THALES

OpenShift Container Platform

INTEGRATION GUIDE

THALES LUNA HSM & DPOD LUNA CLOUD HSM

Document Information

Document Part Number	007-000821-001
Revision	C
Release Date	31 May 2021

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

Red Hat® OpenShift® is a CNCF certified Kubernetes platform and distribution solution. Red Hat OpenShift offers a consistent hybrid cloud foundation for building and scaling containerized applications. Luna HSMs enable you to store keys and manage cryptographic operations to secure container based applications. Following are some of the benefits of using Luna HSMs along with OpenShift container-based applications:

- Secure generation, storage, and protection of cryptographic keys on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of keys.
- HSM audit trail.
- Significant performance improvements by off-loading cryptographic operations from servers.
- Using Cloud services with confidence.

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

Certified Platforms

- Certified platforms for Luna HSM
- Certified platforms for Luna Cloud HSM

Certified platforms for Luna HSM

This integration is certified for Luna HSM on the following platforms:

HSM Type	OpenShift Container Platform
Luna HSM	4.6

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. The 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.

Certified platforms for Luna Cloud HSM

This integration is certified for Luna Cloud HSM on the following platforms:

HSM Type	OpenShift Container Platform
Luna Cloud HSM	4.6

Luna Cloud HSM: Luna Cloud HSM is a cloud-based platform that provides a Cloud HSM service for your organization’s cryptographic operations on DPoD. Using Luna Cloud HSM service 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:

> Set up OpenShift Container Platform

> Configure Luna HSM

> Configure Luna Cloud HSM

Set up OpenShift Container Platform

Refer to OpenShift Documentation for installing and running the OpenShift Container Platform.

Configure Luna HSM

To configure Luna HSM with OpenShift Container Platform:

1. Ensure that the HSM is setup, initialized, provisioned, and ready for deployment.
2. Connect to any Red Hat Enterprise Linux (RHEL) (RHEL7/RHEL8/RHEL CoreOS) host as a user with administrative privileges.
3. Copy the LunaClient Minimal tar file to any location.
4. Create a directory of your choice where you will extract the above file. For example:

   ```
   # mkdir –p /var/usrlocal/luna
   ```

5. Untar the LunaClient Minimal tar file to /var/usrlocal/luna:

   ```
   # tar xvf LunaClient-Minimal-10.3.0-275.x86_64.tar --strip 1 -C
   /var/usrlocal/luna
   ```

6. Change the directory to /var/usrlocal/luna/:

   ```
   # cd /var/usrlocal/luna/
   ```

7. Create a directory for certificates and configuration files.

   ```
   # mkdir -p /var/usrlocal/luna/config/certs
   ```

8. Copy the /var/usrlocal/luna/Chrystoki-template.conf file to /var/usrlocal/luna/config/Chrystoki.conf file.

   ```
   # cp /var/usrlocal/luna/Chrystoki-template.conf
   /var/usrlocal/luna/config/Chrystoki.conf
   ```

9. Set the ChrystokiConfigurationPath environment variable.

   ```
   # export ChrystokiConfigurationPath=/var/usrlocal/luna/config
   ```

10. Set the PATH environment variables for LunaClient utilities.

    ```
    # export PATH="/var/usrlocal/luna/bin/64:${PATH}"
    ```

11. Using the configurator utility, modify the Chrystoki.conf file as the following:

    ```
    # configurator setValue -s Chrystoki2 -e LibUNIX -v
    /var/usrlocal/luna/libs/64/libCryptoki2.so

