

# OpenStack Barbican Dalmatian

v19.0.0

## Integration Guide

u.trust GP HSM

v6.1.1

**utimaco**<sup>®</sup>

## Imprint

Copyright 2025	Utimaco IS GmbH Germanusstr. 4 D-52080 Aachen Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226) EMEA +49 800-627-3081 APAC +81 800-919-1301
Internet	<a href="https://support.hsm.utimaco.com/">https://support.hsm.utimaco.com/</a>
e-mail	<a href="mailto:support@utimaco.com">support@utimaco.com</a>
Document Version	working version
Date	2025-09-23
Status	<b>DRAFT</b>
Document No.	IG-2025-0032
All rights reserved	<p>No part of this documentation may be reproduced in any form (printing, photocopy or according to any other process) without the written approval of Utimaco IS GmbH or be processed, reproduced or distributed using electronic systems.</p> <p>Utimaco IS GmbH reserves the right to modify or amend the documentation at any time without prior notice. Utimaco IS GmbH assumes no liability for typographical errors and damages incurred due to them.</p> <p>All trademarks and registered trademarks are the property of their respective owners.</p>

# Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	About This Guide .....	4
1.2	Target Audience for This Guide .....	4
1.3	Document Conventions .....	4
1.4	Abbreviations .....	5
<b>2</b>	<b>Overview</b> .....	<b>7</b>
2.1	OpenStack Barbican .....	7
2.2	Utimaco u.trust General Purpose HSM Se-Series .....	7
<b>3</b>	<b>Integration Requirements and Prerequisites</b> .....	<b>8</b>
3.1	Tested Versions .....	8
3.2	Software Requirements .....	8
3.3	Hardware Requirements .....	8
3.4	Prerequisites .....	9
<b>4</b>	<b>Installing and Configuring Utimaco SecurityServer Software</b> .....	<b>10</b>
4.1	Download and Install Utimaco SecurityServer Software .....	10
4.2	SecurityServer PKCS#11 Configuration .....	11
4.3	Create SO User and Initialize a Slot .....	13
<b>5</b>	<b>(Optional) Installing OpenStack Barbican</b> .....	<b>14</b>
5.1	Install OpenStack Barbican .....	14
<b>6</b>	<b>Integrating OpenStack Barbican with Utimaco SecurityServer</b> .....	<b>25</b>
6.1	Configure Barbican to use Utimaco HSM .....	25
6.2	Generating MKEK and HMAC Key on Utimaco HSM .....	27
6.3	Encrypting and Decrypting Secrets .....	28
6.4	Generating a symmetric key in Barbican .....	30
6.5	Storing Public Key, Private Key and Certificate in OpenStackBarbican .....	32
6.6	Keys Rotation/Migration .....	39
<b>7</b>	<b>Further Information</b> .....	<b>43</b>
7.1	References .....	43

# 1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle. All Utimaco SecurityServer product documentation is available from Utimaco's website at <https://utimaco.com/>.

## 1.1 About This Guide

This guide describes how to enable HSM integration with OpenStack Dalmatian Barbican. Utimaco HSM securely stores the MKEK and HMAC used by Barbican.

## 1.2 Target Audience for This Guide

This guide is intended for OpenStack Barbican and HSM administrators.

## 1.3 Document Conventions

The following conventions are used in this guide:

<b>Convention</b>	<b>Use</b>	<b>Example</b>
<b>Bold</b>	Items of the Graphical User Interface (GUI), e.g., menu options	Press the <b>OK</b> button.
<b>Monospaced</b>	File names, folder and directory names, commands, file outputs, programming code samples	You will find the file <code>example.conf</code> in the <code>/exmp/demo/</code> directory.
<b>Italic</b>	References and important terms	See Chapter 3, "Sample Chapter", in the <i>CryptoServer - csadm Manual</i> or [CSADMIN].

Table 1: Document conventions

Special icons are used to highlight the most important notes and information.



