IBM Enterprise Key Management Foundation - Web Edition - User Guide

Tables of Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectable rights may be used instead of the IBM product,program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

March 2021 Edition

This edition applies to the product IBM Enterprise Key Management Foundation Web:2.0 and to all subsequent releases and modifications until otherwise indicated in new editions. Comments may be addressed to your IBM representative or the IBM branch office serving your locality. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

About this publication

This document describes how to generate and manage pervasive encryption (PE) key with IBM® EKMF Web for Pervasive Encryption (EKMF Web). PE is based on the use of Advanced Encryption Standard (AES) keys. EKMF Web provides centralized key management for IBM cryptographic products on multiple platforms. This document covers recommended strategies for enabling PE, with specific use cases.

Setting up for keys

In order to be able to use EKMF Web to manage keys in various services, the following steps are generally required:

1. You first need to create an instance of that service, or install an agent (e.g. KMG Agent on System z).
2. You then need to create a connection to those services in EKMF Web, which are referred to as *Keystores* or *Keystore connections* throughout this document and the user interface.
3. The last step is to create a key template that specifies the characteristics of the keys to be created, like naming conventions, key algorithm and key length.

This release of EKMF Web supports the following keystores:

- KMG Agent for keys on z/OS that can be used for Pervasive Encryption

Setting up Pervasive Encryption keys

EKMF Web facilitates the generation and management of the keys that you use for Pervasive Encryption (PE) on IBM Z. It is important that you understand that EKMF Web is not part of the functionality of pervasive encryption. In particular, EKMF Web is not involved in these scenarios:

- When the system assigns a key label for a data set through RACF(SAF), DFSMS, or JCL
- When the system uses a key from the ICSF keystore to encipher/decipher a data set

Setting up a keystore for PE

In order to be able to use EKMF Web to manage PE keys, you need to have a KMG Agent installed and running on a z/OS LPAR that has access to an ICSF keystore. You then need to create a connection to it in EKMF Web, which is referred to as *Keystore* or *Keystore connection*.

Complete the following steps to create a KMG keystore:

1. Navigate via the menu to **Key Management** > **Keystores**
2. Click **Create**
3. Select **KMG Agent** as keystore type
4. In the following detail panel, provide the following details and click **Save**

Setting	Description
Name	The name of a KMG keystore, which is used for its identification within the scope of EKMF Web only.
Address or host name	Network address of the LPAR in the network where the EKMF Agent is running, provided by the system programmer/ administrator of the z/OS system
Port number	Port that the EKMF Agent is listening on in the network, provided either by the system programmer/ administrator of the z/OS system or found in the Agent job log
Public key hash	Public key hash of the Agent's identity key, provided in the Agent job log, EKMF Web uses this key to authenticate the EKMF Agent.
Add this keystore to the following keystore groups	(Optional) Specify keystore groups for filtering and management purposes

After the keystore is defined you manage it through the **Key Management** > **Keystores** panel.

Setting up a key template for PE

All keys created in EKMF Web must be created through a key template.

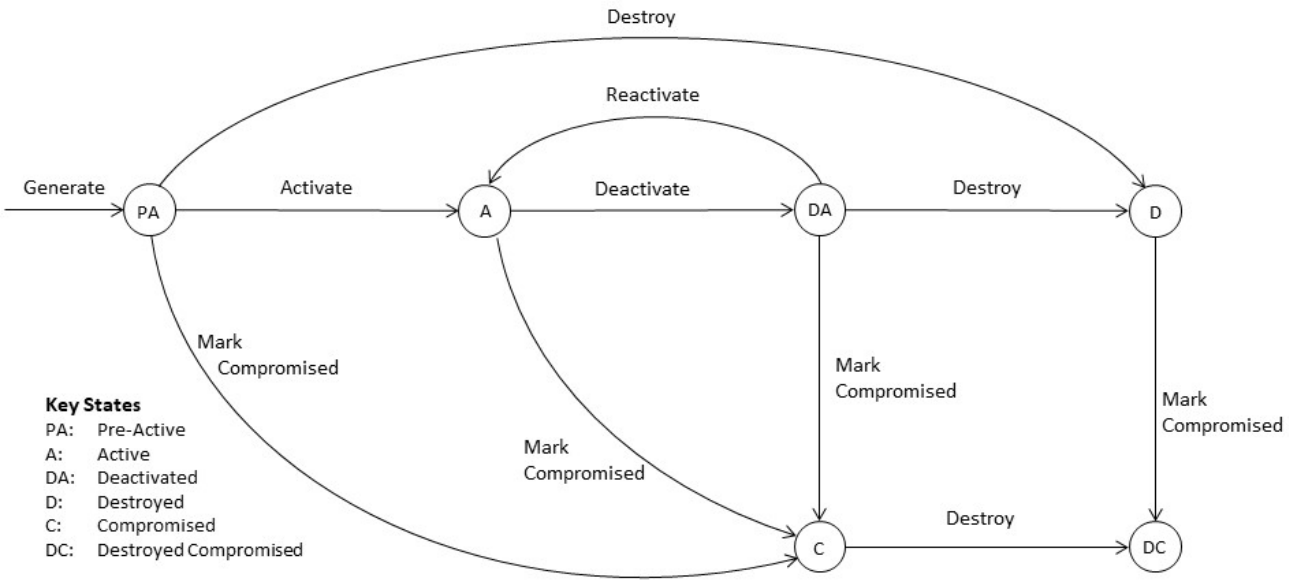
1. Navigate via the menu to **Key Management** > **Key templates**
2. Click the **Create** button
3. In the following detail panel, provide the following details and click **Save**

