



securosys

Securosys External Key Store (XKS) Proxy

Application Note

CloudsHSM or Primus HSM Integration Guide for Securosys External Key Store (XKS) Proxy for AWS KMS



Securosys SA, Max-Högger-Strasse 2,
CH-8048 Zürich, Switzerland
Tel. +41 44 552 31 00 • www.securosys.com
info@securosys.com

Document Information and Revision Control

Version	Date	Author	Description, Changes
1	31.7.2023	MS	Initial document

File: PrimusHSM_XKS_Proxy_AN-E01

Table of Contents

1	Introduction	4
1.1	Target Audience.....	5
1.2	Notations and Symbols.....	5
1.3	Support Contact.....	5
1.4	References and More Information	5
1.5	Glossary	6
1.6	Prerequisites.....	6
2	Primus HSM configuration	7
2.1	Device configuration and partition setup	7
3	Securosys XKS Proxy	10
3.2	Requirements	10
3.3	Docker installation.....	10
3.1	Download Securosys XKS-Proxy files	11
3.2	Securosys XKS proxy configuration	12
3.3	Running Securosys XKS proxy docker image.....	14
3.4	Securosys XKS Proxy Logs	15
3.5	Troubleshooting the Securosys XKS Proxy.....	17
3.6	Securosys XKS proxy Log Error Codes	17
3.7	Updating the Securosys XKS proxy	18
4	Annex	19
4.1	Example .jks file generation for Securosys XKS proxy.....	19
4.3	Example Key Creation in AWS KMS	19

1 Introduction

AWS Key Management Service (AWS KMS) is an encryption and key management service scaled for the cloud. AWS KMS keys and functionality are used by various AWS services, and you can use them to protect data in your own applications that use AWS.

External Key Store (XKS) resources for integration with Amazon Web Services Key Management Service (AWS KMS) allow you to manage keys held in Securosys Primus HSMs (Hardware Security Module) or a Securosys Network CloudsHSM and allows AWS KMS to use the keys for cryptographic operations on demand.

All communication between AWS External Key Store and the Securosys Primus HSM or CloudsHSM is facilitated through the Securosys XKS proxy.

The Securosys XKS Proxy serves as a critical link between AWS Key Management Service (KMS) and the source key material stored in either Securosys Primus HSMs or Securosys Network CloudsHSMs. It empowers you with key sovereignty, allowing you to retain full control over your keys outside of the AWS KMS, ensuring that the cryptographic operations are executed while preserving end-user control and meeting compliance requirements.

Deploying the Securosys XKS proxy is a quick and easy process. By simply configuring and running the Securosys XKS proxy docker image it is possible to establish a link between the AWS KMS and the Securosys HSMs (either on premise or network). The concise Securosys XKS proxy logging can be configured to work on the client server or directed to a remote logging server.

The Securosys XKS proxy can be deployed in different architectures, such as deploying within AWS VPC EC2 instance or a public endpoint connection to AWS services with on premises deployment.

It's important to note that AWS KMS or the Securosys XKS proxy never directly interact with your cryptographic data. Instead, all interactions are only forwarded through the Securosys XKS proxy software that you provide. This ensures that your HSM remains the sole entity responsible for encryption and decryption operations using your cryptographic key material.

This document will assist you in effectively configuring, installing, and managing the Securosys XKS Proxy on the AWS EC2 instance, which connects seamlessly with Amazon VPC Endpoint Service and AWS KMS. By following the steps outlined in this manual, you can ensure a hassle-free setup and configuration process.