*Here you find important safety information that should be followed.*



*Here you find additional notes or supplementary information.*



*This message marks the result expected after the successful execution of an instruction.*

## 1.4 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
<b>API</b>	Application Programming Interface
<b>CSADM</b>	CryptoServer Command-line Administration Tool
<b>CSP</b>	Cryptographic Service Provider
<b>CXI</b>	Cryptographic eXtended Services Interface
<b>DB</b>	Database
<b>GUI</b>	Graphical User Interface
<b>HMAC</b>	Hash-based message authentication code
<b>HSM</b>	Hardware Security Module

<b>IP</b>	Internet Protocol
<b>LAN</b>	Local Area Network
<b>MBK</b>	Master Backup Key
<b>MKEK</b>	Master Key Encryption Key
<b>PCIe</b>	PCI Express Interface
<b>URL</b>	Uniform Resource Locator

Table 2: Abbreviations

## 2 Overview

### 2.1 OpenStack Barbican

Barbican is the OpenStack Key Manager service. It provides secure storage, provisioning, and management of secret data. This includes keying material such as Symmetric Keys, Asymmetric Keys, Certificates, and raw binary data.

### 2.2 Utimaco u.trust General Purpose HSM Se-Series

Utimaco u.trust General Purpose HSM Se-Series is a hardware security module developed by Utimaco IS GmbH. It is a physically protected, specialized computer unit designed to perform sensitive cryptographic tasks and securely manage and store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

### 3 Integration Requirements and Prerequisites

Ensure that the system environment you will be using meets the following hardware and software requirements.

This guide assumes that the user has already installed and configured the required Software.

#### 3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM and OpenStack Barbican:

<b>Operating System</b>	<b>OpenStack Version</b>	<b>Utimaco Security Server Version</b>	<b>Utimaco HSM</b>
RHEL 9	OpenStack Dalmatian	SecurityServer 6.1.1	u.trust General Purpose HSM Se-Series

Table 3: List of Tested Versions

#### 3.2 Software Requirements

<b>Software</b>	<b>Software Requirements</b>
HSM Interfaces	PKCS11

Table 4: List of Software Requirements

#### 3.3 Hardware Requirements

<b>Hardware</b>	<b>Hardware Requirements</b>
Utimaco LAN HSM	u.trust General Purpose HSM Se-Series LAN with firmware SecurityServer 6.1.1

Hardware	Hardware Requirements
Utimaco PCI-e HSM	u.trust General Purpose HSM Se-Series PCIe card with firmware SecurityServer 6.1.1

Table 5: List of Hardware Requirements

### 3.4 Prerequisites

Before you begin, please ensure that you have installed/setup:

- SecurityServer is set up and configured. Refer to the SecurityServer documentation to set up the HSM.
- MBK must be created and stored on each HSM. Refer to the SecurityServer documentation to set up the MBK.
- SecurityServer Default Admin should be replaced with a new admin user.
- Operating system listed in [Tested Versions](#).
- SecurityServer listed in [Tested Version](#).
- An admin user is required to install software.

## 4 Installing and Configuring Utimaco SecurityServer Software

### 4.1 Download and Install Utimaco SecurityServer Software

1. Copy the downloaded software to the appropriate location on the OpenStack Barbican Server.
2. Create a utimaco folder under /opt directory and further create 2 directories /opt/utimaco/bin and /opt/utimaco/lib

```
>_ Console
```

```
# mkdir -p /opt/utimaco/bin  
# mkdir /opt/utimaco/lib
```

3. Copy pkcs11 library file libcs\_pkcs11\_R3.so from Utimaco SecurityServer software to the /opt/utimaco/lib directory

```
>_ Console
```

```
# cp ~/path_to_application_folder/lib/libcs_pkcs11_R3.so /opt/utimaco/lib
```

4. Copy the csadm and p11tool2 files from Utimaco SecurityServer software to /opt/utimaco/bin directory and make both the files executable.

```
>_ Console
```

```
# cd ~/path_to_application_folder/Software/Linux/Administration

# cp csadm p11tool2 gladm /opt/utimaco/bin

# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2 /opt/utimaco/
bin/gladm

# chmod +x /opt/utimaco/lib/libcs_pkcs11_R3.so
```