Setting	Description
Keystore type	The type of keystore, chose Pervasive Encryption to create PE keys

Setting	Description
Name	Specify a name to identify your key template
Key Label (optional)	A key label can contain a number of tags, for example `<env>.AESPE.<app>.<seqno>` Those tags are placeholders that need to be provided during key generation. The `<seqno>` is a special tag. If specified in the key template it will give you the next available sequence number during key generation.
Description (optional)	Description of the key template
Keystore groups	Keystores or keystore groups where keys will be distributed to
Key algorithm	Only AES is available for PE keys
Key size	The only available key size is 256
Key type	Chose between AES Cipher (recommended) or AES Data key
Key state	Chose whether your key will be distributed to the keystore(s) as part of the generation (key state = active) or whether an additional activation step is required (key state = pre-activation)
Allow key export (optional)	Specify whether keys can be exported. Once specified, this setting cannot be changed for this template anymore.
Key's active period	Specify the activation and expiration date for the keys

Key lifecycle Management

Keys in EKMF Web follow the lifecycle that is recommended by NIST.



State	Description	Can be destroyed?

State	Description	Can be destroyed?
pre-activation	First state for all keys that EKMF Web creates. In this state, the key is stored in the central key repository, but no instances of the key have been distributed to keystores.	Yes. You can destroy a **pre-activation** key that was created by mistake. See **destroyed** state in this table.
active	State that follows activation of the key. Activation causes the distribution of key instances to the systems defined in the associated key template.	No. See **deactivated** state in this table.
deactivated	State for a key that is no longer needed. Deactivation removes the key from the keystores it was previously distributed to. It is possible to reactivate a key. This changes the state to **active** and redistributes the key to the keystores it was previously removed from.	Yes. You can destroy a **deactivated** key. See the description of the **destroyed** state in this table.
destroyed	A state in which some key material is removed from the key repository, although other key information is retained.	Not applicable. You can mark a **destroyed** key as also **compromised**. See **destroyed compromised** state in this table.
compromised	A state that falls short of the destroyed state. The key is marked **compromised**, with no further changes. And this key remains in the keystores.	Yes. You can mark a compromised key as also destroyed. See **destroyed compromised** state in this table.
destroyed compromised	A state where your key is marked **compromised** and is also marked **destroyed**.	Not applicable.

EKMF Web allows a **deactivated** or **compromised** key to be reinstalled; this operation redistributes key instances to the systems that are defined in the associated key template.

Creating keys

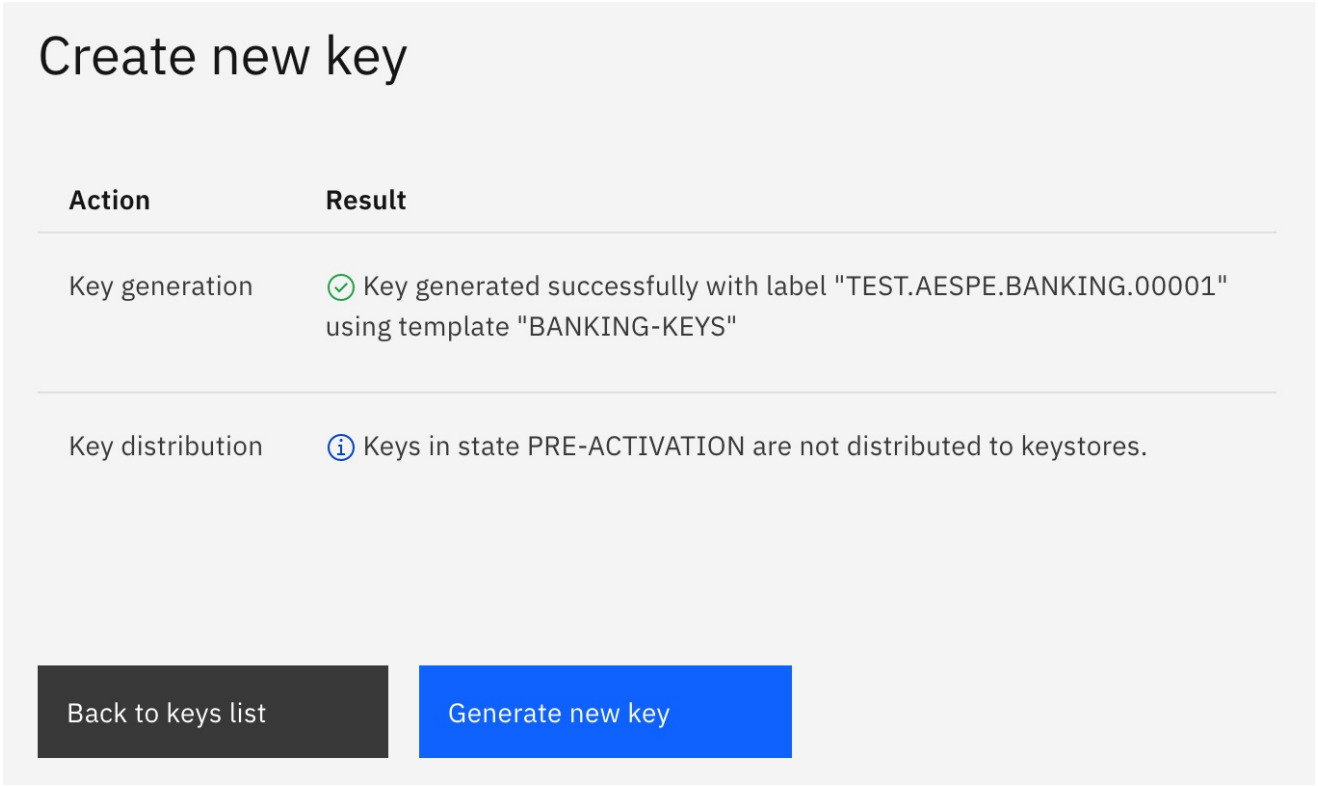
When the setup is completed (i.e. keystores are defined and key templates are created), EKMF Web is ready to generate keys.

