production.
```



Compte rendu

Keycloak n’était pas une solution adaptée dans ce contexte, car il aurait nécessité une infrastructure plus lourde à mettre en place et à maintenir, avec une gestion des identités centralisée qui dépasse le cadre du besoin initial.

En effet, avec Keycloak, il faudrait créer un utilisateur à chaque fois sur Keycloak, puis créer manuellement le même utilisateur sur Guacamole, tout en effectuant des configurations supplémentaires. Cela ajouterait des tâches supplémentaires aux équipes IT, ce qui n’est pas souhaitable. L'équipe préfère une solution simple et facile à gérer, sans complexité inutile car ils ont l'habitude de sous-traiter la plupart de leurs services à des prestataires externes et, en cas de problème, ils créent des tickets pour obtenir une assistance.

Cependant, cette recherche, mise en place et configuration m’ont permis d’approfondir ma compréhension des protocoles d’authentification ainsi que de la configuration de Keycloak, en explorant son intégration avec d’autres services comme ldap , Microsoft authentication , authentification par AD etc et ses mécanismes de gestion des identités.