    # configurator setValue -s Chrystoki2 -e LibUNIX64 -v
    /var/usrlocal/luna/libs/64/libCryptoki2_64.so

    # configurator setValue -s Misc -e ToolsDir -v /var/usrlocal/luna/bin/64

    # configurator setValue -s "LunaSA Client" -e SSLConfigFile -v
    /var/usrlocal/luna/openssl.cnf
    ```

```
# configurator setValue -s "LunaSA Client" -e ClientPrivKeyFile -v
/var/usrlocal/luna/config/certs/dockerclientKey.pem

# configurator setValue -s "LunaSA Client" -e ClientCertFile -v
/var/usrlocal/luna/config/certs/dockerclient.pem

# configurator setValue -s "LunaSA Client" -e ServerCAFile -v
/var/usrlocal/luna/config/certs/CAFile.pem

# configurator removeSection -s "Secure Trusted Channel"
```

NOTE: You can also use any text editor to modify the Chrystoki.conf file.

12. Create a Luna HSM Client certificate for the containers.

```
# vtl createCert -n dockerclient
```

Here dockerclient is the client certificate name.

13. Copy the client certificate to the Luna Network HSM appliance.

```
# scp /var/usrlocal/luna/config/certs/dockerclient.pem admin@10.124.143.158:
```

Here 10.124.143.158 is the HSM IP.

14. Copy the Luna HSM server certificate (server.pem) to /var/usrlocal/luna/config/certs/:

```
# scp admin@10.124.143.158:server.pem /var/usrlocal/luna/config/certs/
```

15. Register the Luna HSM server certificate with the Client.

```
# vtl addServer -c /var/usrlocal/luna/config/certs/server.pem -n 10.124.143.158
```

16. Connect via SSH to the Luna Network HSM appliance and log in to LunaSH.

```
# ssh admin@10.124.143.158
```

Provide the admin password when prompted.

17. Register the client with the Luna HSM.

```
# client register -client dockerclient -hostname dockerclient
```

18. Create a partition, if one does not already exist on the HSM.

```
# partition create -partition <partition_name>
```

19. Assign the partition to the client.

```
# client assignPartition -partition <partition_name> -client dockerclient
```

20. If you want to use multiple containers and all of them have the same ip address, then disable the ntls ipcheck as all the containers appear as the same client to the HSM.

```
# ntls ipcheck disable
```

21. If you want to use multiple containers and all of them have unique ip address, then each container is considered as single client and they all need to have its own configuration file and unique certificates. In this case you can enable ntls ipcheck.

```
# ntls ipcheck enable
```

22. Exit from the HSM SSH session.

23. On the Client workstation, run LunaCM.

```
# lunacm
lunacm (64-bit) v10.3.0-275. Copyright (c) 2020 SafeNet. All rights reserved.
```

```
Available HSMs:
Slot Id ->              0
Label ->
Serial Number ->        1238686731875
Model ->                LunaSA 7.7.1
Firmware Version ->     7.7.1
Bootloader Version ->   1.1.2
Configuration ->        Luna User Partition With SO (PW) Key Export
                         With Cloning Mode
Slot Description ->     Net Token Slot
FM HW Status ->         Non-FM
Current Slot Id: 0
```

24. Initialize Crypto Officer and Crypto User roles for the registered partition.

> **NOTE:** Follow the Luna Network Luna HSM documentation for detailed steps about initializing the partitions and managing various user roles.

25. Create a Dockerfile in /var/usrlocal/ directory.

```
FROM registry.access.redhat.com/ubi8/ubi:latest
# For ubi7 use:
# FROM registry.access.redhat.com/ubi7/ubi:latest
RUN mkdir -p /var/usrlocal/luna
COPY luna /var/usrlocal/luna
ENV ChrystokiConfigurationPath=/var/usrlocal/luna/config
ENV PATH="/var/usrlocal/luna/bin/64:${PATH}"
ENTRYPOINT /bin/bash
```

> **NOTE:** You can also store the configuration files and certificates present in /var/usrlocal/luna/config directory to a secured NFS server and mount it when you run the pod.

Configuring Luna Cloud HSM

To configure Luna Cloud HSM with OpenShift Container Platform:

1. Connect to any Red Hat Enterprise Linux (RHEL) (RHEL7/RHEL8/RHEL CoreOS) host as a user with administrative privileges.
2. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
3. Create a directory of your choice where you will extract the above file.

