

Integration Guide

USING DATA SECURITY MANAGER WITH KEYFACTOR EJBICA (PRIMEKEY)

VERSION 1.0

TABLE OF CONTENTS

| | | |
|------------|---|-------------------------------------|
| 1.0 | INTRODUCTION | 2 |
| 2.0 | PREREQUISITES | 2 |
| 3.0 | INTEGRATION STEPS | 2 |
| 3.1 | Create an App and Copy the App API Key | 2 |
| 3.2 | Install PKCS#11 Driver | 4 |
| 3.3 | Create Crypto Token | 5 |
| 4.0 | DOCUMENT INFORMATION | 9 |
| 4.1 | Document Location | 9 |
| 4.2 | Document Updates | 9 |
| 4.3 | Revision History | Error! Bookmark not defined. |

1.0 INTRODUCTION

This article describes how to integrate **Fortanix Data Security Manager (DSM)** with **Enterprise Java Beans Certificate Authority (EJBCA)**. It also contains the information that a user requires to:


- Create an App in Fortanix DSM.
- Download and install the Fortanix PKCS#11 driver in the EJBCA server.
- Create PKCS#11 Crypto Token in the EJBCA Admin web.
- Generate key pairs using the Crypto Token to create a Certificate Authority.

2.0 PREREQUISITES

- Sudo privilege or Root access on the EJBCA server.
- Internet connectivity from the EJBCA Server to the Fortanix Service.
- Admin Access to the EJBCA UI to configure the Crypto Token.
- The Fortanix PKCS#11 driver can be downloaded from [here](#).

3.0 INTEGRATION STEPS

3.1 CREATE AN APP AND COPY THE APP API KEY

1. Log in to the Fortanix DSM UI.
2. Click the **Apps** tab. On the Apps page click the create a new app icon  to create a new app.

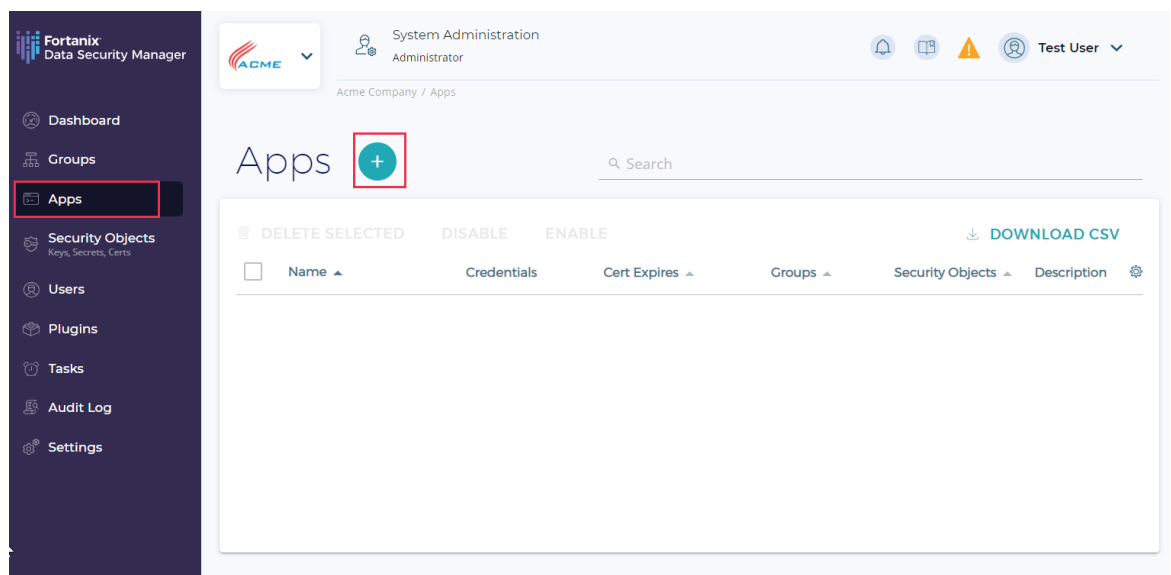


FIGURE 1: CREATE NEW APP

3. Enter the following information:
 - **App name:** This is the name to identify the EJBCA app.
 - **Authentication method:** This can be left at the default **API Key**.
 - **Group:** This is a logical construct that will contain keys created and owned by the EJBCA cluster.
4. Click **Save** to complete creating the application.

FIGURE 2: CREATE APPLICATION

5. Note down the application's API Key to use in *Section 3.3*.
 - a. Go to the detailed view of an app and click the **COPY API KEY** as shown below.