## 4.2 SecurityServer PKCS#11 Configuration

1. Create the directory /opt/utimaco/lib. Locate the Utimaco PKCS#11 configuration file in your SecurityServer directory, Linux/x86-64/Crypto\_APIS/PKCS11\_R3/sample. Copy the Utimaco PKCS#11 configuration file p11\_R3.cfg into /opt/utimaco/lib directory.

```
>_ Console
```

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/Crypto_APIS/PKCS11_R3/sample

# cp cs_pkcs11_R3.cfg /etc/

# cd /etc
```

2. Edit the cs\_pkcs11\_R3.cfg file and make the appropriate changes to the file.

```
cs_pkcs11_R3.cfg
```

```
[Global]

# For unix:

Logpath = /tmp

# LogLevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)

Logging = 4

Keepalive = true

# Set the Device to connect with

[HSMCluster]

# Device specifier

Device = <HSM_IP>
```



*For more information regarding the commands and command parameters, please see the Utimaco SecurityServer documentation. The device may be a SecurityServer (PCIe or LAN) device. The device line will follow one of these patterns, based on the HSM form-factor:*

*Device = 288@<HSM IP address> Hardware (LAN) HSM*

*OR*

*Device = /dev/cs2.0.1 (last number represents cHSM slot (1)) PCIe card*



*To make your testing easier, enable the PKCS#11 log file. You can enable it by editing the logging LogLevel. Set the LogPath and LogLevel to 1. For testing, you may want to increase them to 4.*

*The added LogPath points to a writable directory, not to a file.*

*If you encounter problems, check the log file named cs\_pkcs11\_R3.log in the under LogPath defined directory. When you are done testing, you should change Logging to 1 or 2. This will limit the logging to only critical and important messages.*

### 4.3 Create SO User and Initialize a Slot

You must initialize a slot with a custom label using p11tool2.

First, using p11tool2, create the SO or Security Officer, and then, using the p11tool2 command, initialize the slot you want to use and the slot user, as shown below.

```
>_ Console
```

```
# export CS_PKCS11_R3_CFG="/etc/cs_pkcs11_R3.cfg"
```

```
# ./p11tool2 [Slot=<slot_id>] [Label=<label>] \
```

```
    [Force=<force>] [Login=<admin_name>,<admin_auth_token>] \
```

```
    InitToken=<so_pin>
```

```
# ./p11tool2 [Slot=<slot_id>] LoginSO=12345678 setpin=12345678,123456789
```

```
# ./p11tool2 slot=<slot_no> LoginSO=12345678 InitPin=123456789
```

```
# ./p11tool2 slot=<slot_id> Login=123456789 setpin=123456789,1234567890
```

## 5 (Optional) Installing OpenStack Barbican



*Skip this section if OpenStack Barbican has already been installed. The steps below are only for demonstration purposes and will change based on the actual requirement.*

### 5.1 Install OpenStack Barbican

Use the steps below to install OpenStack Dalmatian via Packstack.

1. Disable SELinux using the command below:

```
> _ Console
```

```
# setenforce 0
```

Alternatively, open the `/etc/selinux/config` file and change `permissive` to `disabled` as shown below. Then reboot your machine.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figure 1 : SELINUX Config File



*This is for demonstration purposes only. Add the OpenStack Barbican files and directories in the SELinux context in a production environment.*

2. Change the hostname to `controller` by using the below command and reboot.

```
> _ Console
```

```
# hostnamectl set-hostname controller  
  
# reboot
```

3. Map barbican and controller to IP in /etc/hosts file.

```
>_ Console  
  
# vi /etc/hosts  
  
<Host_IP>    barbican    barbican.localdomain  
  
<Host_IP>    controller  controller.localdomain
```

```
GNU nano 5.6.1  
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
127.28.14.25  barbican    barbican.localdomain  
127.28.14.25  controller  controller.localdomain
```

Figure 2 : Hosts File

4. Stop the firewall and NetworkManager using the below command.



*This is for demonstration purpose only. In production environment allow openstack barbican services through firewall.*

```
>_ Console  
  
# systemctl stop firewalld NetworkManager  
  
# systemctl disable firewalld NetworkManager
```

5. Set up the timezone.

```
>_ Console
```

```
# timedatectl set-timezone <timezone name>
```

6. Enable the required repositories.