   For example:

   ```
   # mkdir -p /var/usrlocal/luna
   ```

4. Extract the .zip file into /var/usrlocal/luna:

   ```
   # unzip setup-<service_name>.zip -d /var/usrlocal/luna
   ```

5. Change the directory to /var/usrlocal/luna/:

   ```
   # cd /var/usrlocal/luna
   ```

6. Untar the cvclient-min.tar file.

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

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

```
# source ./setenv
```

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

```
# bin/64/lunacm
```

9. Initialize Crypto Officer and Crypto User roles for the registered partition.

> **NOTE:** Follow the Luna Cloud HSM documentation for detailed steps about initializing various user roles.

10. Create a Dockerfile in /var/usrlocal/ directory.

```
FROM registry.access.redhat.com/ubi8/ubi:latest
# For ubi7 use:
# FROM registry.access.redhat.com/ubi7/ubi:latest
RUN mkdir -p /var/usrlocal/luna
COPY luna /var/usrlocal/luna
ENV ChrystokiConfigurationPath=/var/usrlocal/luna
ENV PATH="/var/usrlocal/luna/bin/64:${PATH}"
ENTRYPOINT /bin/bash
```

> **NOTE:** You can also store the configuration files and certificates present in /var/usrlocal/luna directory to a secured NFS server and mount it when you run the pod.

Deploying Luna HSM or Luna Cloud HSM Pod in OpenShift Container Platform

1. Build a container image using podman/docker.

```
# podman build . -t lunaclient-image
```

2. Verify the built container image.

```
# podman images
```

3. Tag this image and push it to your registry.

```
# podman tag localhost/lunaclient-image <registry_ip>:5000/lunaclient-image
# podman push <registry_ip>:5000/lunaclient-image
```

> **NOTE:** It is recommended to keep the image in private repository which is accessible from your OpenShift Cluster nodes. Once it is pushed to your private registry, delete the local container image.

4. Log in to the Openshift platform.

```
# oc login
```

5. Select or create the project under which you want to deploy the Lunaclient container. For example:

```
# oc project lunaproject
```

6. Create a deployment.yaml file.

```
apiVersion: v1
kind: Pod
```

```
metadata:
  name: luna-client-pod
  labels:
   openshift.io/name: luna-client-pod
spec:
  hostNetwork: true
  restartPolicy: Always
  containers:
    - name: luna-client-pod
      image: "<registry_ip>:5000/lunaclient-image"
      imagePullPolicy: IfNotPresent
      # Just spin & wait forever
      command: [ "/bin/bash", "-c", "--" ]
      args: [ "while true; do sleep 30; done;" ]
```

NOTE: This is the minimal file required for creating a lunaclient pod deployment. You can modify this file according to your need.

7. Deploy the pod.

```
# oc apply -f deployment.yaml
```

8. Verify that the pod is running.

```
# oc get pods
```

```
[root@helper ~]# oc get pods
NAME             READY   STATUS    RESTARTS   AGE
luna-client-pod  1/1     Running   0          22m
```

9. Log in to the pod and verify that the pod has access to the Luna HSM partition.

```
# oc rsh luna-client-pod
```

```
sh-4.2$ lunacm
```

```
sh-4.2$ lunacm
lunacm (64-bit) v10.3.0-275. Copyright (c) 2020 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                lunahsm
        Serial Number ->        1238686731875
        Model ->                LunaSA 7.7.1
        Firmware Version ->     7.7.1
        Bootloader Version ->   1.1.2
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot
        FM HW Status ->         Non-FM


        Current Slot Id: 0
```

This completes the integration of OpenShift Container Platform with Luna HSM or Luna Cloud HSM.

Contacting Customer Support

If you encounter a problem at any stage during this integration, contact 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.