
EJBCA: Integration Guide

THALES LUNA HSM AND DPOD LUNA CLOUD HSM

Document Information

| | |
|-----------------------------|-----------------|
| Document Part Number | 007-013323-001 |
| Revision | N |
| Release Date | 27 January 2021 |

Trademarks, Copyrights, and Third-Party Software

Copyright © 2021 Thales Group. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales Group and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

CONTENTS

| | |
|--|----|
| Overview | 4 |
| Certified platforms..... | 4 |
| Prerequisites | 5 |
| Configure Luna HSM | 5 |
| Configure Luna Cloud HSM service | 7 |
| Set up EJBCA server..... | 10 |
| Set up EJBCA on AWS..... | 10 |
| Integrating Luna HSM or DPoD Cloud HSM with EJBCA..... | 11 |
| Configure Thales Luna crypto library on EJBCA | 11 |
| Create PKCS11 crypto token on EJBCA | 12 |
| Generate keys for EJBCA..... | 12 |
| Create root CA | 13 |
| Configure sub-CAs..... | 14 |
| Create certificate profiles for end entities | 15 |
| Create end entity profiles | 16 |
| Configure Publish Queue Process Service | 17 |
| Configure CRL updater | 18 |
| Contacting Customer Support..... | 19 |
| Customer Support Portal | 19 |
| Telephone Support | 19 |
| Email Support | 19 |

Overview

This document contains detailed steps for integrating EJBCA with a Luna HSM device or Luna Cloud HSM service. EJBCA is an enterprise class PKI Certificate Authority (CA) software that has been built using Java (JEE) technology. It is a robust, high performance, platform independent, flexible, and component based CA to be used stand-alone or integrated with other applications. Luna HSM and Luna Cloud HSM Service secures the EJBCA Certificate Authority (CA) master key, off-loading cryptographic operations from the server to the HSM. The integration between Luna HSM or Luna Cloud HSM service and EJBCA uses the industry standard PKCS#11 interface. EJBCA generates 2048-bit RSA keys on the Luna HSM or Luna Cloud HSM service and the 2048-bit RSA keys are used by the CA for Certificate and CRL signing. The benefits of securing the CA key with Luna HSM include:

- Secure generation, storage and protection of the signing private key on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of the keys.
- Access to the HSM audit trail*.
- The advantage of cloud services with confidence.

*Luna Cloud HSM services do not have access to the secure audit trail.

Certified platforms

This integration is certified on the following platforms:

| HSM Type | Platforms Certified |
|----------|---------------------|
| Luna HSM | RHEL 7, RHEL 6 |

NOTE: EJBCA is tested in both HA and FIPS mode.

Luna HSM: Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic.

| HSM Type | Platforms Certified |
|----------------|---------------------|
| Luna Cloud HSM | RHEL 7 |

Luna Cloud HSM: Luna Cloud HSM provides on-demand HSM and Key Management services through a simple graphical user interface. With Luna Cloud HSM, security is simple, cost effective and easy to manage

because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need.

Prerequisites

Before you proceed with the integration, complete the following tasks:

Configure Luna HSM

If you are using Luna HSM:

1. Verify the HSM is set up, initialized, provisioned and ready for deployment. Refer to [Luna HSM documentation](#) for more information.
2. Create a partition that will be later used by EJBCA.
3. If you are using a Luna Network HSM, register a client for the system and assign the client to the partition to create an NTLS connection. Initialize the Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that the partition is successfully registered and configured. The command to see the registered partitions is:

```
/usr/safenet/lunaclient/bin/lunacm
lunacm (64-bit) v10.2.0-111. Copyright (c) 2020 SafeNet. All rights reserved.
Available HSMs:
Slot Id ->                0
Label ->                  EJBCA
Serial Number ->          1280780175877
Model ->                  LunaSA 7.3.0
Firmware Version ->      7.3.0
Configuration ->         Luna User Partition With SO (PW) Key Export With
Cloning Mode
Slot Description ->       Net Token Slot
FM HW Status ->          FM Ready
```

5. For PED-authenticated HSM, enable partition policies 22 and 23 to allow activation and auto-activation.

NOTE: Refer to [Luna HSM documentation](#) for detailed steps about creating NTLS connection, initializing the partitions, and assigning various user roles.