1. Go to **Key management** > **Keys** to view a list of your resources.
2. Click **Generate key** on top of the keys table.
3. Specify the key's details and click **Proceed to additional data** and then **Create**

Setting	Description
Template	Select the template that you have defined as part of the setup for your specific type of key
Key label	This field cannot be edited. It will show the key label that has been specified for the selected template. As you proceed providing the required tags, the key label will be updated to show the final name of the key which will be created.
Tag: **`<tag-name>`**	Define each tag as required by the key template's key label
Description (optional)	Provide a description for your key

If your key template has set the key state to **`active`**, the key will automatically be distributed to all the keystores that have been specified in the key template as part of the creation process. If the key template

defines **`pre-activation`** for the key state, the key needs to be activated in an additional step for distribution.



Activating keys

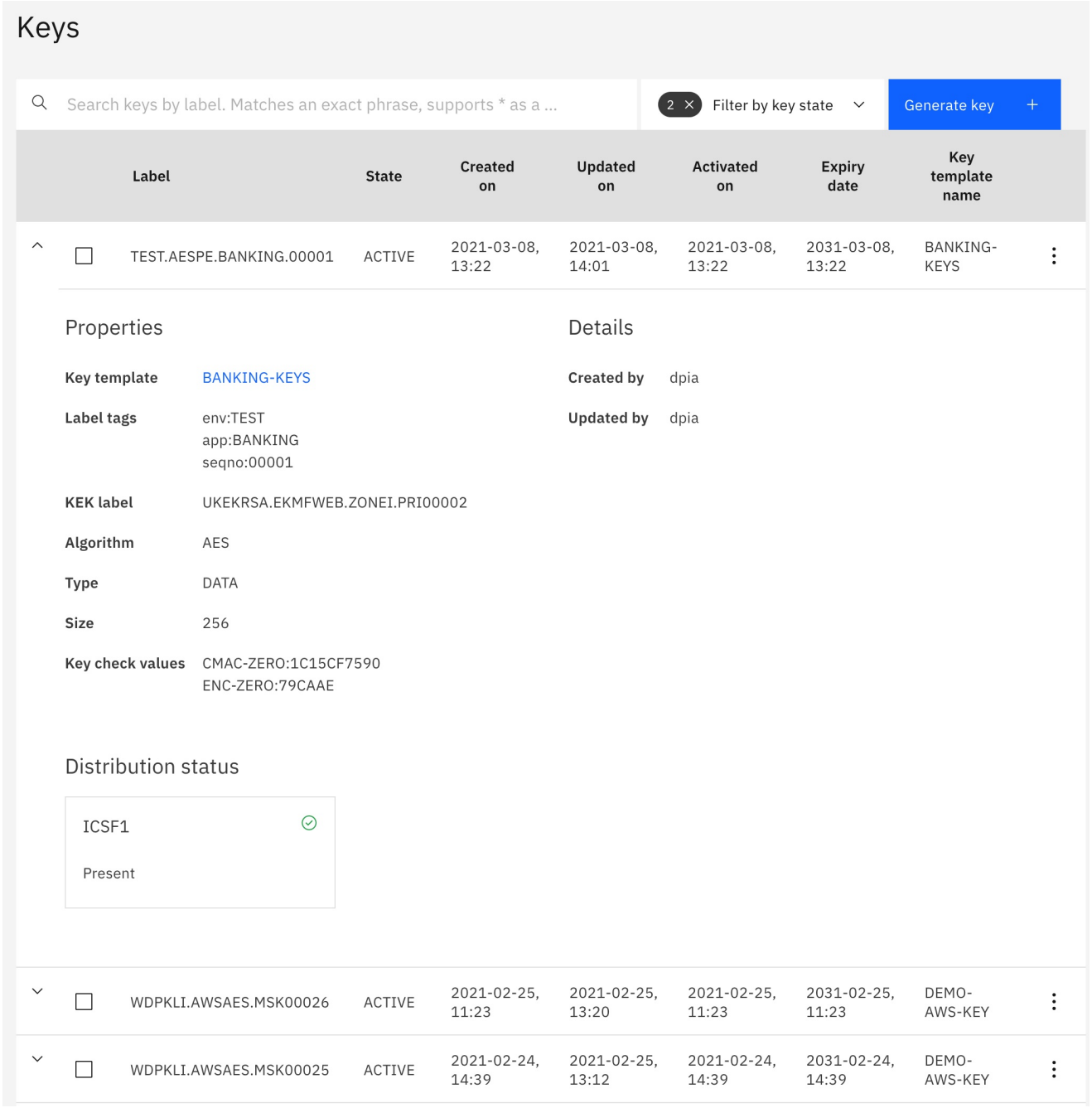
If your key has been created in **`pre-activation`** state, perform the following steps to activate it:

1. Go to **Key management** > **Keys** to view a list of your resources.
2. Select the key that you want to activate and click the overflow icon (⋯) to open a list of options for the key.
3. From the options menu, click **Change state**.
4. Select **Active** as Target state and click on **Change** to confirm.

Verifying the key distribution status

To verify that the keys are distributed and present in keystores, perform the following steps:

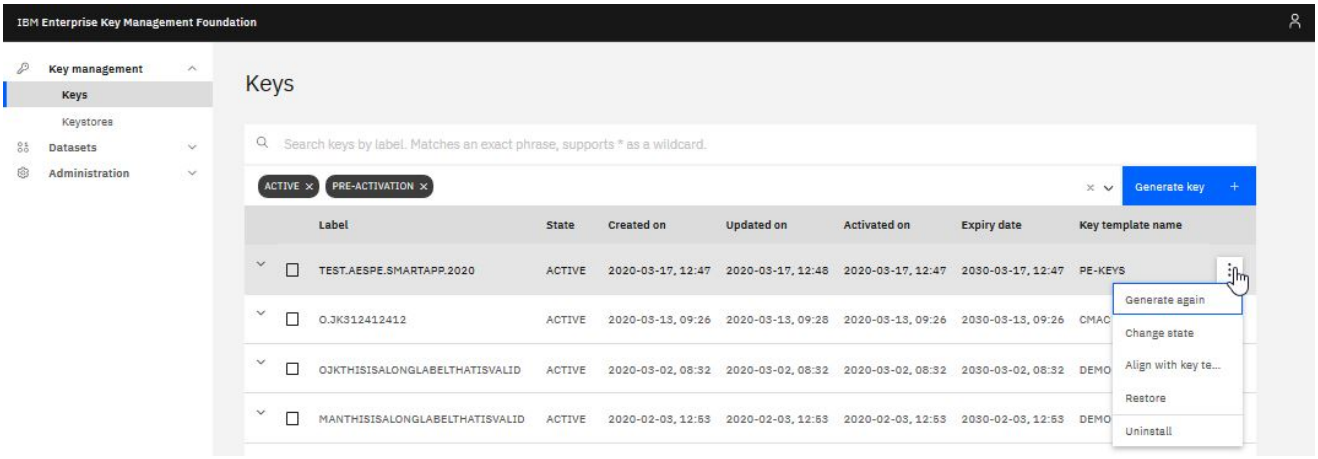
1. Go to **Key management** > **Keys** to view a list of your resources.
2. Locate your key in the list and click on the **`v`**-shaped icon at the beginning of the row to expand it and show the key details section.
3. Verify that the **Distribution status** for all keystores is **`Present`**



Managing keys

All keys can be managed in the Key list view, located at **Key management** > **Keys**. The following operations are available in the overflow menu (⋯) of a key, depending on the current state:

- Generate Again
- Change State
- Align with key template
- Restore
- Uninstall

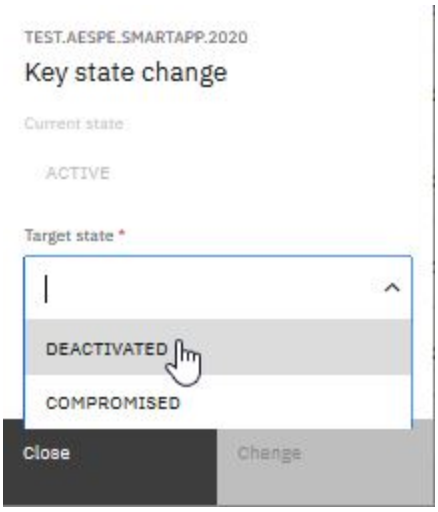


Generate again

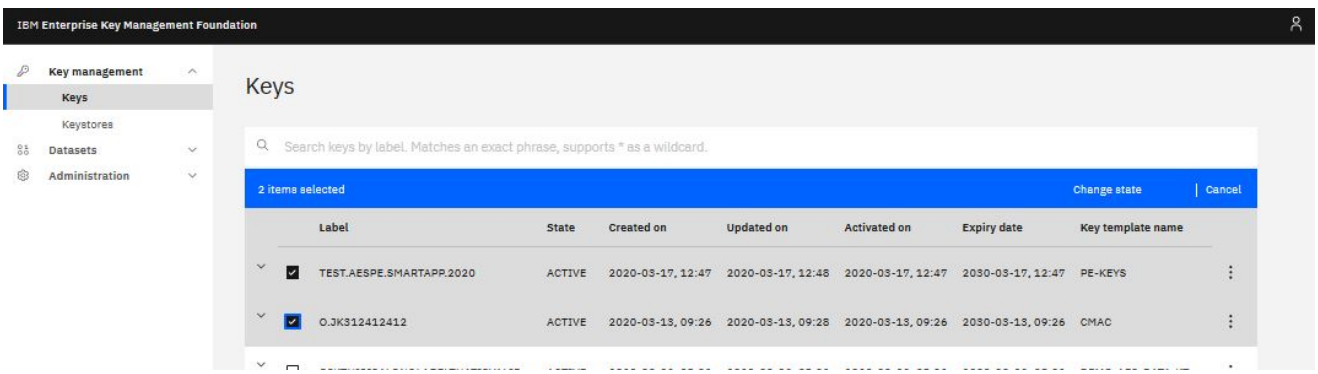
If you need to generate a key with tags filled out similar to an existing key, use the Generate again operation. This prefills the custom tags with the same values as the originating key. In case the **<seqno>** tag is used, the prefilled value will be incremented to the next available sequence number for the template.

Change state

Using this operation on a key allows you to change it's state.



The Key list view also has a multi select function. If you select keys in the same state, you can change state on all of them in one action by pressing the **Change state** in the dark blue bar at the top of the Key list.



Align with key template

If the key template for a given key has been updated, you can align the key to the new version of the template.

What changed in key template	What happens during align
Keystore(s) added	The key will be installed into the additional keystore(s). With this, you can distribute already generated keys to newly defined keystores
Keystore(s) removed	The key will be deleted from the removed keystore(s)
Key label tag change	If the name of a tag changes, tags get added or removed, then the key can not be aligned with the new template
Key label change	If you make changes to your key label that **don't affect the tags**, a new key is generated. The values from the old key label will be used to fill out the new key label and the key with the old label is uninstalled, while the key with the new label is installed to all keystores defined in the new version of the key template.
Key parameters	An align is not possible if other key parameters like key algorithm, key size or key type have been changed in the key template.

Restore

If a key was not distributed to a keystore or is simply missing from a keystore, you can restore it to all keystores. Click **Restore** in the overflow menu of the key. This action distributes the key to all keystores defined in the key template.

Uninstall

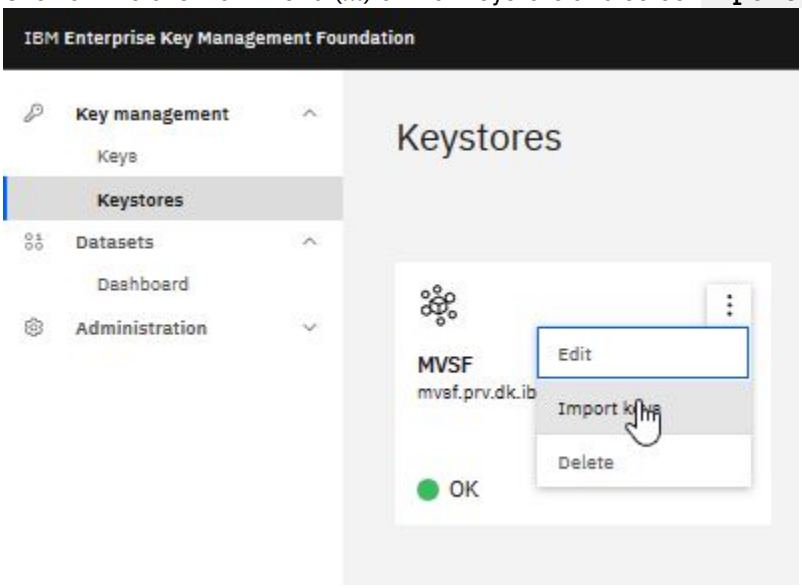
When you want to remove a key from all keystores, without deactivating it, you can uninstall the key. Click **Uninstall** in the overflow menu of the key.

