

Guide d'installation d'un serveur OCSP

EJBCA

EJBCA [fr]

Version 1.3

Le 17/07/2009

Identifiant : -

Fichier original : 9999-02_DOC_EJBCA_Guide-Installation-OCSP_1.3.odt

Historique des évolutions et visas

Visas

	RÉDACTION	APPROBATION	VALIDATION
NOM	David CARELLA		
FONCTION	Expert PKI		
DATE			
VISA			

Historique des évolutions

L'ajout de son nom par tout nouvel **acteur contributeur** exprime son consentement exprès et non vicié à une distribution du document conforme à la licence indiquée en page 3.

VERSION	DATE	ACTEUR CONTRIBUTEUR	OBJET DE L'ÉVOLUTION
1.0	14/09/2006	Bruno BONFILS	Création et version initiale
1.1	26/02/2007	David CARELLA	Relecture et corrections
1.2	15/05/2007	Bruno BONFILS	Mise à jour
1.3	17/07/2009	David CARELLA	Conversion vers le modèle de documents Linagora

Statut du document : **60 – En application**

Licence, diffusion et contributeurs

Licence

Ce document est licencié cumulativement sous licences **GNU FDL 1.3** et **CC-BY-SA 3.0**.

La **GNU FDL** est une licence libre copyleft calquée sur la GNU GPL, parfaitement adaptée aux documentations et qui nécessite que soit annexé systématiquement le texte de la licence.

La **CC-BY-SA** est une licence libre copyleft parfaitement adaptée aux contenus multimédias. Sa grande modularité permet de mixer les réalisations.

Cette double licence permet un usage du document qui soit conforme à l'une ou l'autre des licences. Plusieurs avantages peuvent être avancés :

1. Le contenu sous licence est dès lors compatible avec la totalité des licences qui lui sont adjointes ;
2. L'étendue de la double licence est limitée à celle de la licence la plus permissive ;
3. L'utilisation d'au moins une licence française sécurise la double licence au regard des dispositions françaises.

Exceptions

Par dérogation au paragraphe précédent, certaines exceptions peuvent être apportées à la cession de droits telle que consentie par la licence. Les éléments concernés par ces limitations sont les suivants :

Élément	Titre et/ou description	Licence	Remarques

Diffusion du document

Par dérogation aux paragraphes précédents, la diffusion du document est limitée de la manière qui suit :

Mention de diffusion : Groupe Linagora

NOM	ORGANISME	POUR	MÉDIA
Tous les collaborateurs	Groupe Linagora	Information	GED

Liste des contributeurs

Bruno BONFILS, David CARELLA.

Table des matières

1	Introduction.....	6
1.1	Architecture.....	6
1.2	Liste des étapes.....	6
2	Paramétrage avant compilation.....	7
2.1	Serveur EJBCA.....	7
2.1.1	Paramétrage des informations pour le stockage des données.....	7
2.2	Serveur OCSP.....	8
3	Configuration.....	9
3.1	Configuration de la publication.....	9
3.1.1	Création du service de publication.....	9
3.2	Création des certificats.....	10
3.3	Service de publication.....	10
3.3.1	Utilisation du service de publication.....	10
4	Maintenance.....	12
4.1	Synchronisation manuelle des bases de données.....	12
4.2	Exemple de synchronisation avec MySQL.....	12
4.2.1	Suppression des tables sur la base de données des serveurs OCSP.....	12
4.2.2	Exportation des tables sur la base de données de l'IGC.....	12
4.2.3	Copie des fichiers.....	12
4.2.4	Importation des nouvelles données.....	12
5	Références.....	13

Notations

Le code informatique

Les sorties dans un **terminal** sont représentées ainsi :

```
$ commande1  
# commande2
```

Les commandes à exécuter avec des droits utilisateurs sont précédées du caractère « \$ » tandis que celles à exécuter avec les droits *root* sont précédées du caractère « # ». Ces caractères ne doivent pas être saisis dans la ligne de commandes.

Le contenu ou un extrait de **fichier** est représenté ainsi :

```
# Ceci est un exemple de fichier  
parameter1="value"  
parameter2="<NOM-DU-SERVEUR>"
```

Dans le corps de texte, les noms de variables, les extraits de code et les adresses web seront composés en police à chasse fixe. Exemples :

- l'adresse du site web de Linagora est <http://www.linagora.com/> ;
- le contenu de la variable <DATE> est de la forme JJ/MM/AAAA.

Les notes hors texte

Une **remarque** sera représentée de cette façon :

Ceci est un point sur lequel votre attention doit être attirée.