NOTE: For PED-based Luna HSM, ensure that ProtectedAuthenticationPathFlagStatus is set to '1' in the Misc Section of Chrystoki.conf file.

Set up Luna HSM High-Availability

Refer to [Luna HSM documentation](#) for HA steps and details regarding configuring and setting up two or more HSM boxes on host systems. You must enable the HAOnly setting in HA for failover to work so that if the primary goes down due to any reason all calls automatically route to the secondary until the primary recovers and starts up.

Set up Luna HSM in FIPS mode

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for operation in a FIPS-compliant HSM. If you are using the Luna HSM in FIPS mode, you have to make the following change in the configuration file:

```
[Misc]
RSAKeyGenMechRemap=1
```

The above setting redirects the older calling mechanism to a new approved mechanism when Luna HSM is in FIPS mode.

NOTE: The above setting is not required for Universal Client. This setting is applicable only for Luna Client 7.x.

Control user access to the HSM

NOTE: This section is applicable only for Linux users.

By default, only the root user has access to the HSM. You can specify a set of non-root users that are permitted to access the HSM by adding them to the hsmusers group. The client software installation automatically creates the hsmusers group. The hsmusers group is retained when you uninstall the client software, allowing you to upgrade the software while retaining your hsmusers group configuration.

Add a user to hsmusers group

To allow non-root users or applications access to the HSM, assign the users to the hsmusers group. The users you assign to the hsmusers group must exist on the client workstation.

1. Ensure that you have sudo privileges on the client workstation.
2. Add a user to the hsmusers group.

```
# sudo gpasswd --add <username> hsmusers
```

Where <username> is the name of the user you want to add to the hsmusers group.

Remove a user from hsmusers group

1. Ensure that you have sudo privileges on the client workstation.
2. Remove a user from the hsmusers group.

```
# sudo gpasswd -d <username> hsmusers
```

Where `<username>` is the name of the user you want to remove from the `hsmusers` group. You must log in again to see the change.

NOTE: The user you delete will continue to have access to the HSM until you reboot the client workstation.

Configure Luna Cloud HSM service

You can configure Luna Cloud HSM Service in the following ways:

- > [Standalone Cloud HSM service using minimum client package](#)
- > [Standalone Cloud HSM service using full Luna client package](#)
- > [Luna HSM and Luna Cloud HSM service in hybrid mode](#)

NOTE: Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

Standalone Cloud HSM service using minimum client package

To configure Luna Cloud HSM service using minimum client package:

1. Transfer the downloaded .zip file to your Client workstation using [pscp](#), `scp`, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

[Windows]

```
cvclient-min.zip
```

[Linux]

```
cvclient-min.tar
```

```
# tar -xvf cvclient-min.tar
```

4. Run the `setenv` script to create a new configuration file containing information required by the Luna Cloud HSM service.

[Windows]

Right-click `setenv.cmd` and select Run as Administrator.

[Linux]

Source the `setenv` script.

```
# source ./setenv
```

5. Run the `LunaCM` utility and verify that the Cloud HSM service is listed.

Standalone Cloud HSM service using full Luna client package

To configure Luna Cloud HSM service using full Luna client package:

1. Transfer the downloaded .zip file to your Client workstation using [pscp](#), scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

[Windows]

```
cvclient-min.zip
```

[Linux]

```
cvclient-min.tar
```

```
# tar -xvf cvclient-min.tar
```

4. Run the setenv script to create a new configuration file containing information required by the Luna Cloud HSM service.

[Windows]

Right-click setenv.cmd and select Run as Administrator.

[Linux]

Source the setenv script.

```
# source ./setenv
```

5. Copy the server and partition certificates from the Cloud HSM service client directory to Luna client certificates directory:

Cloud HSM Certificates

```
server-certificate.pem
```

```
partition-ca-certificate.pem
```

```
partition-certificate.pem
```

LunaClient Certificate Directory

[Windows default location for Luna Client]

```
C:\Program Files\Safenet\Lunaclient\cert\
```

[Linux default location for Luna Client]

```
/usr/safenet/lunaclient/cert/
```

NOTE: Skip this step for Luna Client v10.2 or higher.