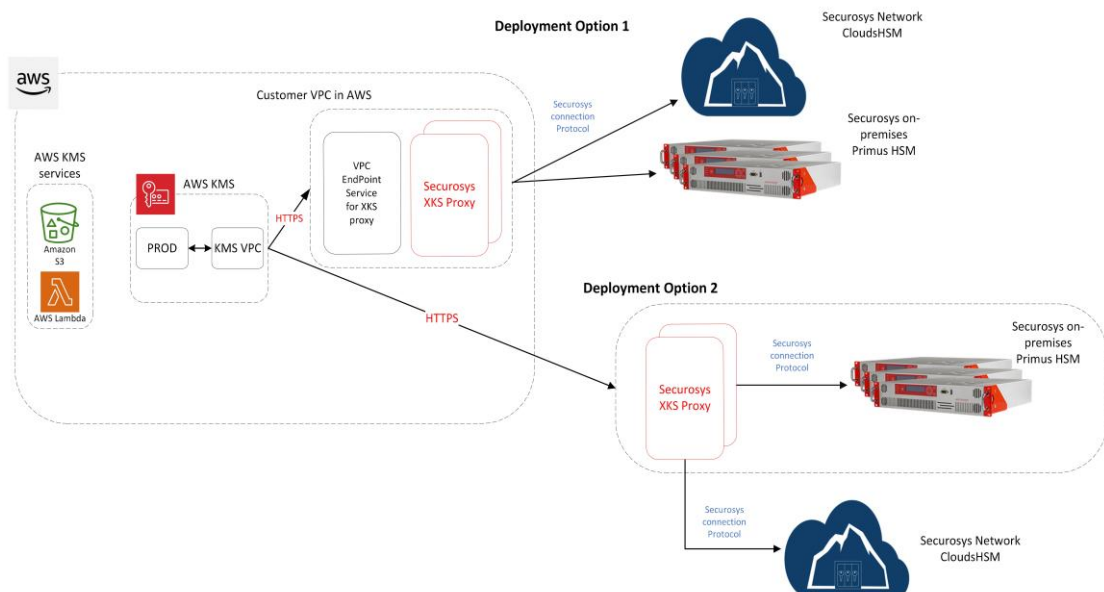


Figure 1: Securosys XKS proxy AWS KMS connection architectures

1.1 Target Audience

This document is intended for Securosys HSM or CloudsHSM and AWS services administrators. Running the Securosys XKS proxy requires that you are already familiar with using Docker Engine.

1.2 Notations and Symbols



NOTE: contains helpful or important information



CAUTION: contains important information that may help prevent unexpected results or data loss



WARNING: be careful and make sure to follow provided instructions

Feature or action requires role activation using

- Genesis Card



- Security Officer (SO) Cards 2 of n




The following symbols are used for HSM configuration setup:

HSM Graphical User Interface, Menu Navigation, e.g.

- SETUP ⌚ CONFIGURATION ⌚ NETWORK ⌚ SERVICES ⌚ JCE ⌚ TCP PORT :2300

and HSM console commands,

 e.g. `hsm_net_list_config serv=2 serv_port`

Configuration file contents, command lines and their output are boxed using the Consolas font. Commands are printed **bold**, values which need to be adapted by the user are marked in **bold blue**, and comments are marked in *italic brown*, e.g.

```
ppin -u -e HSM_USERNAME
...
Provide setup password for 'HSM_USERNAME': <enter User Setup Password, no echo>
...
```

1.3 Support Contact

If you encounter a problem while installing/configuring the provider or integrating the HSM into the Securosys External Key Store (XKS) Proxy, make sure that you have read the referenced documentation. If you cannot resolve the issue, contact your AWS support or Securosys Customer Support. For Securosys XKS Proxy specific requests the Securosys Support Portal is reachable under <https://support.securosys.com>.

1.4 References and More Information

[1] KMS keys in external key store - <https://docs.aws.amazon.com/kms/latest/developerguide/use-xks-key.html>

[2] Primus HSM User Guide latest Edition, downloadable from the Securosys Support Portal: <https://support.securosys.com/external/knowledge-base/article/63>

[3] Securosys CloudsHSM Service
<https://www.securosys.com/en/product/cloudshsm>

1.5 Glossary

Acronym	Definition
API	Application programming interface
AWS	Amazon Web Services
CloudsHSM	HSM as a service, operated by Securosys
ECC	Elliptic-Curve Cryptography
FIPS 140-2	Federal Information Processing Standard 140-2
FW	Firewall
HSM	Hardware Security Module (physical or as a service)
KMS	Key Management service
SSH	Secure Shell
XKS	External Key Store
HA	High Availability

1.6 Prerequisites

Before starting the process of integrating the Securosys CloudsHSM or on-premises Primus HSM with Securosys External Key Store (XKS) Proxy and AWS KMS External Key Store, please make sure to fulfill all the necessary requirements listed below:

- Existing AWS account,
- Configured AWS VPC, AWS KMS external key store successfully connected, Example basic VPC configuration guide shown in the document annex
- Securosys XKS Proxy v1 or newer,
- Securosys Support portal account access,
- Securosys CloudsHSM Service Account (HSM as a Service) or Securosys Primus HSM, firmware v2.8.21, v2.10.5 or newer with JCE API license

2 Primus HSM configuration

Before we can install and configure the Securosys XKS Proxy we need to configure the Securosys on-premises Primus HSM. This chapter describes the initial process of configuring the Primus HSM device and the partition to the point where it can be used with the Securosys XKS proxy. For further information on the Primus HSM setup please see the [Primus HSM User Guide](#).




