installer et configurer Apereo/CAS 6.3.2 sur debian 10 (Buster)

Configurer la Debian

Ajouter les backports dans le dépôt

```
echo "deb http://deb.debian.org/debian buster-backports main contrib non-free" >> /etc/apt/sources.list
```

Mettre à jour le système

```
#apt update
```

```
#apt upgrade
```

Installation des paquets nécessaire

```
#apt install tomcat9 tomcat9-admin tomcat9-user openjdk-11-jdk openjdk-11-jre maven build-essential git
```

Configurer la variable d'environnement JAVA

```
#echo "JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/" >> /etc/environment
```

```
#source /etc/environment
```

Vérifier la variable d'environnement

```
echo $JAVA_HOME
```

Configurer Tomcat9

Aller dans /etc/default
Ouvrir le **fichier tomcat9**
Rajouter la ligne suivante :

```
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

Vérifier de nouveau si vous avez Java 11

```
update-alternatives --display java
```

Configurer le Tomcat9 manager

Aller dans /etc/tomcat9
Ouvrir le fichier **tomcat-users.xml**
Tout en bas du fichier mettre

```
<role rolename="admin-gui"/>
<user username="admin" password="toor" roles="manager-gui,admin-gui"/>
```

Redémarrer Tomcat

```
systemctl restart tomcat9
```

Mettre TOMCAT9 en HTTPS.

Vérifier que le port 80 et 443 sort bien vers l'extérieur.

Maintenant nous allons utiliser Let's Encrypt

Pour installer cerbot via les paquet

```
apt install python-certbot-apache
```

Utiliser la commande suivante pour avoir le domaine en https :

```
certbot --apache -d cas.domaine-univ.fr
```

il vous demande de mettre votre adresse mail.

Suivre les indications.

Récupérer le chemin ou se trouve les fichier pem qui vous donne let's encrypte

Il vous donne le chemin ou se trouve les fichier pem.

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/cas.domaine-univ.fr/fullchain.pem

   Your key file has been saved at:
/etc/letsencrypt/live/cas.domaine-univ.fr/privkey.pem

- Your cert will expire on 2021-06-31. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option.
  To non-interactively renew *all* of your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot configuration directory at **/etc/letsencrypt**. You should make a secure backup of this folder now.
  This configuration directory will also contain certificates and private keys obtained by Certbot so making regular backups of this folder is ideal.

Créer le dossier pour les fichier SSL

J'ai créé un dossier dans **/opt/tomcat/conf**

```
Mkdir /opt/tomcat/conf
```

```
Mkdir /opt/tomcat
```

```
Mkdir /opt/tomcat/conf
```

J'ai copié les trois fichiers dans **/opt/tomcat/conf** :

```
cp cert.pem chain.pem privkey.pem /opt/tomcat/conf
```

J'ai mi les droits pour les fichier *.pem :

```
chown tomcat:tomcat *.pem
```

Ajout des ligne SSL sur le serveur Tomcat9

Je suis allé dans /etc/tomcat9/server.xml

J'ai ajouté les lignes suivantes :

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true">
  <SSLHostConfig>
    <Certificate certificateFile="/opt/tomcat/conf/cert.pem"
                 certificateKeyFile="/opt/tomcat/conf/privkey.pem"
                 certificateChainFile="/opt/tomcat/conf/chain.pem" />
  </SSLHostConfig>
</Connector>
```

J'ai redémarré tomcat

Pré-installe Apereo / CAS

Nous commençons par installer le projet nécessaire à l'utilisation de cas-overlay-Template
Récupérer le projet
Aller dans le dossier /opt

```
#git clone https://github.com/apereo/cas-overlay-template
```

Ouvrir le dossier cas-overlay-template

```
#cd cas-overlay-template
```

Ajout de module LDAP (Dépendance)