```
>_ Console
```

```
# subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms --  
enable=rhel-9-for-x86_64-appstream-rpms --enable=rhel-9-for-x86_64-  
supplementary-rpms --enable=codeready-builder-for-rhel-9-x86_64-rpms
```

7. Install OpenStack Packstack by using the following commands.

```
>_ Console
```

```
# dnf install https://trunk.rdoproject.org/rdo_release/rdo-  
release.el9s.rpm
```

```
# dnf upgrade
```

```
# dnf install -y openstack-packstack
```

8. Install the OpenStack instance.

```
>_ Console
```

```
# packstack --allinone
```

```
[root@controller ~]# packstack --answer-file 24.11.22.conf
Welcome to the Packstack setup utility

The installation log file is available at: /var/tmp/packstack/20221129-114039-ygh54ih1/openstack-setup.log

Installing:
Clean Up [ DONE ]
Discovering ip protocol version [ DONE ]
Setting up ssh keys [ DONE ]
Preparing servers [ DONE ]
Pre installing Puppet and discovering hosts' details [ DONE ]
Preparing pre-install entries [ DONE ]
Setting up CACERT [ DONE ]
Preparing AMQP entries [ DONE ]
Preparing MariaDB entries [ DONE ]
Fixing Keystone LDAP config parameters to be undef if empty [ DONE ]
Preparing Keystone entries [ DONE ]
Preparing Glance entries [ DONE ]
Checking if the Cinder server has a cinder-volumes vg [ DONE ]
Preparing Cinder entries [ DONE ]
Preparing Nova API entries [ DONE ]
Creating ssh keys for Nova migration [ DONE ]
Gathering ssh host keys for Nova migration [ DONE ]
Preparing Nova Compute entries [ DONE ]
Preparing Nova Scheduler entries [ DONE ]
Preparing Nova VNC Proxy entries [ DONE ]
Preparing OpenStack Network-related Nova entries [ DONE ]
Preparing Nova Common entries [ DONE ]
Preparing Neutron API entries [ DONE ]
Preparing Neutron L3 entries [ DONE ]
Preparing Neutron L2 Agent entries [ DONE ]

Preparing Neutron DHCP Agent entries [ DONE ]
Preparing Neutron Metering Agent entries [ DONE ]
Checking if NetworkManager is enabled and running [ DONE ]
Preparing OpenStack Client entries [ DONE ]
Preparing Horizon entries [ DONE ]
Preparing Swift builder entries [ DONE ]
Preparing Swift proxy entries [ DONE ]
Preparing Swift storage entries [ DONE ]
Preparing Gnocchi entries [ DONE ]
Preparing Redis entries [ DONE ]
Preparing Ceilometer entries [ DONE ]
Preparing Aodh entries [ DONE ]
Preparing Puppet manifests [ DONE ]
Copying Puppet modules and manifests [ DONE ]
Applying 172.23.0.80_controller.pp [ DONE ]
172.23.0.80_controller.pp: [ DONE ]
Applying 172.23.0.80_network.pp [ DONE ]
172.23.0.80_network.pp: [ DONE ]
Applying 172.23.0.80_compute.pp [ DONE ]
172.23.0.80_compute.pp: [ DONE ]
Applying Puppet manifests [ DONE ]
Finalizing [ DONE ]

**** Installation completed successfully ****
```

Figure 3 : PackStack Configuration

- An answer file will be generated in /root/. This file contains the passwords automatically generated by Packsack and used by MariaDB and other services.

9. Open mysql as root.

```
> _ Console
```

```
# mysql -u root -p
```

10. Create a database Barbican and allow full permission to the Barbican user.

```
>_ Console

> CREATE DATABASE barbican;

> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'localhost' IDENTIFIED
BY '123456';

>GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'%' IDENTIFIED BY
'123456';

> exit;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit;
Bye
[root@controller ~(keystone_admin)]#
```

Figure 4 : Mysql command output

11. Restart the http daemon:

```
>_ Console

# systemctl restart httpd
```

12. Set the environment variable for OpenStack.

```
>_ Console
```

```
# source /root/keystonerc_admin
```

13. Create a user barbican in OpenStack.