If you will be integrating your Securosys CloudsHSM partition with your Securosys XKS Proxy or have already correctly configured your Securosys Primus HSM, please skip this chapter.


2.1 Device configuration and partition setup







The subsequent chapters address global device security settings, if you have specific user or partition settings, apply them accordingly using analogous configurations per partition (e.g., `hsm_sec_enter_user_config`, `hsm_user_...`).

The console commands should be adapted as follows:

- To list parameters: use: `hsm_sec_list_config` → `hsm_user_list_config`
- To change parameters: use: `hsm_sec_set_config` → `hsm_user_set_config`

Step	 HSM User Interface (LC Display) Primus X/S-Series	<input type="checkbox"/> HSM Console Primus HSM, all Series
1)		For Primus E-Series (and X-Series) you can setup the device via the console input (<input type="checkbox"/>). Connect a PC (with terminal program) over the serial port with the following settings: 115200 8N1 (speed of 115200bps, 8 data bits, no parity bit, 1 stop bit).
2)	Power-up the device and wait for completion of the boot procedure, the blue moving blue LEDs to settle into 4 steady LEDs. This indicates completion of the power-up sequence and self-tests.	
3)	Verify the firmware version of the device (example): The LC Display shows the version on the lower line right side of the screen: <ul style="list-style-type: none"> • SECUROSYS PRIMUS-HSM-X V2.8.52 	<input type="checkbox"/> Press "ENTER", enter the default login password "ABCD", followed by "ENTER". <input type="checkbox"/> <code>hsm_diagnostics fw</code> (or " <code>hsm_diagnostics frw</code> " on v2.7.x firmware) Device firmware diagnostics: Operation mode: Normal Firmware version: RX-2.8.52
4)	Activation of SO role	
	<ul style="list-style-type: none"> • SETUP ⌚ ROLE ACTIVATION 	<input type="checkbox"/> <code>hsm_so_activation</code>
5)	Install and setup Root Key Store	
	Please ensure that you have copied the obtained license file to a USB stick. Insert the USB stick into the device before proceeding with the following step.	<input type="checkbox"/> <code>hsm_sec_install_rke</code> <input type="checkbox"/> <code>hsm_sec_setup_rks</code>
7)	Configure JCE-API Access on device level To utilize the basic Securosys XKS functionality, ensure that the JCE-API is enabled. Enabling this feature grants access to execute the following endpoint.	

Step	 HSM User Interface (LC Display) Primus X/S-Series	<input type="checkbox"/> HSM Console Primus HSM, all Series
	<ul style="list-style-type: none"> • Service Information (Information about the service) • Synchronous Key Operations (Synchronous operations that are directly forwarded to the HSM.) • Keys (Access to the HSM KeyStore) <p>These settings are NOT optional and must be set in order to correctly integrate the Primus HSM with the Securosys XKS Proxy.</p>	<input type="checkbox"/> hsm_sec_set_config crypto_access=true <input type="checkbox"/> hsm_sec_set_config key_auth=true <input type="checkbox"/> hsm_sec_set_config jce=true
9)	Configure additional device security settings Please note that for a comprehensive understanding of the following settings being configured, it is advised to consult the Primus HSM User Guide.	
	<ul style="list-style-type: none"> • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY SESSION OBJECTS • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY KEY IMPORT • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY KEY EXPORT • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY KEY EXTRACT 	<input type="checkbox"/> hsm_sec_set_config session_objects=true <input type="checkbox"/> hsm_sec_set_config key_import=true <input type="checkbox"/> hsm_sec_set_config key_export=true <input type="checkbox"/> hsm_sec_set_config key_extract=true
10)	Configure Key Invalidation (optional) Activated Key Invalidation creates a shadow copy of the key when it is deleted . This may prevent creation of a new key with the same key name and key id.	
	<ul style="list-style-type: none"> • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY KEY INVALIDATION 	<input type="checkbox"/> hsm_sec_set_config inval_keys=true
11)	Configure Object destruction If set to false, key cannot be deleted (delete will always fail)	
	<ul style="list-style-type: none"> • SETUP CONFIGURATION SECURITY DEVICE SECURITY CRYPTO POLICY OBJECT DESTRUCTION 	<input type="checkbox"/> hsm_sec_set_config destroy_objects=true

