

installer et configurer Apereo/CAS 6.0.4 sur debian 9

Configurer la Debian

Ajouter les backports dans le dépôt

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

Mettre à jour le système

```
#apt update
```

```
#apt upgrade
```

Installation des paquets nécessaire

```
#apt install tomcat8 tomcat8-admin tomcat8-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
```

Suppression de JAVA 8
Faire un

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

Si vous voyez lien secondaire java.1.gz : /usr/lib/jvm/java-8-openjdk-amd64/jre/man/man1/java.1.gz
Supprimer le.

```
apt remove openjdk-8-jre-headless --purge
```

Configurer Tomcat8

Aller dans /etc/default
Ouvrir le **fichier tomcat8**
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 Tomcat8 manager

Aller dans /etc/tomcat8
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 tomcat8
```

Pré-installe Apereo / CAS

On commence 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 la ligne ci-dessous :

```
dependencies {  
    compile "org.apereo.cas:cas-server-webapp:${project.appServer}:${casServerVersion}"  
    // Other CAS dependencies/modules may be listed here...  
    compile "org.apereo.cas:cas-server-support-ldap:${project.'cas.version'}"  
    compile "org.apereo.cas:cas-server-support-json-service-registry:${casServerVersion}"  
}
```

Enregistrer le fichier

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

```
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
```

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.

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 tomcat8: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/tomcat8/webapps/
```

Relancer le service de Tomcat8

```
systemctl restart tomcat8.service
```

Maintenant nous allons tester la connexion
Pour ici l'adresse ip est : [*http://192.168.0.104:8080/cas*](http://192.168.0.104:8080/cas)
Cliquer sur « **se connecter** »

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.

Connexion non sécurisée

Vous accédez actuellement au serveur CAS via une connexion non sécurisée. L'authentification unique NE FONCTIONNERA PAS. Pour faire fonctionner l'authentification unique, vous devez vous authentifier en HTTPS.

Liens vers les ressources CAS

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

Copyright © 2005–2018 Apereo, Inc. Powered by Apereo Central Authentication Service 6.0.4 2019-05-21T07:23:20Z

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

CAS

Connexion réussie

Bienvenue **admincas**. Vous vous êtes authentifié(e) auprès du Service Central d'Authentification. Toutefois, vous voyez cette page car CAS ne connaît pas votre destination finale ni comment vous y rediriger. Examinez la requête d'authentification et assurez-vous qu'une application ou service cible autorisé et enregistré auprès de CAS est spécifié.

[Click here](#) to view attributes resolved and retrieved for **admincas**.

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

Liens vers les ressources CAS

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

Copyright © 2005–2018 Apereo, Inc. Powered by Apereo Central Authentication Service 6.0.4 2019-05-21T07:23:20Z

En cliquant sur

Connexion réussie

Bienvenue **admincas**. Vous vous êtes authentifié(e) auprès du Service Central d'Authentification. Toutefois, vous voyez cette page car CAS ne connaît pas votre destination finale ni comment vous y rediriger. Examinez la requête d'authentification et assurez-vous qu'une application ou service cible autorisé et enregistré auprès de CAS est spécifié.

[Click here](#) to view attributes resolved and retrieved for **admincas**.

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

Attribute	Value(s)
cn	[Admincas]
givenName	[Cas]

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 :

```
compile "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-1559902436
```

```
{
  "@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:8080/cas

URL de login SSO : https://cas.domaine-univ.fr:8080/cas/login

CAS

Annuler Sauvegarder les modifications

Activé ☒ Oui ☐ Non

URL de base pour SSO

URL de login SSO

Version CAS

Réinitialiser les paramètres de la section Réinitialiser

Vous devez mettre vos attributs

Gestion des attributs

Toujours synchroniser les données utilisateur ☒ Oui ☐ Non

Attributs de carte

Réinitialiser les paramètres de la section Réinitialiser

Ne pas oublier de mettre le domaine en ssl.

Pour cela, nous utilisons Let's Encrypt

Nous installons avec la commande suivante :

```
apt-get install certbot python-certbot-apache -t stretch-backports
```

Lancer la commande

```
certbot --apache
```

Suivre les indications.

Récupérer le fichier **.pem** du dossier `/etc/letsencrypt/live/cas.domaine.univ.fr` et les copier dans `/var/lib/tomcat8/conf`

```
cp /etc/letsencrypt/live/cas.domaine.univ.fr/*.pem /var/lib/tomcat8/conf
```

ouvrir le fichier dans `/etc/tomcat/server.xml`

Insérer les lignes suivantes :

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

Redémarrer Tomcat

```
systemctl restart tomcat9.service
```

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

Cliquer sur AUTHENTIFICATION CAS



ROCKET.CHAT

AUTHENTIFICATION CAS

Se connecter

[Créer un nouveau compte](#)

[Mot de passe oublié](#)

By proceeding you are agreeing to our [Terms of Service](#), [Privacy Policy](#) and [Legal Notice](#).

Powered by [Open Source Chat Platform Rocket.Chat](#).

English Version

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

Mettre son identifiant et son mot de passe du serveur AD

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](#)

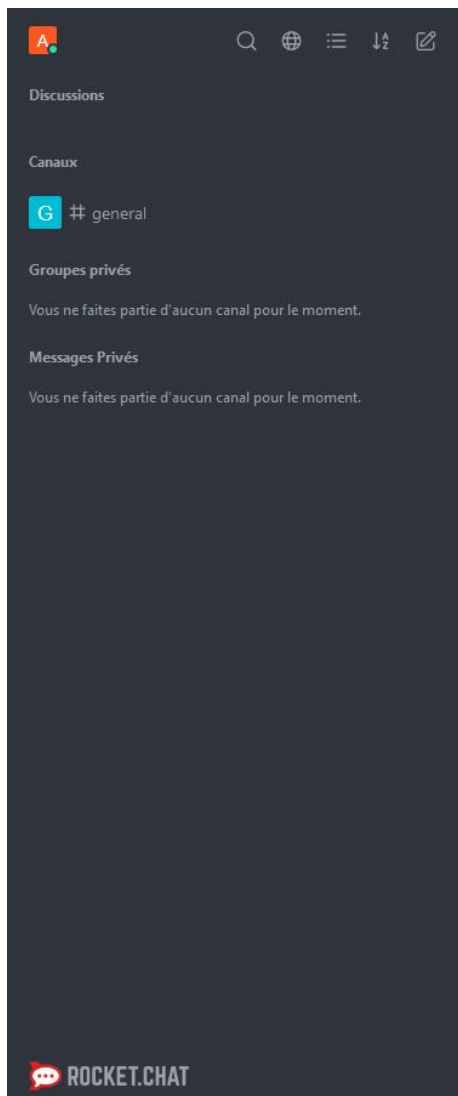
 [Pull Requests](#)

 [Guide pour les
contributeurs](#)

 [Listes de diffusion](#)

 [Salon de discussion](#)

 [Blog](#)



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](#).

Reste à faire

- ☒ Ajouter cas-management