```
> _ Console openstack
```

```
# openstack user create --domain default --password 123456 barbican
```

```
[root@controller user(keystone_admin)]# openstack user create --domain default --password 123456 barbican
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+
| Field | Value |
+-----+-----+
| default_project_id | None |
| domain_id | default |
| email | None |
| enabled | True |
| id | c6426a7dbe1345bfa67f8d7db4fdcc2e |
| name | barbican |
| description | None |
| password_expires_at | None |
+-----+-----+
```

14. Add the admin role to the user barbican in project services.

```
> _ Console
```

```
# openstack role add --project services --user barbican admin
```

15. Add the creator role.

```
> _ Console
```

```
# openstack role create creator
```

```
[root@controller user(keystone_admin)]# openstack role create creator
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+
| Field | Value |
+-----+-----+
| id     | 4640be74da5441788609bfeacea5a0dc |
| name   | creator |
| domain_id | None |
| description | None |
+-----+-----+
```

Figure 5 : OpenStack Role Creation

16. Add the creator role to the barbican user.

```
>_ Console

# openstack role add --project services --user barbican creator

[root@controller bin(keystone_admin)]# openstack role add --project services --user barbican creator
[root@controller bin(keystone_admin)]# █
```

Figure 6 : Add Creator Role

17. Create the barbican service entities.

```
>_ Console

# openstack service create --name barbican --description "key manager" key-
manager

[root@controller user(keystone_admin)]# openstack service create --name barbican --description "key manager" key-manager
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+
| Field | Value |
+-----+-----+
| id     | f11eda1aee1248bc95789d7142b847e9 |
| name   | barbican |
| type   | key-manager |
| enabled | True |
| description | key manager |
+-----+-----+
```

Figure 7 : OpenStack Service Creation

18. Create the Key Manager service API public endpoint.

```
>_ Console
```

```
# openstack endpoint create --region RegionOne key-manager public http://
172.28.14.25:9311
```

```
[root@controller ~](keystone_admin)# openstack endpoint create --region RegionOne key-manager public http://controller:9311
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 78b2f48139914f659f4c6216dac6cd27 |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 856011803d674e678cc15e7f3caf94ad |
| service_name | barbican |
| service_type | key-manager |
| url | http://controller:9311 |
+-----+-----+
```

Figure 8 : OpenStack Public Endpoint Creation

19. Create the Key Manager service API Internal endpoint.

```
>_ Console
```

```
# openstack endpoint create --region RegionOne key-manager internal http://
172.28.14.25:9311
```

```
[root@controller ~](keystone_admin)# openstack endpoint create --region RegionOne key-manager internal http://controller:9311
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 0c755c2012c748a6ac64047794051507 |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 856011803d674e678cc15e7f3caf94ad |
| service_name | barbican |
| service_type | key-manager |
| url | http://controller:9311 |
+-----+-----+
```

Figure 9 : OpenStack Internal Endpoint Creation

20. Create the Key Manager service API Admin endpoint.

```
>_ Console
```

```
# openstack endpoint create --region RegionOne key-manager admin http://
172.28.14.25:9311
```

```
[root@controller ~](keystone_admin)# openstack endpoint create --region RegionOne key-manager admin http://controller:9311
```

Field	Value
enabled	True
id	93e0798fef324e61a8a262bb683d7947
interface	admin
region	RegionOne
region_id	RegionOne
service_id	856011803d674e678cc15e7f3caf94ad
service_name	barbican
service_type	key-manager
url	http://controller:9311

Figure 10 : OpenStack Admin Endpoint Creation

21. List the Key Manager service API endpoints created.

```
>_ Console
```

```
# openstack endpoint list --service barbican
```

```
[root@controller ~](keystone_admin)# openstack endpoint list --service barbican
```

ID	Region	Service Name	Service Type	Enabled	Interface	URL
0c755c2012c748a6ac64047794051507	RegionOne	barbican	key-manager	True	internal	http://controller:9311
78b2f48139914f659f4c6216dac6cd27	RegionOne	barbican	key-manager	True	public	http://controller:9311
93e0798fef324e61a8a262bb683d7947	RegionOne	barbican	key-manager	True	admin	http://controller:9311

Figure 11 : Endpoint List

22. Install OpenStack Barbican:

```
>_ Console
```

```
# dnf install -y openstack-barbican
```

23. Add the below information to the /etc/barbican/barbican.conf file. You can replace the username and password with the ones you set earlier during installation.