Ouvrir le **fichier build.gradle**
Ajouter les 3 lignes en gras ci-dessous :

```
dependencies {
    // Add modules in format compatible with overlay casModules property
    if (project.hasProperty("casModules")) {
        def dependencies = project.getProperty("casModules").split(",")
        dependencies.each {
            def projectsToAdd = rootProject.subprojects.findAll {project ->
                project.name == "cas-server-core-${it}" || project.name == "cas-server-support-${it}"
            }
            projectsToAdd.each {implementation it}
        }
    }
    // CAS dependencies/modules may be listed here statically...


implementation "org.apereo.cas:cas-server-webapp-init:${casServerVersion}"
implementation "org.apereo.cas:cas-server-support-ldap:${project.'cas.version'}"
implementation "org.apereo.cas:cas-server-support-json-service-registry:${casServerVersion}"

}
```

Enregistrer le fichier

Ajouter dans le **fichier cas.properties** la configuration LDAP

Allller dans /opt/cas-overlay-template/etc/cas/config/cas.properties

```
cas.server.name=https://cas.domaine-univ.fr:8443
cas.server.prefix=${cas.server.name}/cas
logging.config: file:/etc/cas/config/log4j2.xml
```

ATTENTION ENLEVER LE # POUR ÊTRE EN VERSION PRODUCTION
 # cas.authn.accept.users=

```
### Desactivation des comptes locaux

cas.authn.accept.users=
### Connexion LDAP
cas.authn.ldap\[0\].providerClass=org.ldaptive.provider.unboundid.UnboundIDProvider
cas.authn.ldap\[0\].type=AUTHENTICATED
cas.authn.ldap\[0\].useSsl=false
cas.authn.ldap\[0\].ldapUrl=ldap://domaine-univ.fr:389
cas.authn.ldap\[0\].baseDn= dc= domaine-univ,dc=fr
cas.authn.ldap\[0\].subtreeSearch=true
cas.authn.ldap\[0\].searchFilter=sAMAccountName=\{user\}
cas.authn.ldap\[0\].principalAttributeList=cn,givenName,mail

### Credential to connect to LDAP
cas.authn.ldap\[0\].bindDn=CN=Admincas,CN=CasAdmin,DC= domaine-univ,DC=fr
cas.authn.ldap\[0\].bindCredential=P@ssW0rd
```

Activation en LDAPS

Par contre si vous voulez metre le LDAP en LDAPS

Il vous faudra activer userSsl=true

```
cas.authn.ldap\[0\].useSsl=false
```

Vérification du port de connexion LDAP 389

```
telnet domaine-univ.fr 389
```

Si vous avez ce message-là :
Trying 192.168.0.54...
Connected to domaine-univ.fr.
Escape character is '^]'.
C'est ok.

Vérification du port de connexion LDAP 636

```
telnet domaine-univ.fr 636
```

Si vous avez ce message-là :
Trying 192.168.0.54...
Connected to domaine-univ.fr.
Escape character is '^]'.
C'est ok.

Création du dossier log pour CAS

Créer un dossier dans /var/log/cas

```
mkdir /var/log/cas
```

Copie des fichiers cas.log et cas_audit.log

Copier ou créer les fichiers cas.log et cas_audit.log dans le dossier

Droit sur le dossier

Pour mettre les droits sur le dossier faire la commande suivante :

```
chown -R tomcat:adm /var/log/cas
```

Modifier le fichier `log4j2.xml` dans le dossier cas-overlay-template-master/etc/cas/config

Mettre à la place de

```
<Property name="baseDir">/var/log</Property>
```

```
<Property name="baseDir">/var/log/cas</Property>
```