- Open the configuration file from the Cloud HSM service client directory and copy the XTC and REST section.

Windows

```
crystoki.ini
```

Linux

```
Chrystoki.conf
```

- Edit the Luna Client configuration file and add the XTC and REST sections copied from Cloud HSM service client configuration file.
- Change server and partition certificates path from step 5 in XTC and REST sections. Do not change any other entries provided in these sections.

[XTC]

```
. . .
PartitionCAPath=<LunaClient_cert_directory>\partition-ca-certificate.pem
PartitionCertPath00=<LunaClient_cert_directory>\partition-certificate.pem
. . .
```

[REST]

```
. . .
SSLClientSideVerifyFile=<LunaClient_cert_directory>\server-certificate.pem
. . .
```

NOTE: Skip this step for Luna Client v10.2 or higher.

- Edit the following entry from the Misc section and update the correct path for the plugins directory:

```
Misc]
PluginModuleDir=<LunaClient_plugins_directory>

[Windows Default]

C:\Program Files\Safenet\Lunaclient\plugins\

[Linux Default]

/usr/safenet/lunaclient/plugins/
```

- Save the configuration file. If you wish, you can now safely delete the extracted Cloud HSM service client directory.
- Reset the ChrystokiConfigurationPath environment variable and point back to the location of the Luna Client configuration file.

Windows

In the Control Panel, search for "environment" and select Edit the system environment variables. Click Environment Variables. In both list boxes for the current user and system variables, edit ChrystokiConfigurationPath and point to the crystoki.ini file in the Luna client install directory.

Linux

Either open a new shell session, or export the environment variable for the current session pointing to the location of the Chrystoki.conf file:

```
# export ChrystokiConfigurationPath=/etc/
```

12. Run the LunaCM utility and verify that the Cloud HSM service is listed. In hybrid mode, both Luna and Cloud HSM service will be listed.

NOTE: Follow the [Luna Cloud HSM documentation](#) for detailed steps for creating service, client, and initializing various user roles.

Luna HSM and Luna Cloud HSM service in hybrid mode

To configure Luna HSM and Luna Cloud HSM service in hybrid mode, follow the steps mentioned under the [Standalone Cloud HSM service using full Luna client package](#) section above.

NOTE: Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

To use Luna Cloud HSM Service in FIPS mode

Cloud HSM service operates in both FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, enable the Allow non-FIPS approved algorithms check box when configuring your Cloud HSM service. The FIPS mode is enabled by default. Refer to the Mechanism List in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

Set up EJBCA server

To set up EJBCA server, refer to the EJBCA official documentation page <https://doc.primekey.com/ejbca/ejbca-installation>.

Set up EJBCA on AWS

To set up EJBCA on AWS, ensure that:

1. PrimeKey EJBCA Cloud from Amazon Web Services (AWS) marketplace is deployed and accessible. The EJBCA Cloud documentation is available at: <https://doc.primekey.com/ejbca-cloud>
2. The Luna HSM client is installed and configured on the EJBCA instance and an NTLS connection has been established between the client and the Luna HSM.
3. The EJBCA portal is configured. For details, refer to the [EJBCA Cloud AWS Launch Guide](#).

Integrating Luna HSM or DPoD Cloud HSM with EJBCA

To integrate EJBCA Application Server with a Luna HSM device or Luna Cloud HSM service, complete the following steps:

- > [Configure Thales Luna crypto library on EJBCA](#)
- > [Create the PKCS11 Crypto Token on EJBCA](#)
- > [Generate the keys for EJBCA](#)
- > [Create the root CA](#)
- > [Configure sub-CAs](#)
- > [Create certificate profiles for end entities](#)
- > [Create end entity profiles](#)
- > [Configure Publish Queue Process Service](#)
- > [Configure CRL updater](#)

Configure Thales Luna crypto library on EJBCA

To configure Thales Luna Crypto Library on EJBCA:

1. By default EJBCA locate the Thales Luna Crypto Library from one of the following locations :

- /usr/lunasa/lib/libCryptoki2_64.so
- /usr/lunapci/lib/libCryptoki2_64.so
- /Program Files/LunaPCI/cryptoki.dll
- /usr/safenet/lunaclient/lib/libCryptoki2_64.so
- /opt/thales/dpodclient/libs/64/libCryptoki2.so

If the library is located in another location other than one of the above then open <ejbca_installation_directory>/conf/web.properties and add the following code:

```
cryptotoken.p11.lib.xx.name=<any_crypto_token_lib_name>
cryptotoken.p11.lib.xx.file=<path_to_luna_crypto_library>/crypto_library_name
```

Where xx is the crypto library no. which can be set accordingly and path_to_luna_crypto_library is the path where Luna Client Crypto library is located.