Une **alerte** sera représentée de cette façon :

Ceci est un **point critique** à prendre en compte et pour lequel votre attention est requise.

Les langues étrangères

Les **mots anglo-saxons** sont composés de deux façons : soit en caractères italiques pour les mots du langage courant ou considérés comme tels, soit en caractères droits pour les titres.

Exemples :

Voici un *English text* dans du texte français.

Le protocole Secure Socket Layer est un ...

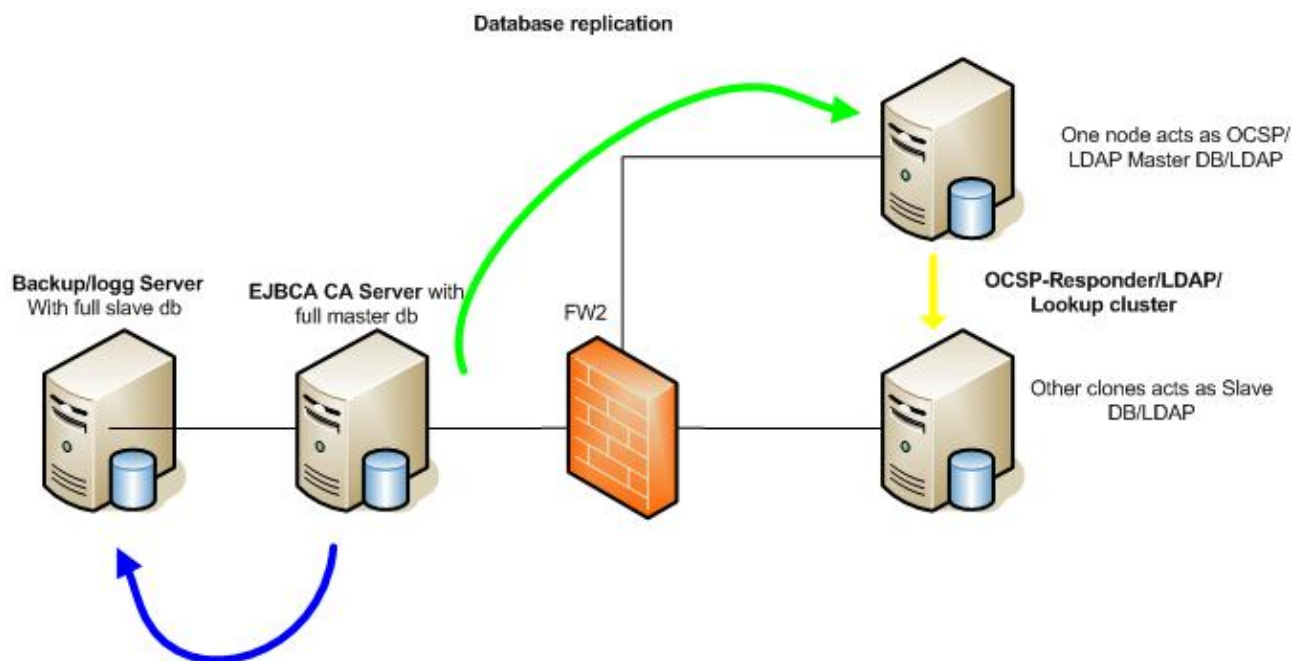
1 Introduction

Ce document couvre l'installation d'un serveur OCSP autonome qui permet une isolation par rapport aux serveurs de l'autorité de certification et d'enregistrement.

Ce type d'architecture est fortement recommandé quand le serveur OCSP est accessible aux publics (utilisateurs, partenaires, Internet, etc.). Ce mode de déploiement permet de rendre accessible le serveur OCSP et d'isoler les entités sensibles, telles que l'AC ou l'AE. En complément, une sécurisation du sens des flux par la mise en place d'un pare-feu est conseillée entre la base de données de l'AC et la base de données du serveur OCSP, le trafic étant autorisé seulement de la base de données de l'AC vers celle du serveur OCSP, l'inverse étant interdit.

Cette architecture peut être aussi utilisée pour un besoin important de performance. Les serveurs pouvant être déployés en mode *cluster*.

1.1 Architecture



1.2 Liste des étapes

- Paramétrage de la base de données côté serveur EJBCA ;
- Paramétrage de la base de données côté serveur(s) OCSP ;
- Configuration du service de publication ;
- Configuration des profils de certificats ;
- Création des certificats OCSP.

2 Paramétrage avant compilation

2.1 Serveur EJBCA

La configuration d'un serveur OCSP externe le paramétrage des deux fichiers de configuration suivants :

- « `ocsp.properties` », fichier de configuration pour la base maître du répondeur OCSP ;
- « `databases.properties` », fichier de configuration pour la base de données locale aux répondeurs OCSP.