Enregistrer le fichier

Installation de Gradle

```
./gradlew clean
```

#.

```
./gradlew clean copyCasConfiguration build
```

Création de la clé
#

```
./gradlew createKeystore
```

Il faut récupérer le fichier

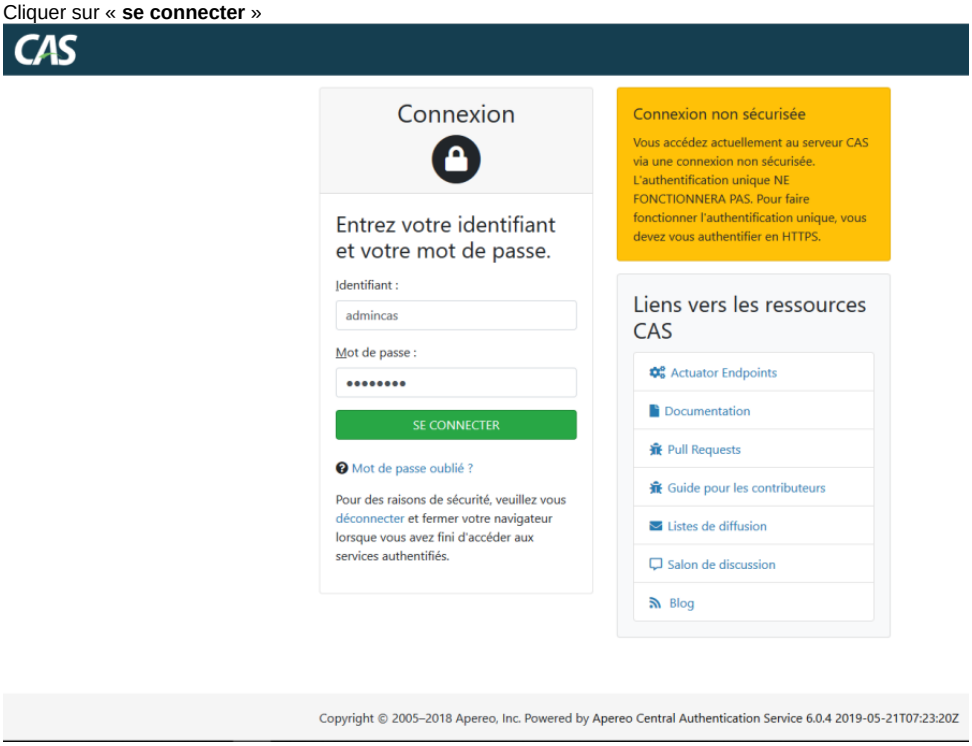
Copier le fichier cas.war

```
cp /opt/cas-overlay-template/build/libs/cas.war /var/lib/tomcat9/webapps/
```

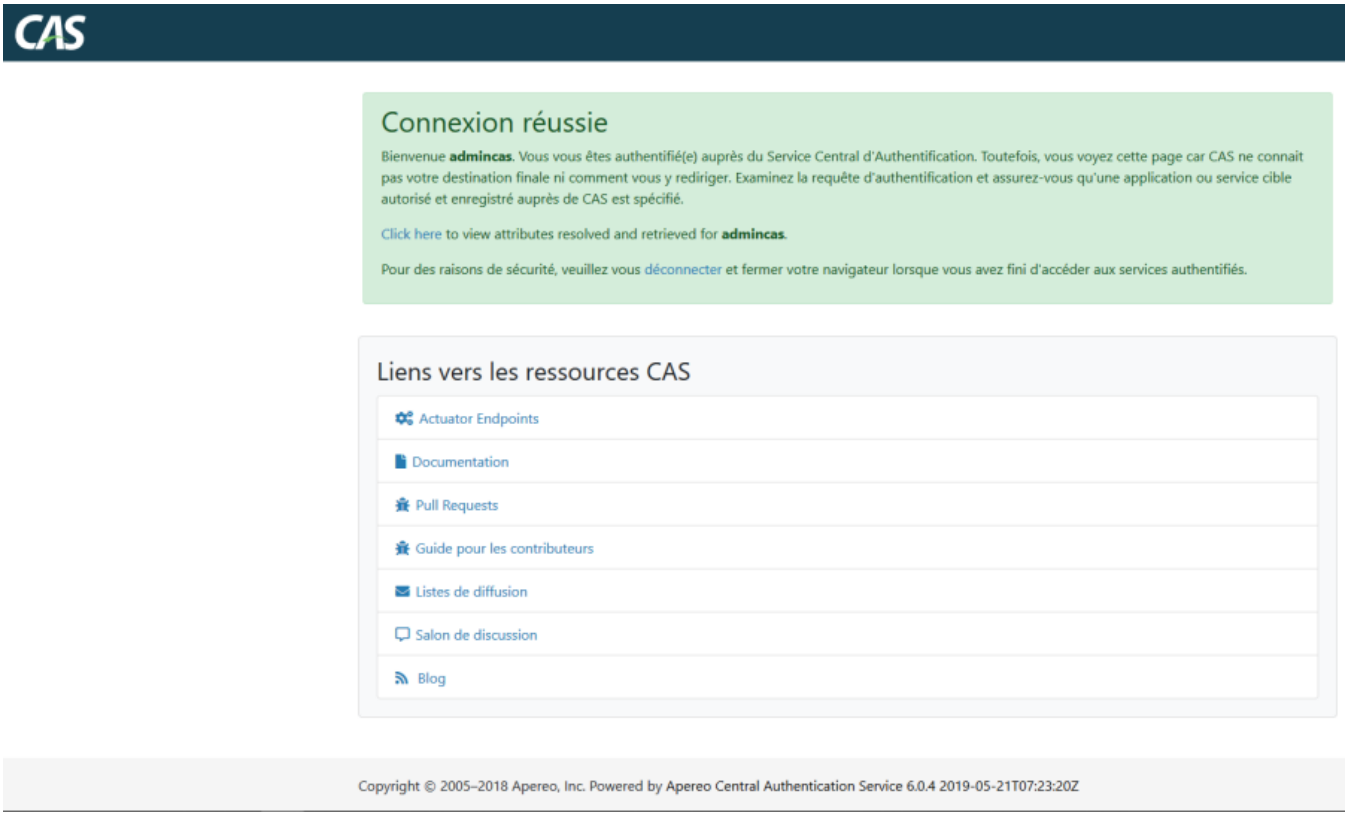
Relancer le service de Tomcat9

```
systemctl restart tomcat9.service
```