For example:

```
cryptotoken.p11.lib.23.name=Thales Luna Client
cryptotoken.p11.lib.23.file=/home/ejbca/safenet/lunaclient/lib/libCryptoki2_64.so
```

2. Build EJBCA.

```
# ant clean deployear
```

Create PKCS11 crypto token on EJBCA

To create the PKCS11 crypto token on EJBCA:

1. Open a web browser and login to the EJBCA admin page:
`https://<EJBCA Server IP Address>/ejbca/adminweb`
2. Select **Crypto Tokens** under **CA Functions**. The **Manage Crypto Tokens [?]** page will appear on the screen.
3. Scroll to the bottom of the table and click **Create new...** The **New Crypto Token** page will appear on the screen.
4. Enter a name in the **Name** field and select **Type** as **PKCS#11**.

NOTE: This guide uses Thales as the Crypto Token Name for demonstration.

5. Enter the details to create a PKCS11 token using the Luna crypto library name you added earlier. The **Authentication Code** is the Luna HSM Crypto Officer password.

NOTE: The PKCS#11 library, Reference Type, and Reference are selected automatically by EJBCA if crypto library is configured properly. If there are multiple PKCS#11 libraries, then select the Luna crypto library that you have configured.

6. Click **Save**. The following message will appear on the screen: **CryptoToken created successfully**.

Generate keys for EJBCA

Generate the encryption keys for EJBCA using the EJBCA Admin web portal and the Thales Crypto Token. To generate the keys for EJBCA:

1. Access the **Crypto Token : Thales**.
2. Scroll to the bottom of the page and enter a Key Name. Click the Key Size drop-down menu and set a key size.

- Click the **Generate new key pair** button. Repeat this procedure two more times to generate additional keys for the Root CA and Sub CA.

EJBCA
PKI by PrimeKey

Crypto Token : Thales

Back to Crypto Token overview

Switch to edit mode

ID: -224652013
Name: Thales
Type: PKCS11CryptoToken
Used:
Active:
Auto-activation:
Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]:
PKCS#11 : Library: SafeNet Luna Client
PKCS#11 : Reference Type: Slot/Token Label
PKCS#11 : Reference: HA
PKCS#11 : Attribute File: Default

| Alias | Key Algorithm | Key Specification | SubjectKeyID | Action |
|----------------------------------|---------------|-------------------|---|---------------------------------|
| <input type="checkbox"/> signKey | RSA | 4096 | 042687bd2eb21ef983786da96010e06080f2731 | Test Remove Download Public Key |

Remove selected

signKey RSA 4096 Generate new key pair

Create root CA

Verify the availability of the Luna HSM PKCS#11 cryptographic token and use the token to create the EJBCA Root CA. To create the root CA:

- Click **Crypto Tokens** in the EJBCA web portal and verify that the PKCS#11 token is listed in the **Manage Crypto Tokens [?]** table. Additionally, verify that the entry displays the Thales Luna PKCS#11 library and slot ID and that the crypto token is in the **Active** and **Used** state.
- Click the **Certification Authorities** tab from the left pane and then add a CA name. As an example, this guide uses “ExampleRootCA” as the CA name.

EJBCA
PKI by PrimeKey

Manage Certification Authorities [?]

List of Certification Authorities

ManagementCA, (Active)

Edit CA Delete CA Import CA keystore... Import CA certificate...