Importing Pervasive Encryption keys

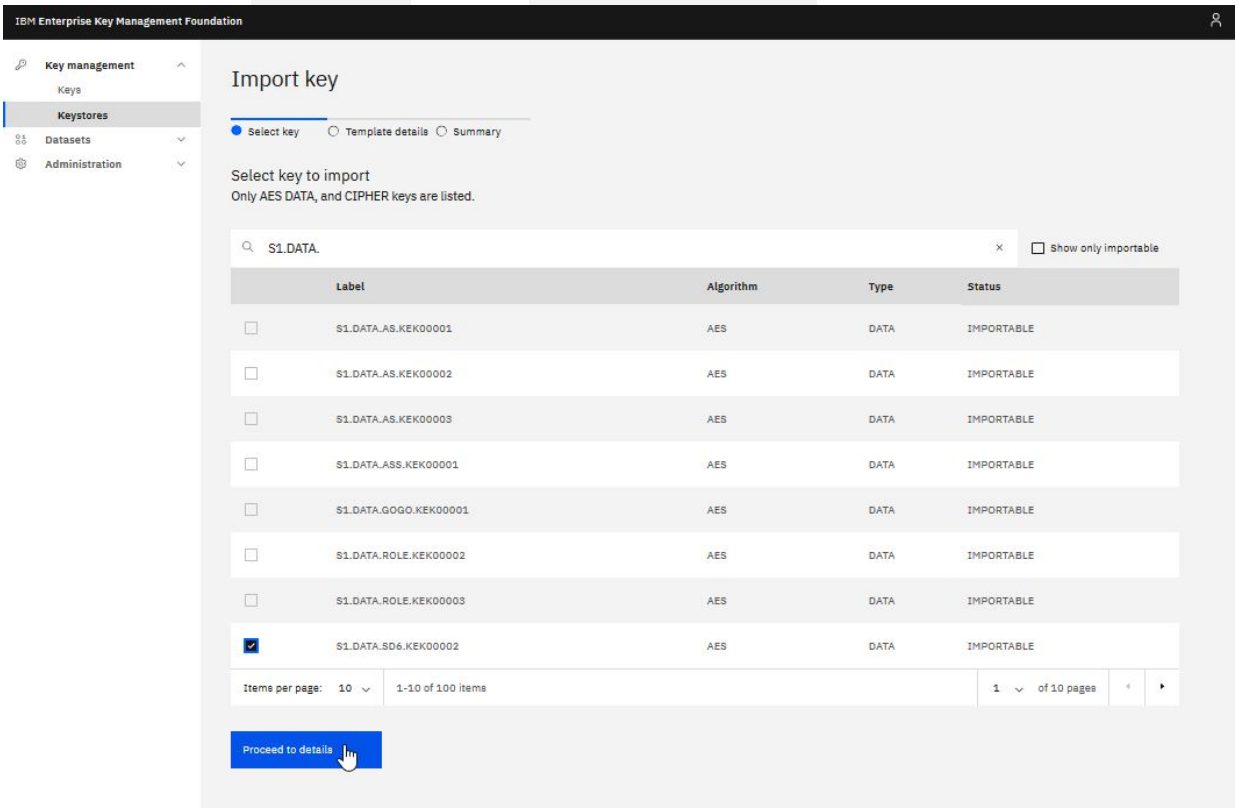
In the event that AES DATA keys for PE have already been generated and stored in a keystore, prior to deploying EKMF Web, it is possible to import the key and store it in the Db2 repository for better backup and recovery. The process is as follows:

1. Go to **Key Management** > **Keystores** menu to see the current keystores.
2. Locate the the KMG keystore that connects to the system you want to import keys from.