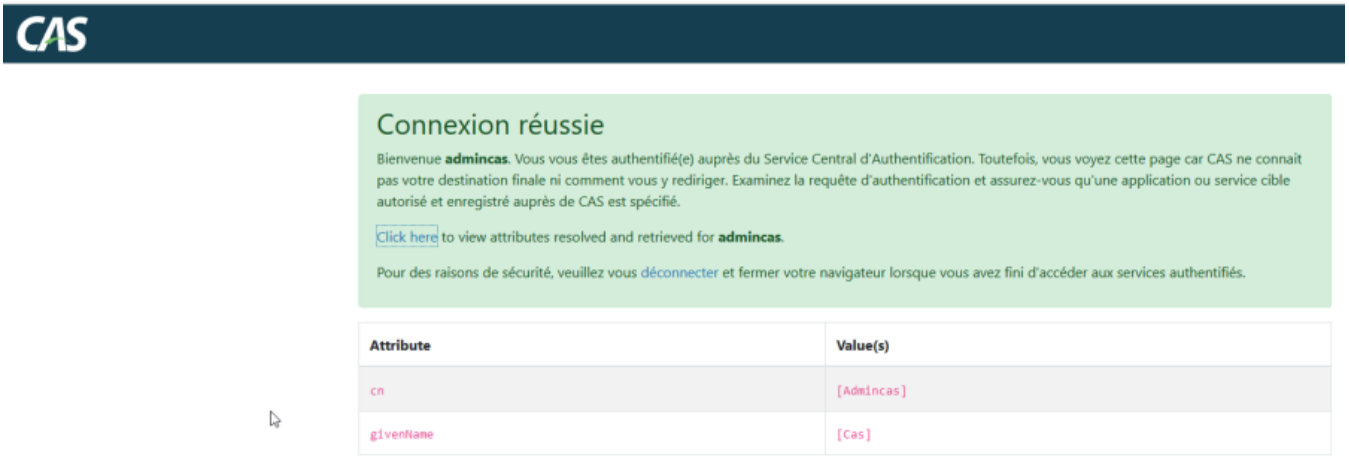
Maintenant nous allons tester la connexion
Pour ici l'adresse ip est : https://cas.domaine-univ.fr:8443/cas
Cliquer sur « **se connecter** »



Quand vous êtes connectés, vous devez avoir ce message-là : **Connexion réussie**



En cliquant sur



Nous pouvons voir les attributs de l'active Directory.