```
>_ Console
```

```
# vi /etc/barbican/barbican.conf

[DEFAULT]

host_href = http://barbican:9311

debug = true

transport_url = rabbit://guest:guest@controller

log_file = /tmp/logfile

[keystone_authtoken]

www_authenticate_uri = http://controller:5000

auth_url = http://controller:5000/v3

auth_version = v3

insecure = true

region_name = RegionOne

memcached_servers = localhost:11211

auth_type = password

project_name = services

user_domain_id = default

project_domain_id = default

username = barbican

password = 123456

[database]

connection = mysql+pymysql://barbican:123456@localhost/barbican
```

24. Populate the Key Manager service database.

```
>_ Console

# su -s /bin/sh -c "barbican-manage db upgrade" barbican

[root@controller barbican]# su -s /bin/sh -c "barbican-manage db upgrade" barbican
2022-11-30 11:45:46.872 727099 WARNING oslo_db.sqlalchemy.engines [-] MySQL SQL mode is 'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION', consider enabling TRADITIONAL or STRICT_ALL_TABLES
2022-11-30 11:45:46.875 727099 INFO alembic.runtime.migration [-] Context impl MySQLImpl.
2022-11-30 11:45:46.875 727099 INFO alembic.runtime.migration [-] Will assume non-transactional DDL.
2022-11-30 11:45:46.899 727099 INFO alembic.runtime.migration [-] Running upgrade → 39cf2e645cba, 0cata rebase
2022-11-30 11:45:48.081 727099 INFO alembic.runtime.migration [-] Running upgrade 39cf2e645cba → 0f8c192a061f, Add Secret Consumers table
[root@controller barbican]#
```

Figure 12 : Database Upgrade

25. Restart Barbican API service.

```
>_ Console

# systemctl restart openstack-barbican-api.service
```

26. Restart and enable the httpd service.

```
>_ Console

# systemctl restart httpd

# systemctl enable httpd
```

## 6 Integrating OpenStack Barbican with Utimaco SecurityServer

### 6.1 Configure Barbican to use Utimaco HSM

1. Add the below information to the barbican.conf file.

```
> _ Console
```

```
# vi /etc/barbican/barbican.conf

[secretstore]

namespace = barbican.secretstore.plugin enabled_secretstore_plugins =
store_crypto

[crypto]

enabled_crypto_plugins = p11_crypto

[crypto provjeriti]

enabled_crypto_plugins = simple_crypto
[simple_crypto_plugin]

kek = REPLACE_WITH_KEY
[p11_crypto_plugin]
# Path to Utimaco PKCS11 library

library_path = /opt/utimaco/lib/libcs_pkcs11_R3.so
# CryptoUser PIN to login to PKCS11

login = <PKCS11 Slot User PIN>
# Master KEK label as stored in the HSM

mkek_label = mkek_utimaco
# Master KEK length in bytes. (integer value)

mkek_length = 32
# Master HMAC Key label (as stored in the HSM) (string value)

hmac_label = hmac_utimaco
# HSM Slot ID (integer value)

slot_id = 3

encryption_mechanism = CKM_AES_CBC
```



*mkek\_utimaco and hmac\_utimaco keys will be generated on the Utimaco HSM in slot 3 in the next section of this document.*

## 6.2 Generating MKEK and HMAC Key on Utimaco HSM

1. Generate the MKEK using the below command.

```
>_ Console

# chmod 666 /tmp/cs_pkcs11_R3.log
# su -s /bin/sh -c "barbican-manage hsm gen_mkek --library-path '/opt/
utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890 --slot-id 3 --label
'mkek_utimaco' --length 32" barbican

[root@controller bin(keystone_admin)]# barbican-manage hsm gen_mkek --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphr
ase '123456' --slot-id 1 --label 'mkek_utimaco' --length 32
MKEK successfully generated!
```

Figure 13 : MKEK Key Generation

2. Generate the HMAC Key using the below command.