3. Click on the overflow menu (...) of that keystore and select **`Import keys`**.

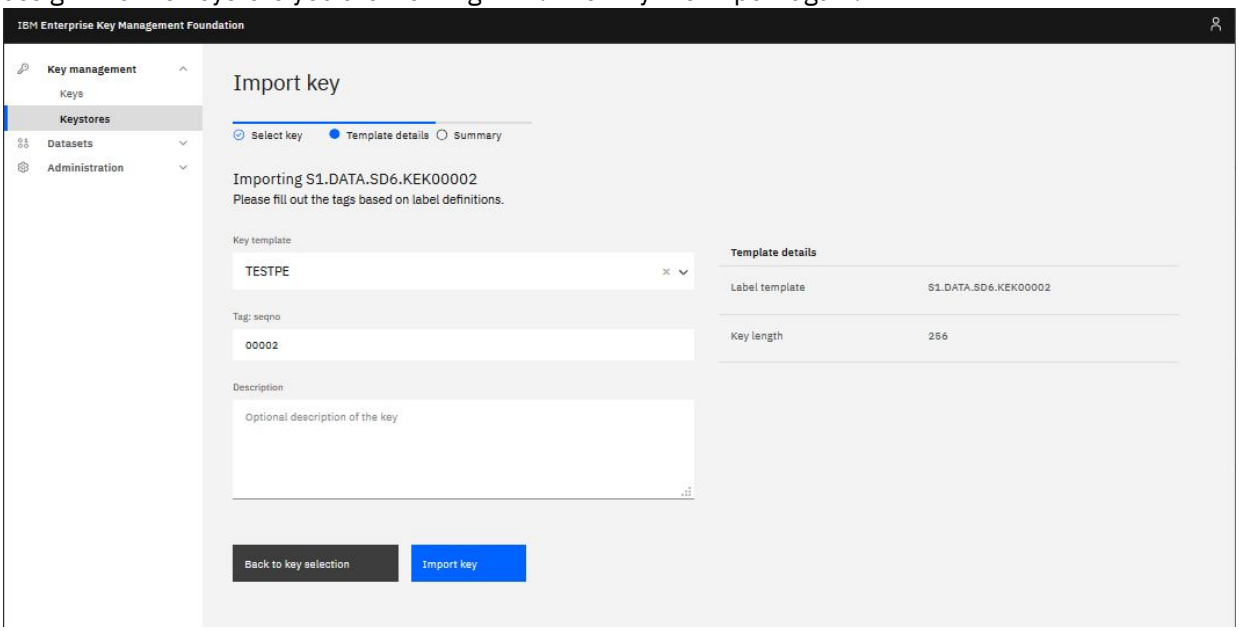


4. The following view will display a list of all AES DATA keys in the keystore including their **`Label`**, **`Algorithm`**, **`Type`** and **`Status`** (**`NON-IMPORTABLE`**, **`IMPORTABLE`**, **`MANAGED`**). Select one or more keys that have a status of **`IMPORTABLE`** and click **`Proceed to details`** at the bottom.



5. If existing key templates are defined in a way that would match the key label, they are available for selection and after choosing one, you can **`Import key`** after which the key is securely imported from the keystore and saved in the EKMF Web repository.
   If no template can be determined, the dropdown will be empty and you will need to first create one and

assign it to the keystore you are working with. Then try the import again.



Keys marked as **`NON-IMPORTABLE`** are AES CIPHER keys which cannot be imported at this time. Keys marked as **`MANAGED`** are already present in the Db2 repository, so import is not necessary.

Rotating recovery key and KEK

1. To rotate the recovery key, first import the new recovery key into the keystore, e.g. via TKE .
2. Navigate via the menu to **Administration** > **Settings**
3. Specify a new key label for
   - **`Key Label for the EKMF Web Recovery Key (AES)`**
   - **`Key Label for KEK for AES CIPHER Keys (AES)`**
   - **`Key Label for KEK for AES DATA Keys (RSA)`**
4. Save settings after each individual update.

From now on, all new keys generated with any key template in EKMF Web will use the new KEKs that are protected with the new recovery key. In order to rotate KEKs for existing keys in the repository, perform the following steps:

1. Navigate to **Key management** > **Keys**, locate the key you want to rotate the KEKs for and expand the details to show the corresponding key template.
2. Click on the key template name in the expanded view and then on **Edit** to edit it. Alternatively, navigate to **Administration** > **Key templates** and locate the key template there.
3. Make any kind of update in the key template, e.g. change the description, and click **Save**.
4. Navigate to **Key management** > **Keys**, click on the overflow menu (...) of the key you want to rotate the KEK for and select **Align with key template**.
5. In the detail panel you will be informed that the Template status is **`Outdated`** and the Alignment progress is **`Can be aligned`**. Click **Align**.
6. (Optional) Verify that the new KEK label is specified in the expanded details view of your key.
7. Repeat the steps for every additional key. If keys share the same key template, then the key template does not need to be edited again. You can directly progress to align the key.

To verify that the new recovery key (e.g. **`TZMKAES.KEYMNGNT.ZONEICSF.KEK00002`**) is used to protect the updated KEKs, you could run an SQL query against the database which will return all KEKs that are still

protected by the old recovery key.

```
select *
from EKMF_WEB_KEYS
where KEY_TEMPLATE_NUMBER in ('AES-W011', 'RSA-W050')
  and KEK_LABEL <> 'TZMKAES.KEYMNGNT.ZONEICSF.KEK00002'
```

To verify that new KEKs (e.g. **TDATAKEK.KEYMNGNT.ZONEICSF.PRI00002** and **TAESKEK.KEYMNGNT.ZONEICSF.IMP00002**) are used by all keys in the repository, you can issue an SQL query like the following to return all keys that are not protected by the new, rotated KEKs.

```
select *
from EKMF_WEB_KEYS
where KEY_TEMPLATE_NUMBER not in ('AES-W011', 'RSA-W050')
  and KEK_LABEL not in ('TDATAKEK.KEYMNGNT.ZONEICSF.PRI00002',
'TAESKEK.KEYMNGNT.ZONEICSF.IMP00002')
```

Use cases for Pervasive Encryption

Always exempt some data sets from encryption, even if they can be encrypted. For example, you MUST not encrypt the EKMF key database as you would not be able to do a recovery in case you loose your keys. Never unconditionally create definitions in RACF(SAF), DFSMS, or JCL that encrypt all data sets. Otherwise, if you encrypt it and the encryption key is lost, the key cannot be restored from the EKMF key repository.

Use case: Separation of duties

One of the main benefits of pervasive encryption is removal of storage administrators from the group of people that have access to data. Storage administrators need access to the encrypted data set. However, they don't need access to the encryption key that makes it possible to decrypt the enciphered data.

Consider this scenario:

- You have an Application A, with RACF profile **PROD.APPLA.****
- The access list for the profile consists of the users who have access to Application A (group **GRPA**) and the storage administrators (group **STGADMIN**)
- You name your encryption key, **PROD.PE.KEYA.01**
- The corresponding RACF profile has an access list, **PROD.PE.KEYA.****

In this scenario, the access list should contain only the group of users with access to the application (group **GRPA**). This approach ensures that only the application users can see the data in clear. All other users only see encrypted data, regardless of the access that they have to the data set.