Choix de la Configuration des applications

Nous avons le choix pour configurer les applications soit avec :

- Avec Json
- Base de donnée module JPA

Pour démarrer nous allons utiliser le Json.

Configuration du Json

Vérifier que dans le fichier **build.gradle** nous avons bien :

```
implementation "org.apereo.cas:cas-server-support-json-service-registry:${project.'cas.version'}"
```

Une fois vérifier nous pouvons commencer.

Configurer le fichier cas.properties

Ouvrir le fichier cas. properties

```
cd \opt\cas-template-overlay
```

```
nano etc/cas/config/cas.properties
```

Ajouter la ligne ci-dessous :

Configuration de JSON

```
cas.serviceRegistry.json.location: file:/etc/cas/services
```

Une fois que la ligne a été ajouté nous devons créer un dossier.

Création du dossier services

Nous utilisons la commande mkdir pour créer le dossier

```
mkdir /etc/cas/services
```

Ajouter les applications

Nous allons créer pour chaque applications un fichier en « .json »
Il est recommandé de nommer les nouveaux fichiers JSON comme suit:

```
serviceName-serviceNumericId.json"
```

Pour créer l'ID nous utilisons la commande suivant :

```
date +%s
```

Voici ce qui donne

```
root@cas:/opt/cas-overlay-template# date +%s
1559915619
```

Ce numéro est notre ID.
Donc le fichier sera « application-1559915619.json »
Le fichier doit être dans /etc/cas/services/

Création du fichier json

« Application » est le nom de votre application dans cette exemple

```
touche application-1559915619.json
```

Configuration des applications

Ouvrir le fichier que nous venons de créer

```
nano application-1559915619.json
```

Attention l'exemple ci-dessous à éviter à l'utilisation

```
{
/*
 Ne pas utiliser cette définition dans un environnement de production.
    */
    "@class" : "org.apereo.cas.services.RegexRegisteredService",
    "serviceId" : "^(https|imaps)://.*",
    "name" : "HTTPS and IMAPS wildcard",
    "id" : 1503925297,
    "evaluationOrder" : 99999
    }
```

Voici pour l'application Rocketchat

Ouvrir le fichier http_rocketchat-1559902436.json

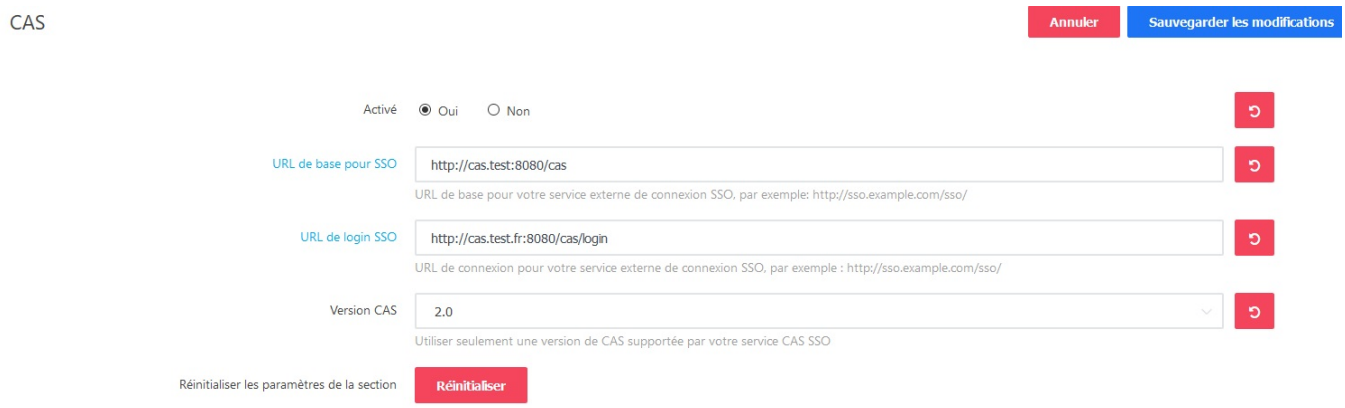
```
nano http_rocketchat-15599902436
```

```
{
"@class" : "org.apereo.cas.services.RegexRegisteredService",
"serviceId" : "^(http)://192.168.0.111/_cas/.*",
"name" : "Rocketchat",
"id" : 1503925297,
"evaluationOrder" : 99999
}
```