Create Authenticated Certificate Signing Request

Add CA

ExampleRootCA Create... Rename selected

- Click **Create**.
- Make changes, as indicated in the example below:

Signing Algorithm: SHA256WithRSA

Crypto Token: Thales.

```
defaultKey=defaultKey
certSignKey=signKey
Description: Root CA for Example Inc
Subject DN: CN=ExampleRootCA,O=Example Inc,C=RS
Validity: 20y
Default CRL Dist. Point: Click on Generate button.
CRL Expire Period: 1y
CRL Overlap Time: 2d
```

5. Click the **Create...** button.

When the operation gets completed, a new certificate authority will be available in the list of CAs.

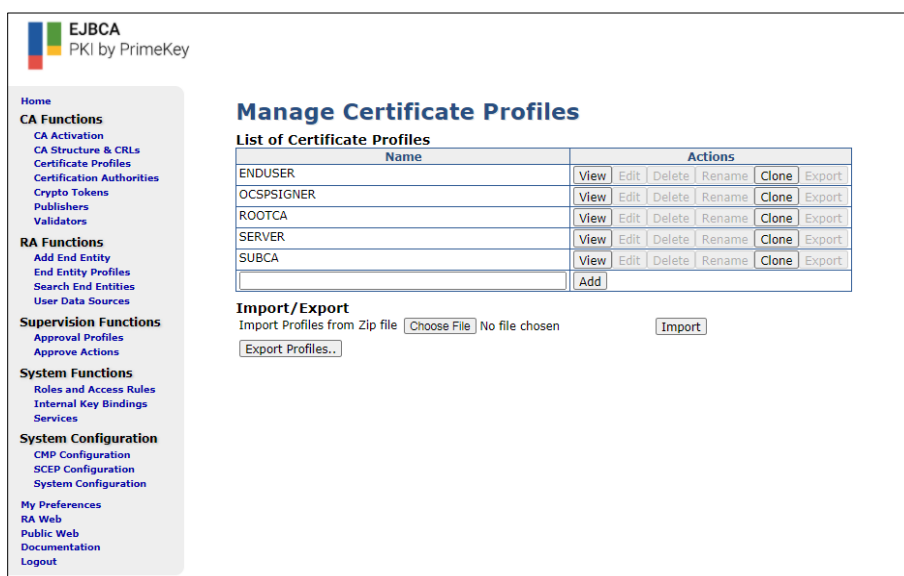
Configure sub-CAs

To configure sub-CAs, complete the following tasks:

- > [Clone the sub-CA template](#)
- > [Create sub-CAs](#)

Clone the sub-CA template

1. Open the **Certificate Profiles** page, from the **List of Certificate Profiles**.
2. Click the **Clone** button next to the **SUBCA** profile.



EJBCA
PKI by PrimeKey

Home

CA Functions

- CA Activation
- CA Structure & CRLs
- Certificate Profiles
- Certification Authorities
- Crypto Tokens
- Publishers
- Validators

RA Functions

- Add End Entity
- End Entity Profiles
- Search End Entities
- User Data Sources

Supervision Functions

- Approval Profiles
- Approve Actions

System Functions

- Roles and Access Rules
- Internal Key Bindings
- Services

System Configuration

- CMP Configuration
- SCEP Configuration
- System Configuration

My Preferences

- RA Web
- Public Web
- Documentation
- Logout

Manage Certificate Profiles

List of Certificate Profiles

| Name | Actions |
|------------|--|
| ENDUSER | View Edit Delete Rename Clone Export |
| OCSPSIGNER | View Edit Delete Rename Clone Export |
| ROOTCA | View Edit Delete Rename Clone Export |
| SERVER | View Edit Delete Rename Clone Export |
| SUBCA | View Edit Delete Rename Clone Export |
| | Add |

Import/Export

Import Profiles from Zip file No file chosen

NOTE: The exact configuration of certificate profiles depends on your use case, the standard you need to be compliant with, and your certificate policy. Configuration should be adapted as necessary.

3. Enter a Name in the **Name of new certificate profile** field. As an example, this guide uses “Example Sub-CA” as the certificate profile name.

4. Click **Create from Template**.

A new certificate profile will appear with properties copied from the SUBCA profile.