Step	 HSM User Interface (LC Display) Primus X/S-Series	 HSM Console Primus HSM, all Series
12)	<p>Create new User / Generate new setup password of existing user (the setup password has limited lifetime, default 3 days)</p> <p><i>Note down the user's setup password. It is required to setup the TSB connection to the HSM.</i></p>	
	<ul style="list-style-type: none"> ROLES  USER  CREATE NEW USER 	<ul style="list-style-type: none"> <input type="checkbox"/> hsm_sec_create_user SO >>> Enter new username: <input type="checkbox"/> TEST_USERGUIDE SO >>> Temporary setup password is: <i>B7GSW-2WjB3-eZZjN-zHnGx-2Sdoc</i> SO >>> User created.
	<ul style="list-style-type: none"> ROLES  USER  NEW SETUP PASSWORD 	<ul style="list-style-type: none"> <input type="checkbox"/> hsm_sec_new_setup_pass SO >>> Enter username: <input type="checkbox"/> TEST_USERGUIDE SO>>> Temporary setup password is: <i>ze2kJ-5aGJG-wwh54-c4pkf-273aw</i> <input type="checkbox"/> SO>>> Successfully finished.

After a successful Securosys Primus HSM configuration the integration with the Securosys XKS proxy is possible. For additional configuration and granularity regarding the configuration please see the [Primus HSM User Guide](#).

3 Securosys XKS Proxy

This chapter assumes that you have already successfully configured your on-premise Primus HSM and have correctly configured the AWS VPC, KMS External Key store and deployed a EC2 instance. For an sample KMS External Key Store configuration please see the Annex chapter 4.2 .

Running the Securosys XKS Proxy requires that you are already familiar with using Docker engine.

It is recommended to establish redundancy in your environment. For more information on Securosys XKS proxy redundancy please refer to AWS documentation [Creating a network load balancer](#).

To configure your Primus HSM devices in High Availability please visit the [\[2\] Primus User Guide](#) chapter 8.1 Manual Cloning.

This chapter focuses on the deployment of Securosys XKS Proxy within the AWS EC2 instance which is deployed in the Customer VPC in AWS.

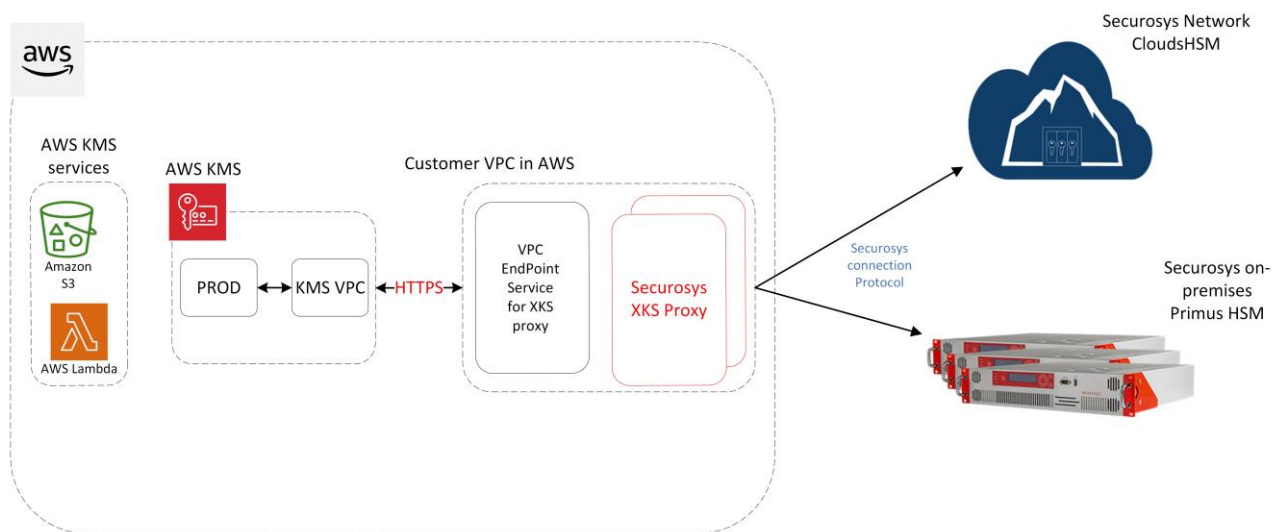


Figure 2: Securosys XKS Proxy deployment within Customer VPC in AWS architecture

3.2 Requirements

Please note on some occasion's commands may require sudo permissions. Your system and docker user permissions should be configured beforehand to avoid any potential permission issues.

3.3 Docker installation

This chapter assumes that you have already configured and deployed an EC2 instance on your AWS account and have configured it with the AWS VPC. For more information on how to deploy an EC2 instance in AWS please visit: [EC2 Get Started](#)

A basic example deployment of a EC2 instance is shown in the annex chapter:

If docker is already installed on your EC2 instance, please skip this chapter, and move to chapter 3 [Securosys XKS Proxy](#).