2.1.1 Paramétrage des informations pour le stockage des données

La configuration du fichier « `ocsp.properties` » est réalisé depuis l'entité qui signe les requêtes de certificat et de révocation, c'est-à-dire l'autorité de certification. Pour ce faire, il est nécessaire de paramétrer dans le fichier « `ocsp.properties` » les directives suivantes :

```
ocsp-database.url=jdbc:mysql://<ADRESSE-IP>/ocsp
ocsp-datasource.jndi-name=OcspDS
ocsp-database.driver=org.postgresql.Driver
ocsp-database.username=ejbca
ocsp-database.password=<PASSWORD>
ocsp.usecasigningcert=true
ocsp.keys.dir=/opt/java/jboss-<VERSION>.GA/server/ocsp/conf/keys
ocsp.keys.keyPassword=uz89mKwz
```

Directives	Description
<code>ocsp-database.url</code>	URI JDBC de connexion à la base de donnée utilisée pour les répondeurs OCSP
<code>ocsp-datasource.jndi</code>	Nom JDNI de la datasource, il n'est généralement pas nécessaire de modifier ce nom
<code>ocsp-database.driver</code>	Driver à utilisé pour se connecter à la base de donnée
<code>ocsp-database.username</code>	Nom d'utilisateur
<code>ocsp-database.password</code>	Mot de passe
<code>ocsp.usecasigningcert</code>	
<code>ocsp.keys.dir</code>	Répertoire où se trouve les certificats utilisés pour signer les requêtes. Ce certificat doit être généré en utilisant le profil OCSPSIGNER.
<code>ocsp.keys.keyPassword</code>	Mot de passe protégeant les certificats déposés dans le répertoire. Le même mot de passe doit être utilisé pour l'ensemble des certificats.

2.2 Serveur OCSP

Sur le serveur agissant en tant que répondeur OCSP, il convient de configurer les variables définissant les propriétés de la base de données dans le fichier « `database.properties` », typiquement, pour reprendre l'exemple précédent, nous aurons :

```
database.url=jdbc:mysql://<ADRESSE-IP>/ocsp
datasource.jndi-name=OcspsDS
database.driver=org.postgresql.Driver
database.username=ejbca
database.password=<PASSWORD>
ocsp.defaultresponder=CN=AdminCA1,O=EJBCA Sample,C=SE
[...]
```

Attention : les accès à la base de données pour un répondeur OCSP se font au travers des directives `database.*` et non pas `ocsp-database.*` (qui sont utilisés par la partie EJBCA CA). En revanche, les directives `ocsp.*` sont utilisées par le répondeur OCSP, notamment pour spécifier la localisation des certificats utilisés pour signer les réponses OCSP.

Une fois la configuration effectuée, il faut procéder au déploiement du répondeur OCSP via la commande :

```
# ant ocsps-deploy
```

3 Configuration

3.1 Configuration de la publication

3.1.1 Création du service de publication

Pour que l'état de chaque certificat émis par l'IGC soit disponible dans la base de données du répondeur OCSP, il convient tout d'abord de créer un service de publication :

- Cliquer sur le lien « Services de publication » du menu EJBCA ;
- Définir un nom de service (e.g. « Répondeur OCSP externe ») et cliquer sur le bouton « Ajouter » ;
- Sélectionner le service, puis cliquer sur le bouton « Éditer un service de publication » ;
- Choisir « définir le service de publication » comme type de service ;

Nom	OCSP
Type de service de publication	définir le service de publication
définir les paramètres du service de publication:	
Chemin de classe	org.ejbca.core.model.ca.publisher.Exter
Propriété de paramétrage du service de publication	dataSource java:/OcspDS
Paramètres généraux:	
Description	OCSP Responder
<input type="button" value="Sauvegarder et tester la connexion"/> <input type="button" value="Sauvegarder"/> <input type="button" value="Annuler"/>	

- Utiliser « `org.ejbca.core.model.ca.publisher.ExternalOCSPPublisher` » comme chemin de classe ;
- Utiliser « `dataSource java:/OcspDS` » comme paramètre au service ;
- Cliquer sur le bouton « Sauvegarder et tester la connexion » ;
- Si le message « connexion testé avec succès » apparaît, le service de publication est maintenant prêt pour utilisation.