FIGURE 3: COPY APP API KEY

3.2 INSTALL PKCS#11 DRIVER

1. SSH to the EJBCA server.
2. Download the Fortanix PKCS#11 driver.

```
curl -L https://download.fortanix.com/clients/4.2.1500/fortanix-pkcs11-4.2.1500-0.x86_64.rpm -o fortanix-pkcs11-4.2.1500-0.x86_64.rpm
```

3. Install the Fortanix PKCS#11 driver.

```
sudo dnf localinstall -y fortanix-pkcs11-4.2.1500-0.x86_64.rpm  
rm -rf fortanix-pkcs11-4.2.1500-0.x86_64.rpm
```

4. Change to the `wildfly` user and open the `web.properties` file to edit.

```
sudo su - wildfly  
vim /opt/ejbca/conf/web.properties
```

5. Add the following to the end of the `web.properties` file.

```
cryptotoken.p11.lib.60.name=Fortanix  
cryptotoken.p11.lib.60.file=/opt/fortanix/pkcs11/fortanix_pkcs11.so
```

6. Save and close the file and exit the `wildfly` account.

```
:wq  
exit
```



NOTE: To log in to Fortanix DSM from the Docker EJBCA container and create keys, add the following command to `/opt/primekey/bin/start.sh`:

```
export FORTANIX_API_ENDPOINT=https://sdkms.fortanix.com
```

The above command is for Linux only.

3.3 CREATE CRYPTO TOKEN

1. Restart the Wildfly Application Server.

```
sudo systemctl restart wildfly
```

2. Access the EJBCA adminweb with a web browser.
3. Click **Crypto Tokens** in the left navigation pane to create a new crypto token.

EJBCA
PKI by PrimeKey

Version : EJBCA 7.8.2.1 Enterprise (d1f1833cd9d9e11d4a5235be211b1268e1eacfb1)

Welcome 1Feb2022-Skyrim-SA to EJBCA Administration.

Node hostname ejbca01.solitude.skyrim
Server time 2022-03-17 20:22:33-04:00

| CA Status[?] | | |
|------------------|------------|------------|
| CA Name | CA Service | CRL Status |
| ManagementCA | ✓ | ✓ |
| Solitude-Root-CA | ✓ | ✓ |
| Solitude-Sub-CA | ✓ | ✓ |

| Publisher Queue Status[?] | |
|------------------------------------|--------|
| Publisher | Length |
| validationAuthorityPeerPub-OCSP-01 | 22 |

FIGURE 4: EJBCA ADMINWEB

4. Click the **Create new...** link to create a new crypto token.

EJBCA
PKI by PrimeKey

Manage Crypto Tokens [?]

| Name | Type | Library | Reference Type | Reference | Active | Auto-activation | Used | Actions[?] |
|--------------------|------------|--------------------------|------------------|--------------------|--------|-----------------|------|-------------------|
| ManagementCA | PKCS#11 NG | /usr/lib64/libsoftsm2.so | Slot/Token Label | Management_CA_SLOT | ✓ | ✓ | Yes | Reactivate Delete |
| peeringCryptoToken | PKCS#11 NG | /usr/lib64/libsoftsm2.so | Slot/Token Label | KeyBinding_SLOT | ✓ | ✓ | Yes | Reactivate Delete |
| Solitude-Root-CA | PKCS#11 NG | /usr/lib64/libsoftsm2.so | Slot/Token Label | Root_CA_SLOT | ✓ | ✓ | Yes | Reactivate Delete |
| Solitude-Sub-CA | PKCS#11 NG | /usr/lib64/libsoftsm2.so | Slot/Token Label | Sub_CA_SLOT | ✓ | ✓ | Yes | Reactivate Delete |

[Create new...](#)

FIGURE 5: CREATE NEW CRYPTO TOKEN

- a. In the **Type** field, select **PKCS#11 NG** from the drop down menu.

The screenshot shows the 'New Crypto Token' page in the EJBCA administration interface. The left sidebar contains navigation links for CA Functions (Activation, Structure & CRLs, Profiles, Authorities, Tokens, Publishers, Validators) and RA Functions (Add End Entity, End Entity Profiles, Search End Entities, User Data Sources). The main form area has the following fields and options:

- Name:** A text input field.
- Type:** A dropdown menu with the following options: Azure Key Vault, PKCS#11, **SOFT** (checked), AWS KMS, and **PKCS#11 NG** (highlighted with a red circle).
- Authentication Code:** A text input field.
- Repeat Authentication Code:** A text input field.
- Auto-activation:** A checkbox.
- Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]:** A checkbox.
- Allow export of private keys [?]:** A checkbox.
- Save:** A button at the bottom right.