To install docker on your instance follow the below steps. In the following example we are using the Amazon Linux 2023 AMI, the commands may differ if you are using a different Linux distribution:

- Connect to your AWS EC2 instance via a SSH client, example connection command:

```
ssh -i "XKS_proxy_key.pem" ec2-user@ec2-1-2-3.eu-central-1.compute.amazonaws.com
```

- Replace the "XKS_proxy_key.pem" with your EC2 private key file.
- Replace the "ec2-user@ec2-1-2-3.eu-central-1.compute.amazonaws.com" parameter with your instance public DNS.

For more options on how to connect to your EC2 instance see [Connect to your Linux instance](#).

Execute the following commands to download and install Docker:

- Apply pending updates using the yum command:

```
yum update
```

- To install docker, run the following command:

```
yum install docker
```

- Start docker service:

```
systemctl start docker.service
```

Optionally enable docker service at AMI boot time by executing the following command:

```
systemctl enable docker.service
```

After successfully installing Docker, it is recommended to configure the docker groups and assign users according to your company specifications.

3.1 Download Securosys XKS-Proxy files

Access and credentials to download the Securosys XKS proxy Docker image can be obtained by:

- Downloading from the Securosys Support Portal Securosys XKS Proxy Knowledge Base
- Securosys XKS Proxy pull and download information: [DownloadLink-v.1.0.0.txt](#)

To download the Securosys XKS Proxy follow the example steps below.

- Example command to download the latest version of Securosys XKS Proxy zip file:

```
curl -L -XGET https://<Username>:<Password>@securosys.jfrog.io/artifactory/external-xks/securosys-xks-config-files/securosys_xks_1.0.0.zip -o securosys_xks_1.0.0.zip
```

- Replace the <Username> and <Password> variables with the provided credentials in the Download-Link.txt file.

- Optionally download the latest Release notes for the Securosys XKS Proxy:

```
curl -XGET https://<Username>:<Password>@securosys.jfrog.io/artifactory/external-xks/securosys-xks-release-notes/Release_Notes.md -o securosys-xks_ReleaseNotes.md
```

- Replace the <Username> and <Password> variables with the provided credentials in the Download-Link.txt file.

- Unzip the downloaded securosys_xks_1.0.0.zip file by executing the below command:

```
unzip securosys_xks_1.0.0.zip
```

The unzipped file contains the following content:

Installation directory	File description
~/securosys_xks_1.0.0/config/	Securosys XKS Proxy configuration file directory
~/securosys_xks_1.0.0/config/application.yaml	Application config for the Securosys XKS Proxy Docker container.
~/securosys_xks_1.0.0/config/log/logback.xml	Config file for logging behaviour of securosys-ska docker container

3.2 Securosys XKS proxy configuration

The application.yaml file is used to configure the Securosys XKS proxy.

Make sure to provide your domain .jks file to be able to authenticate to your specified domain, an example on how to create a .jks file is provided in the Annex chapter 4.1 Example .jks file generation for Securosys XKS proxy

The default application.yaml file can be seen below, please adapt the **blue** marked parameters according to your environment:

```
# All necessary credentials from hsm
hsm:
  attestationKeyName: 'attestation-key'
  # Make sure you allowed an outbound firewall rule, to allow traffic to the HSM
  # Hosts should be entered sequentially using the list as in the example below
  host:
    - 'your-proxy_host'
    - 'another-proxy_host'
  #Ports should be entered sequentially using the list just like hosts
  port:
    - 'your-proxy_port'
    - 'another-proxy_port'
  user: 'replace-me_hsm-username'
  setupPassword: 'replace-me_hsm-setupPassword'
  proxyUser: 'replace-me_proxy-username'
  proxyPassword: 'replace-me_proxy-password'

# All necessary credentials from AWS
aws:
  host: 'replace-me_domain-name'
  regionName: 'replace-me_aws-region-name'
  accessKeyID: 'replace-me_keyId'
  secretAccessKey: 'replace-me_secretAccessKey'
  #These parameters should not be modified
  serviceName: 'kms-xks-proxy'
  httpMethodName: 'POST'
  debug: 'false'

# Your dns credentials
server:
  port: 443
  address: 0.0.0.0
  ssl:
    key-store: file:/etc/app/config/replace-me_server.jks
    key-store-password: replace-me_keystore-password
    key-alias: replace-me_keystore-keyname

# Logging configuration
logging:
  config: /etc/app/config/log/logback.xml

#Do not modify!
spring:
  mvc:
    throw-exception-if-no-handler-found: true
```

Host parameters:

Configuration Parameters for hsm:	Description
AttestationKeyName: 'attestation-key'	Specify the attestation key name. If no attestation key has been created, a new one will be generated when running the Securosys XKS proxy image.