Aller dans RocketChat mettre les informations du serveur cas.

URL de base pour SSO : https://cas.domaine-univ.fr:8443/cas

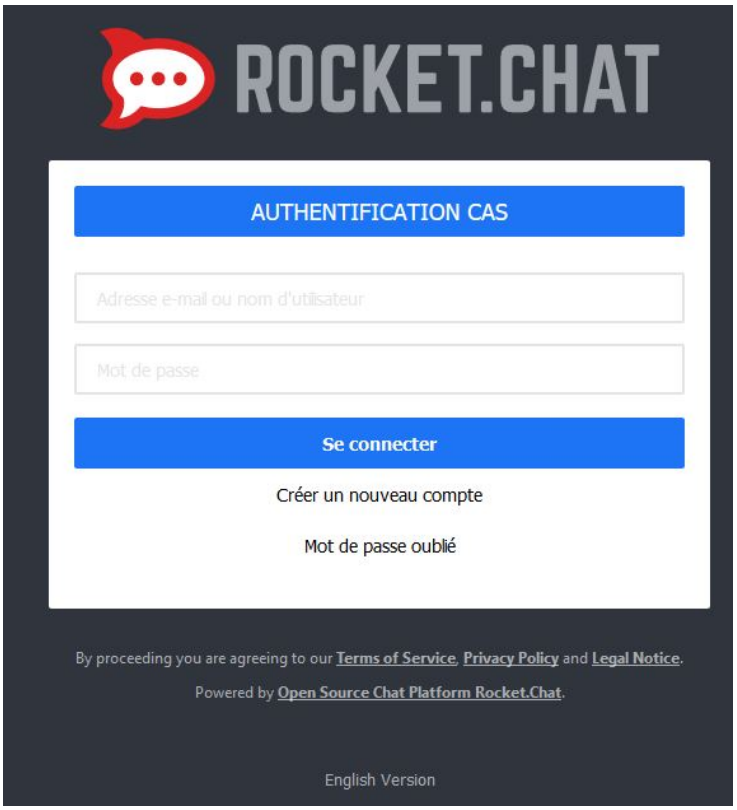
URL de login SSO : https://cas.domaine-univ.fr:8443/cas/login



Vous devez mettre vos attributs

Ouvrir la page de RocketChat pour ici c'est 192.168.0.113:3000

Cliquer sur AUTHENTIFICATION CAS



La page s'ouvre et nous voyons que RocketChat est bien dans CAS

Mettre son identifiant et son mot de passe du serveur AD

CAS

Connexion

Entrez votre identifiant et votre mot de passe.

Identifiant :

Mot de passe :

SE CONNECTER

Mot de passe oublié ?