Create sub-CAs

1. From the **List of Certificate Profiles** field, select the **Example Sub-CA** and click the **Edit** button. Make the changes to this profile, as indicated in the example below:

Available bit lengths: 2048 bits

Validity: 15y

Allow validity override: Off

CRL Distribution Points: On

Use CA defined CRL Dist. Point: On

Available CAs: ExampleRootCA

2. Click **Save**.

3. Create the CA for issuing certificates to the servers. Open the **Certification Authorities** page, and enter name for example **ExampleServerCA** in the **Add CA** box. Click the **Create** button. Make the changes indicated in the example below:

Signing Algorithm: SHA256WithRSA

Crypto Token: Thales.

defaultKey=myKey

Description: Example's CA in charge of issuing certificates for servers within the organization.

Subject DN: CN=ExampleServerCA,O=Example Inc,C=RS

Signed By: ExampleRootCA

Certificate Profile: Sub-CA

Validity (*y *mo *d) or end date of the certificate: 15y

Default CRL Dist. Point: Click on Generate button

CRL Expire Period (*y *mo *d *h *m): 14d

CRL Overlap Time (*y *mo *d *h *m): 12h

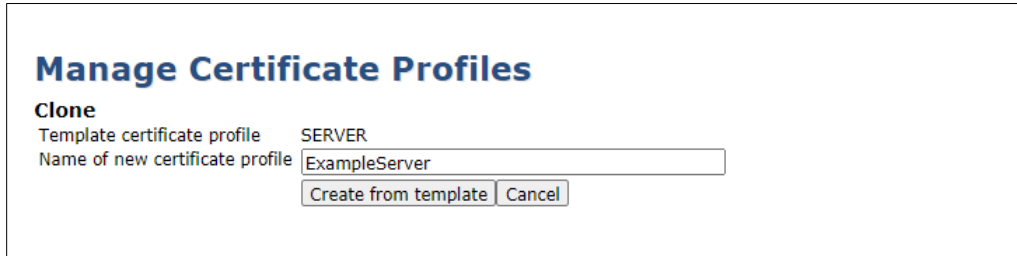
4. Click the **Create** button to finalize the basic CA hierarchy.

Create certificate profiles for end entities

Create certificate profiles for the end entities. Base these profiles on the default EJBCA profiles. To create certificate profiles for end entities:

1. Open the **Certificate Profiles** page from the **List of Certificate Profiles**.
2. Click the **Clone** button next to the **SERVER** profile.

3. Enter a name in the **Name of new certificate profile** field. This guide uses “ExampleServer” as the name.
4. Click the **Create from Template** button.



Manage Certificate Profiles

Clone

Template certificate profile: SERVER

Name of new certificate profile: ExampleServer

Buttons: Create from template, Cancel

A new certificate profile will appear with properties copied from the SUBCA profile.

5. Select the **ExampleServer** certificate profile and click **Edit**. Make the following changes to the certificate profile:

Available bit lengths: 1024, 2048

CRL Distribution Points: On

Use CA defined CRL Dist. Point: On

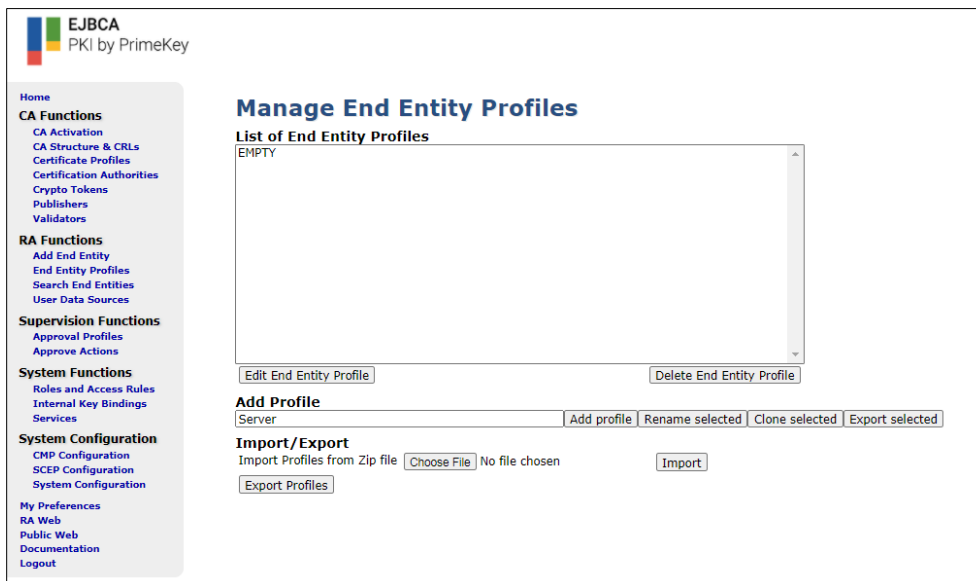
Available CAs: ExampleServerCA

6. Click **Save**. This concludes the creation of basic certificate profiles.

Create end entity profiles

Create the End Entity profiles using the cloned EJBCA certificate profile. To create the end entity profiles:

1. Click the **End Entity Profiles** button and enter any name for example **Server** in the **Add Profile** text box. Click **Add**.



EJBCA
PKI by PrimeKey

Manage End Entity Profiles

List of End Entity Profiles

EMPTY

Buttons: Edit End Entity Profile, Delete End Entity Profile

Add Profile

Server

Buttons: Add profile, Rename selected, Clone selected, Export selected

Import/Export

Import Profiles from Zip file

Buttons: Choose File, No file chosen, Import

Buttons: Export Profiles

2. Select the **Server** profile and click **Edit End Entity Profile**.
3. Add the following Subject DN attributes and mark them all as **Required** and **Modifiable**.

O, Organization

C, Country (ISO 3166)

4. Change the ExampleServer Server profile fields, as indicated in the example below:

Username: Server

Password: Server

Batch generation (clear text pwd storage) use: On

CN, Common name: Server

O, Organization: Example Inc

C, Country (ISO 3166): RS

Default Certificate Profile: ExampleServer

Available Certificate Profiles: ExampleServer

Default CA: ExampleServerCA

Available CAs: ExampleServerCA

Default Token: User Generated

Available Tokens: User Generated

5. Click Save.

Configure Publish Queue Process Service

After you begin publishing certificates and CRLs to remote locations, you need to configure the Publish Queue Process Service that would allow EJBCA to publish certificates and CRLs even in case of a network outage. To configure the Publish Queue Process service:

1. Navigate to the **Administration > Services** page.
2. Enter a service name in the Add Service box. This guide uses “Publish Queue Process Service” as the service name.
3. Click **Add**.

The screenshot shows the 'Manage Services' interface. At the top, there is a header 'Manage Services'. Below it, a section titled 'List of Services' contains a scrollable list with one entry: 'Publish Queue Process Service (Inactive)'. Below the list are two buttons: 'Edit Service' and 'Delete Service'. At the bottom, there is an 'Add Service' section with a text input field, an 'Add' button, and two other buttons: 'Rename selected' and 'Clone selected'.

4. Select the **Publish Queue Process Service** tab and then click the **Edit Service** button. Enter the following information:

Select Worker: Publish Queue Process Service

Select Interval: Periodical Interval

Period: 1 minutes

Select Action: No Action

Active: On

Pin to Specific Node(s): ca.example.com

Description: Publish certificates and CRL's from the publisher queue.

5. Click the **Save** button.

Configure CRL updater

The CRL updater generates CRLs and regenerates CRLs and certificates as soon as they expire. To configure the CRL updater:

1. Navigate to the **Administration > Services** page.
2. Enter a service name in the **Add Service** box. This guide uses "CRL Updater" as the service name.
3. Click **Add**.
4. Select the **CRL Updater** service and click **Edit Service**. Enter the required information as indicated in the example below:

Select Worker: CRL Updater

CAs to Check: ExampleRootCA, ExampleServerCA

Select Interval: Periodical Interval

Period: 5 minutes

Select Action: No Action

Active: On

Pin to Specific Node(s): ca.example.com

Description: Updates the CRL's if necessary. Checks are made every 5 minutes.

5. Click **Save** and apply the changes.

This completes the integration of Luna HSM or Luna Cloud HSM with EJBCA.

Contacting Customer Support

If you encounter a problem during this integration, contact your supplier or [Thales Customer Support](#). Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at <https://supportportal.thalesgroup.com>, is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

NOTE: You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.