Host: - 'your-proxy_host' - 'another-proxy_host'	Specify the host dns or IP of your on premiese Primus HSM or cloudsHSM, example: a-api.cloudshsm.com For high availability (HA) list multiple hosts in seperate lines with “-”
Port: - 'your-proxy_port' - 'another-proxy_port'	Specify the JCE port parameter
User: 'replace-me_hsm-username'	Specify your hsm user (partition) name to be used with the AWS services.
Setup Password: 'replace-me_hsm-setupPassword'	Specify the setup password of your HSM partition, example: B7GSW-2WjB3-eZZjN-zHnGx-2Sdoc Once the first connection is established the connection credentials are stored in a hidden encrypted .secret file
proxyUser: 'replace-me_proxy-username'	Comment or delete the line if not connecting to cloudsHSM. When connecting to your cloudsHSM partition specify the proxy username from your cloudsHSM credentials.
proxyPassword: 'replace-me_proxy-password'	Comment or delete the line if not connecting to cloudsHSM. When connecting to your cloudsHSM partition specify the proxy password from your cloudsHSM credentials.

Aws parameters:

Configuration Parameters for aws:	Description
host: 'replace-me_domain-name'	Replace with your domain name.
regionName: 'replace-me_aws-region-name'	Replace with the region where the XKS proxy will be deployed, example eu-central-1.
accessKeyId: 'replace-me_keyId'	Replace with the KeyID of your AWS KMS External key store.
SecretAccessKey: 'replace-me_secretAccess-Key'	Replace with the secret access key created with your AWS KMS External key store.
ServiceName: 'kms-xks-proxy'	Do not change this parameter. Specifies the service name for AWS.
httpMethodName: 'POST'	Do not change this parameter.
debug: 'false'	Do not change this parameter.

Server parameters:

Configuration Parameters for server:	Description
port	Leave the port at port 443.
address	Keep the server IP address at 0.0.0.0.
ssl: key-store: file:/etc/app/config/replace-me_server.jks	Adapt the path to your domain credentials .jks file to match your environment
ssl: key-store-password: replace-me_keystore-password	Insert your keystore password for the .jks file
ssl: key-alias: replace-me_keystore-keyname	Insert your key store alias

Logging parameters:

Configuration Parameters for logging:	Description
---------------------------------------	-------------

config: /etc/app/config/log/logback.xml	Adapt the path to the logback.xml file to match your environment
---	--

- 'hsm' parameters: Change the 'host', 'port', and 'user' parameters to match your own HSM client configuration.
- Configure the 'aws' parameters:
 - Adjust the 'host' parameter to reflect your DNS hostname.
 - Update the 'regionName', 'access-KeyID', and 'secretAccessKey' parameters to correspond with your AWS KMS configuration for the external key store.
 - If not changed leave the parameters for 'serviceName', 'httpMethodName' and 'debug' as default.
- Specify the 'server' parameters. Within the 'ssl' parameters, provide your DNS credentials.

3.3 Running Securosys XKS proxy docker image

Run the docker image with the following command, replace the **variables** according to your configuration:

```
docker run [-d]-name <NameOfContainer> --add-host <YourHostDomain>:127.0.0.1 \
--network=host -v /home/ec2-user/securosys_xks_1.0.0/config-files:/etc/app/config/ \
securosys.jfrog.io/external-xks/securosys-xks:1.0.0.20230706T1936Z
```

Parameters for running the docker image:

Command parameters	Command and parameter description
[-d]	Optionally use the -d parameter when the xks proxy is desired to be ran in the background.
--name <NameOfContainer>	Sets a name for the running container. Replace <NameOfContainer> with the file name of the compressed container image.
--add-host <YourHostDomain>:127.0.0.1	Adds an entry to the container's /etc/hosts file, mapping <YourHostDomain> to 127.0.0.1 (localhost). Replace <YourHostDomain> parameter with your host domain (used when creating the VPC). The IP address should point to localhost (127.0.0.1)
--network=host	Configures the container to use the host's network stack instead of creating a separate network namespace.
-v /home/ec2-user/securosys_xks_1.0.0/config-files:/etc/app/config/	Mounts the directory home/ec2-user/securosys_xks_1.0.0/config-files from the host machine to /etc/app/config/ inside the container. Replace the path /home/ec2-user/securosys_xks_1.0.0/config-files to match your environment path to Securosys XKS proxy configuration files
securosys.jfrog.io/external-xks/securosys-xks:1.0.0.20230706T1936Z	Link to Securosys jfrog repository for Securosys XKS proxy