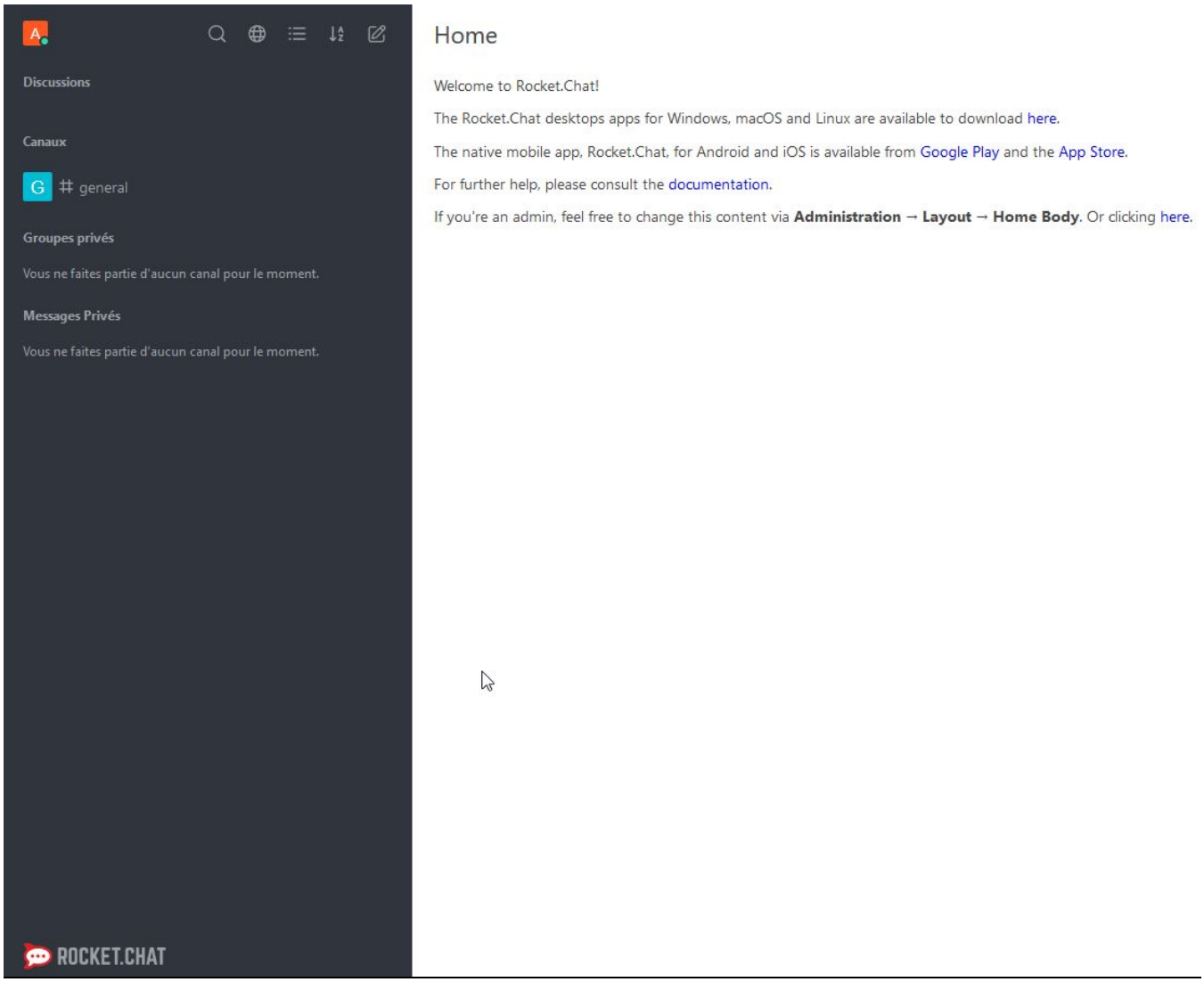
Pour des raisons de sécurité, veuillez vous déconnecter et fermer votre navigateur lorsque vous avez fini d'accéder aux services authentifiés.

RocketChat

RocketChat

Liens vers les ressources CAS

- Actuator Endpoints
- Documentation
- Pull Requests
- Guide pour les contributeurs
- Listes de diffusion
- Salon de discussion
- Blog


Discussions
Canaux
general
Groupes privés
Vous ne faites partie d'aucun canal pour le moment.
Messages Privés
Vous ne faites partie d'aucun canal pour le moment.
ROCKET.CHAT
Home
Welcome to Rocket.Chat!
The Rocket.Chat desktops apps for Windows, macOS and Linux are available to download here.
The native mobile app, Rocket.Chat, for Android and iOS is available from Google Play and the App Store.
For further help, please consult the documentation.
If you're an admin, feel free to change this content via Administration → Layout → Home Body. Or clicking here.