FIGURE 6: CRYPTO TOKEN TYPE

- Select **Fortanix** from the **PKCS#11 : Library** drop down menu.
- Select **Slot ID** from the **PKCS#11 : Reference Type** drop down menu.
- Use the default value for the **PKCS#11 : Reference** field.
- Type a **Name** for the Crypto token, for example, **Fortanix**.
- Type the **Fortanix App API key** for the **Authentication Code**, and the **Repeat Authentication Code** fields.

The screenshot shows the 'New Crypto Token' page with the following configuration:

- Name:** Fortanix
- Type:** PKCS#11 NG
- Authentication Code:** [Masked with dots]
- Repeat Authentication Code:** [Masked with dots]
- Auto-activation:** ☒ Use
- Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]:** ☒ Use
- PKCS#11 : Library:** Fortanix
- PKCS#11 : Reference Type:** Slot ID
- PKCS#11 : Reference:** 0
- PKCS#11 : Attribute File:** Default
- Save:** A button at the bottom right.

FIGURE 7: CONFIGURE CRYPTO TOKEN

- Click **Save** to save the changes.

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
 Audit Log
System Functions
 Roles and Access Rules
 Internal Key Bindings

New Crypto Token

[Back to Crypto Token overview](#)

Name: Fortanix

Type: PKCS#11 NG

Authentication Code: (existing activation PIN, can not change or set PIN on the token)

Repeat Authentication Code:

Auto-activation: ☒ Use

Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]: ☒ Use

PKCS#11 : Library: Fortanix

PKCS#11 : Reference Type: Slot ID

PKCS#11 : Reference: 0

PKCS#11 : Attribute File: Default

Save

FIGURE 8: SAVE THE NEW CRYPTO TOKEN

6. Use the default name for the key (**signKey**), select the key size (**RSA4096**), and select **Sign and Encrypt** for the key usage.
7. Click the **Generate new key pair** button.

EJBCA
PKI by PrimeKey

Crypto Token : Fortanix

Back to Crypto Token overview

Switch to edit mode

ID: 1799060266

Name: Fortanix

Type: Pkcs11NgCryptoToken

Used: ☐

Active: ☒

Auto-activation: ☒

Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]: ☐

PKCS#11 : Library: Fortanix

PKCS#11 : Reference Type: Slot ID

PKCS#11 : Reference: 0

PKCS#11 : Attribute File: Default

Crypto Token currently does not contain any key pairs.

signKey RSA 4096 Sign and Encrypt

Generate new key pair

FIGURE 9: CREATE KEY PAIR

8. Repeat *Steps 6-7* to create the **defaultKey** and **testKey**.

EJBCA
PKI by PrimeKey

Crypto Token : Fortanix

Back to Crypto Token overview

Switch to edit mode

ID: 1799060266

Name: Fortanix

Type: Pkcs11NgCryptoToken

Used: ☐

Active: ☒

Auto-activation: ☒

Use explicit ECC parameters (ICAO CSCA and DS certificates) [?]: ☐

PKCS#11 : Library: Fortanix

PKCS#11 : Reference Type: Slot ID

PKCS#11 : Reference: 0

PKCS#11 : Attribute File: Default

| | Alias | Key Algorithm | Key Specification | SubjectKeyID | Action |
|--------------------------|------------|---------------|-------------------|--|---------------------------------|
| <input type="checkbox"/> | defaultKey | RSA | 4096 | 87ccc6a57163ea438d3a929b63db37cecc3a378e | Test Remove Download Public Key |
| <input type="checkbox"/> | signKey | RSA | 4096 | b7b83f2ff0ad2038ff58dd69a22376342a0aef35 | Test Remove Download Public Key |
| <input type="checkbox"/> | testKey | RSA | 1024 | 242d7bd5f7a8b4dcf2ea6d2dfbdf2b337a6134d | Test Remove Download Public Key |

testKey RSA 1024 Sign and Encrypt

Generate new key pair

Remove selected

FIGURE 10: CREATE KEY PAIRS

9. The three keys are created, and the crypto token can now be used to create a CA.

4.0 DOCUMENT INFORMATION

4.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

<https://support.fortanix.com/hc/en-us/articles/5708233328020-Using-Fortanix-Data-Security-Manager-with-EJBCA>

4.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com

© 2016 – 2022 Fortanix, Inc. All Rights Reserved.

Fortanix® and DSM Applications are trademarks of Fortanix, Inc. All other trademarks are trademarked by their respective owners.

NOTICE: This document was produced by Fortanix, Inc. (Fortanix) and contains information which is proprietary and confidential to Fortanix. The document contains information that may be protected by patents, copyrights, and/or other IP laws. If you are not the intended recipient of this material, please destroy this document and inform info@fortanix.com immediately.