Example command:

```
docker run [-d]-name SecurosysXKSproxy--add-host xks.securosys.com:127.0.0.1 \
--network=host -v /home/ec2-user/securosys_xks_1.0.0/config-files:/etc/app/config/ \
securosys.jfrog.io/external-xks/securosys-xks:1.0.0.20230706T1936Z
```

If the command is successful, the Securosys XKS proxy will be started. If not already present, a new attestation key will be generated. The logs from the boot will output a healthy status:

```
2023-07-05 05:56:58,280 INFO [https-jsse-nio-0.0.0.0-443-exec-2] com.xks_proxy.controller.HealthController: AWS REQUEST: HealthStatusRequest(kmsRequestId=1e967d20-9e, kmsOperation=KmsHealthCheck)
```



```

        class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>${LOGS}/xks_proxy.log</file>
<encoder
    class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
    <Pattern>%d %p %C [%t] %m%n</Pattern>
</encoder>

<rollingPolicy
    class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- rollover daily and when the file reaches 10 MegaBytes -->
    <fileNamePattern>${LOGS}/xks_proxy-%d{yyyy-MM-dd}.log
</fileNamePattern>
    <maxHistory>30</maxHistory>
</rollingPolicy>
</appender>

<!-- REMOTE LOGGING -->

<!-- remote logging to a Syslog server -->
<!--
<appender name="SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">
    <syslogHost>127.0.0.1</syslogHost>
    <port>514</port>
    <facility>AUDIT</facility>
    <suffixPattern>%level [%thread] [%logger] %msg</suffixPattern>
</appender>
-->

<!-- remote logging to Splunk -->
<!--
<appender name="SPLUNK" class="com.splunk.logging.TcpAppender">
    <RemoteHost>127.0.0.1</RemoteHost>
    <Port>4560</Port>
    <layout class="ch.qos.logback.classic.PatternLayout">
        <pattern>%date{ISO8601} %level [%thread] [%logger] %msg%n</pattern>
    </layout>
</appender>
-->
<!--
<appender name="SplunkHEC" class="com.splunk.logging.HttpEventCollectorLogbackAp-
pender">
    <url>https://localhost:8088</url>
    <token>58d7c29c-93c5-41b7-bc68-833e144777d5</token>
    <disableCertificateValidation>true</disableCertificateValidation>
    <layout class="ch.qos.logback.classic.PatternLayout">
        <pattern>%msg</pattern>
    </layout>
</appender>
-->

<!-- remote logging to TCP Socket -->
<!--
<appender name="Socket" class="ch.qos.logback.classic.net.SocketAppender">
    <remoteHost>127.0.0.1</remoteHost>
    <port>4560</port>
    <includeCallerData>false</includeCallerData>
    <reconnectionDelay>30000</reconnectionDelay>
    <queueSize>128</queueSize>
    <eventDelayLimit>100</eventDelayLimit>
</appender>
-->

<!-- LOG everything at INFO level -->
<root level="info">
    <appender-ref ref="RollingFile" />
    <appender-ref ref="Console" />
</root>

```



```
<logger name="securosys" level="trace" additivity="false">
  <appender-ref ref="RollingFile" />
  <appender-ref ref="Console" />
</logger>

</configuration>
```

3.5 Troubleshooting the Securosys XKS Proxy

Troubleshooting the Securosys XKS proxy can be done via.

It is possible to troubleshoot the proxy by doing:

- Check the connection to AWS and the Securosys HSM by using the command `GetHealthStatus` built into the AWS CLI. This command is occasionally automatically executed and logged in the Securosys XKS proxy logs.

It is recommended to review the Securosys XKS proxy logs for any error codes and possible connectivity issues. More about the Securosys XKS Proxy logging can be found in chapter 3.6 Securosys XKS proxy Log Error Codes.