Application	RACF profile for data sets	Access	Key label	RACF profile for keys	Access
Application A	PROD.APPLA.**	GRPA, STGADMIN	PROD.PE.KEYA.01	PROD.PE.KEYA.**	GRPA

Use case: Separation of application data

You can use pervasive encryption to separate application data, such that Application A cannot read data from Application B. In addition to controlling access to data sets with RACF, each application can have its own encryption key.

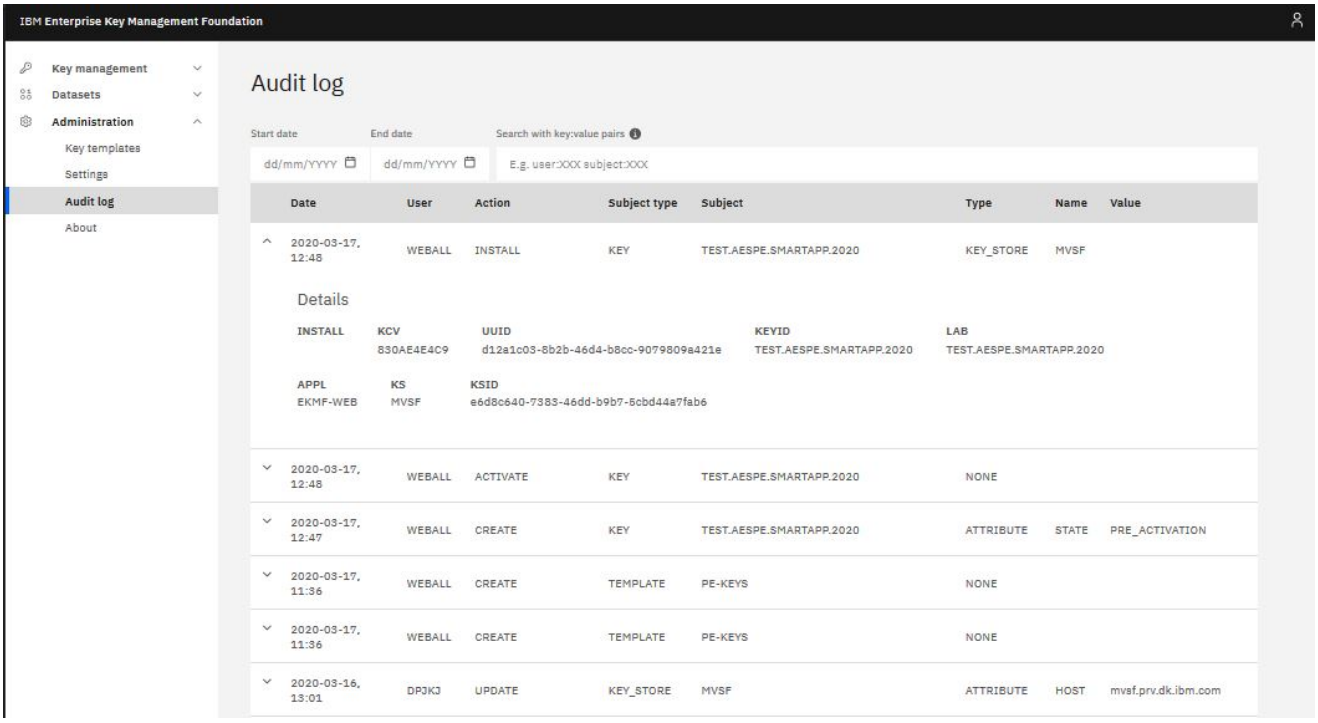
Application	RACF profile for data sets	Access	Key label	RACF profile for keys	Access
Application A	PROD.APPLA.**	GRPA, STGADMIN	PROD.PE.KEY.A.01	PROD.PE.KEY.A.**	GRPA
Application B	PROD.APPLB.**	GRPB, STGADMIN	PROD.PE.KEY.B.01	PROD.PE.KEY.B.**	GRPB

If two applications need to exchange data, yet another key can be created for this.

Application	RACF profile for data sets	Access	Key label	RACF profile for keys	Access
Application A & B	PROD.XCHG.AB.*	GRPA, GRPB, STGADMIN	PROD.PE.KEYXCHG.AB.01	PROD.PE.KEYXCHG.AB.**	GRPA, GRPB

Auditing Events in EKMF Web

All actions performed on keystores, key templates and keys are logged in the audit log Db2 table. This Audit log can be displayed by users with adequate role access and is found in the **Administration** > **Audit log** menu.



The Audit log is searchable with the following search options:

- show log entries in a given date range
- show log entries for a specific user id (case insensitive)
- show log entries for a specific subject:
  - key or key template uuid
  - key label
  - key template number

Trademarks

The following terms are trademarks of other companies:

- Microsoft Azure, Microsoft Corp.
- AWS, Amazon.com, Inc.