3.2 Création des certificats

Pour que le serveur OCSP puisse répondre aux requêtes, il faut qu'il dispose d'au moins un certificat et de sa clé (qui sera utilisé pour signer les réponses OCSP) ainsi que la chaîne de certificats associé. Pour cela, il convient de créer un certificat avec le profil suivant :

- **Key Usage :**
 - Digital Signatures ;
- **Extended Key Usage :**
 - OCSPSigner.

Une fois le certificat créé, il convient de le déposer dans le répertoire spécifié dans le fichier « `ocsp.properties` » (par défaut `$JBOSS_HOME/bin/keys`). Le certificat doit être au format JKS ou PKCS #12 (p12), le mot de passe de l'entité utilisé pour sa création doit être connu, et également spécifier dans le fichier « `ocsp.properties` ».

Il existe déjà un profil de certificats (nommé OCSPSIGNER) correspondant à ces besoins. Il suffit donc de créer un certificat (un pour chaque AC nécessaire) au format JKS en utilisant un profile basé sur OCSPSIGNER.

Attention ! Les certificats utilisés pour les répondeurs OCSP doivent être publiés dans la base de données des répondeurs, pour cela, ils doivent être créés après la configuration de la publication ! Et donc à fortiori un profil de certificats où la publication est active.

3.3 Service de publication

3.3.1 Utilisation du service de publication

Profil de certificat actuel

ENDUSER (FIXED)
ENDUSER One Day
ENDUSER and OCSP Publisher
OCSPSIGNER (FIXED)
ROOTCA (FIXED)
SUBCA (FIXED)

Editer un profil de certificat Effacer Profil de certificat

Ajouter

 Ajouter Renommer sélection Utiliser le gabarit sélectionné

Maintenant que le service de publication est configuré, il convient de l'utiliser pour les profils de certificats. Par exemple, pour publier tous les certificats utilisant le profil ENDUSER, il convient de créer un autre profil de certificats (e.g. **ENDUSER and OCSP publisher**) à partir du

profil sélectionné, et de modifier les profils d'entités pour utiliser ce nouveau profil de certificats plutôt que l'ancien.

Profil de certificat par défaut	ENDUSER and OCSP Publisher
Profil de certificat disponible	<ul style="list-style-type: none"> ENDUSER ENDUSER One Day <li style="background-color: #e0e0e0;">ENDUSER and OCSP Publisher OCSPSIGNER

4 Maintenance

4.1 Synchronisation manuelle des bases de données

Dans certains cas (coupure réseau, etc.), il peut se produire une désynchronisation de la base de données des répondeurs OCSP par rapport à la base de données maître de l'IGC. Pour procéder à la resynchronisation manuelle, il convient de procéder à la destruction des tables `CertificateData` et `TableProtectData` sur le serveur OCSP pour y insérer les données des mêmes tables depuis la base de données de l'IGC.

4.2 Exemple de synchronisation avec MySQL

1. Suppression des tables du serveur OCSP ;
2. Exportation des tables du serveur de l'IGC ;
3. Copie des fichiers de données sur serveur de l'IGC vers le serveur OCSP ;
4. Importation des nouvelles données.

4.2.1 Suppression des tables sur la base de données des serveurs OCSP

Se connecter localement sur le serveur MySQL des serveurs OCSP :

```
# mysqladmin -u <UTILISATEUR> -p drop <BASE-DE-DONNÉES>
# mysqladmin -u <UTILISATEUR> create <BASE-DE-DONNÉES>
```

4.2.2 Exportation des tables sur la base de données de l'IGC

Se connecter localement sur le serveur MySQL de l'IGC :

```
# mysqldump -u <UTILISATEUR> -p --compress <BASE-DE-DONNÉES> CertificateData >
CertificateData.dat
# mysqldump -u <UTILISATEUR> -p --compress <BASE-DE-DONNÉES> TableProtectData >
TableProtectData.dat
```

4.2.3 Copie des fichiers

Copier les fichiers `CertificateData.dat` et `TableProtectData.dat` du serveur de l'IGC vers celui utilisé par les répondeurs OCSP.

4.2.4 Importation des nouvelles données

Se connecter localement sur le serveur MySQL des serveurs OCSP :

```
# cat CertificateData.dat | mysql -u <UTILISATEUR> -p <BASE-DE-DONNÉES>
# cat TableProtectData.dat | mysql -u <UTILISATEUR> -p <BASE-DE-DONNÉES>
```

5 Références

Références Linagora

RÉFÉRENCE	VER.	PROJET	TITRE OU IDENTIFIANT

Références externes

RÉFÉRENCE	VER.	ÉDITEUR	TITRE OU IDENTIFIANT

Références web

RÉFÉRENCE	TITRE	LANG	ADRESSE WEB