3.6 Securosys XKS proxy Log Error Codes

The table below shows the HTTP error code and Error Name returned by the Securosys XKS Proxy to signal various error conditions for different APIs. Column 3 indicates the scenario when the corresponding error is returned and column 4 lists the applicable XKS proxy APIs:

Error Code	Error Name	Error Scenario	XKS Proxy APIs
400	ValidationException	The request was rejected because one or more input parameters are invalid.	ALL except GetHealthStatus
400	InvalidStateException	The request was rejected because the specified external key or key store is disabled, deactivated, or blocked.	ALL
400	InvalidCiphertextException	The request was rejected because the specified ciphertext, initialization vector, additional authenticated data, or authentication tag is corrupted, missing, or otherwise invalid.	Decrypt
400	InvalidKeyUsageException	The request was rejected because the specified key does not support the requested operation.	Decrypt, Encrypt
401	AuthenticationFailedException	The request was rejected due to an invalid AWS SigV4 signature.	ALL
403	AccessDeniedException	The request was rejected because the operation is not authorized based on request metadata.	ALL except GetHealthStatus

404	KeyNotFoundException	The request was rejected because the specified external key is not found.	ALL except GetHealthStatus
404	InvalidUriPathException	The request was rejected because the specified URI path is not valid.	ALL
429	ThrottlingException	The request was rejected because the request rate is too high. The proxy may send this either because it is unable to keep up or the caller exceeded its request quota.	ALL
500	InternalException	This is a generic server error. For example, this exception is thrown due to failure of the backing key manager or failure of a dependency layer.	ALL
501	UnsupportedOperationException	The request was rejected because the specified cryptographic operation is not implemented, or if a parameter value exceeded the maximum size currently supported by a specific implementation beyond the minimum size required by this API specification.	ALL
503	DependencyTimeoutException	The XKS proxy timed out while trying to access a dependency layer to fulfill the request.	ALL

3.7 Updating the Securosys XKS proxy

To update the Securosys XKS proxy with as little downtime as possible follow the next steps:

- Download the latest Securosys XKS proxy file and configure it, refer to chapter 3.1 Download Securosys XKS-Proxy files.
- Run the latest image by running it with `restart always docker` command. Replace the blue marked variables with your environment parameters. Example command:

```
docker run -d --restart always -name <NameOfLatestContainer> --add-host \
<YourHostDomain>:127.0.0.1 --network=host -v /home/ec2-user/securosys_xks_1.0.0/
--network=host -v /home/ec2-user/securosys_xks_1.0.0/config-
files:/etc/app/config/ securosys.jfrog.io/external-xks/securosys-
xks:1.0.0.20231007T130000Z
```

- Stop the previous version of the XKS proxy by running the command:

```
docker stop <NameOfContainer>
```

4 Annex

4.1 Example .jks file generation for Securosys XKS proxy

The .jks domain file is required for the Securosys XKS proxy to authenticate to your domain. To correctly configure the Securosys XKS proxy a path to the .jks file must be adapted in the `application.yaml` file, see parameter `ssl: key-store` in chapter 3.2 Securosys XKS proxy configuration.

There are multiple ways to generate a .jks file. In this example we are using openssl and keytool utilities. It is required to have these utilities preinstalled on the device where the .jks file will be created.

A prerequisite step for this example is to generate a certificate for your domain. For the next steps you will require your *.ca, *.crt files and a private key.

- To generate a .jks from these files you need to combine your *.crt and *.ca files. Manually copy all data from *.ca into *.crt, and then you can use the following command. When prompted provide a password for the newly generated .p12 file Replace the **blue** marked variables with your environment parameters:

```
openssl pkcs12 -export -in abc.crt -inkey abc.key -out abc.p12
```

- To generate the .jks file execute the following command with the **keytool** utility. When prompted provide the same password used with the **openssl** command. Replace the **blue** marked variables with your environment parameters:

```
keytool -importkeystore -srckeystore abc.p12 \  
srcstoretype PKCS12 \  
destkeystore abc.jks \  
deststoretype JKS
```

Make sure to import your .jks file to the AWS EC2 instance where the Securosys XKS proxy will be ran.

4.3 Example Key Creation in AWS KMS

To create keys in AWS Key Management Service (AWS KMS), follow these steps:

Open the AWS Key Management Service (AWS KMS) console at <https://console.aws.amazon.com/kms>.

In the navigation pane, choose "Customer managed keys".

Choose "Create key".

Step 1:

- Key type: Symmetric.
- Key usage: Encrypt and decrypt.
- Advanced options: External key store.
- Click "Next".

Step 2:

- External key stores: Select the external key store you created in the previous part of the instructions.
- External key: Enter the name that will identify the created key.
- Click "Next".

Step 3:

- Enter an alias for your key. It can be a shorthand for the external key ID.
- Click "Next".

Step 4:

- Key administrators: Choose the IAM users and roles who can administer this key through the KMS API.
- Click "Next".

Step 5:

- Key users: Select the IAM users and roles that can use the KMS key in cryptographic operations.

Step 6:

- Check if all the data matches what you entered and ensure that the values are suitable for your needs.
- Click the "Finish" button to complete the process.