```
>_ Console

# su -s /bin/sh -c "barbican-manage hsm gen_hmac --library-path '/opt/
utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890 --slot-id 3 --
label 'hmac_utimaco' --length 32" barbican

[root@controller bin(keystone_admin)]# barbican-manage hsm gen_hmac --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphr
ase '123456' --slot-id 1 --label 'hmac_utimaco' --length 32
HMAC successfully generated!
```

Figure 14 : HMAC Key Generation

3. Verify that the keys are generated on the Utimaco HSM using the p11tool2 command.

```
>_ Console

#./p11tool2 slot=<slot_id> LoginUser=<Crypto_User_PIN> ListObjects
```

4. Restart the OpenStack-Barbican-API and httpd services.

```
>_ Console
```

```
# systemctl restart openstack-barbican-api.service
```

```
# systemctl restart httpd
```

## 6.3 Encrypting and Decrypting Secrets

1. Create a secret or password.

```
>_ Console
```

```
# openstack secret store --name utimaco123 --payload password
```

```
[root@rhelserver user(keystone_admin)]# openstack secret store --name utimaco123 --payload password
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
```

Field	Value
Secret href	<a href="http://172.28.14.25:9311/v1/secrets/20d68ada-e4fd-4e33-9f6b-e08927ab5d8f">http://172.28.14.25:9311/v1/secrets/20d68ada-e4fd-4e33-9f6b-e08927ab5d8f</a>
Name	utimaco123
Created	None
Status	None
Content types	None
Algorithm	aes
Bit length	256
Secret type	opaque
Mode	cbc
Expiration	None

Figure 15 : Creating a secret with OpenStack Barbican

Here utimaco123 is the secret name and its value is password. This secret is stored in an encrypted form in openstack barbican.

2. You can also verify the encryption operation logging in PKCS11 log file cs\_pkcs11\_R3.log during secret generation as shown below.

```
[root@rhelserver user(keystone_admin)]# tail /tmp/cs_pkcs11_R3.log
03.06.2025 14:15:24.606 | [01885307:01885307] C_GetSessionInfo | T: enter C_GetSessionInfo(hSession: 0x00000004)
03.06.2025 14:15:24.607 | [01885307:01885307] C_GetSessionInfo | T: leave C_GetSessionInfo()
03.06.2025 14:15:24.607 | [01885307:01885307] C_GenerateRandom | T: enter C_GenerateRandom(hSession: 0x00000004, pRandomData: CK_BYTE[10])
03.06.2025 14:15:24.607 | [01885307:01885307] C_GenerateRandom | T: leave C_GenerateRandom()
03.06.2025 14:15:24.607 | [01885307:01885307] C_EncryptInit | T: enter C_EncryptInit(hSession: 0x00000004, pMechanism: CKM_AES_CBC (0x1082), hKey: 0x00000003)
03.06.2025 14:15:24.607 | [01885307:01885307] C_EncryptInit | T: leave C_EncryptInit()
03.06.2025 14:15:24.607 | [01885307:01885307] C_Encrypt | T: enter C_Encrypt(hSession: 0x00000004)
03.06.2025 14:15:24.608 | [01885307:01885307] C_Encrypt | T: leave C_Encrypt()
03.06.2025 14:15:24.608 | [01885307:01885307] C_CloseSession | T: enter C_CloseSession(hSession: 0x00000004)
03.06.2025 14:15:24.608 | [01885307:01885307] C_CloseSession | T: leave C_CloseSession()
```

Figure 16 : pkcs#11 Logs

3. Fetch the secret that was created without its value.

```
>_ Console

# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce4cae-
b1d9-a2a818ea30a0
```

```
[root@rhelserver user(keystone_admin)]# openstack secret get http://172.28.14.25:9311/v1/secrets/a2e72976-914e-4ede-a230-02d14629603e
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+
| Field | Value |
+-----+-----+
| Secret href | http://172.28.14.25:9311/v1/secrets/a2e72976-914e-4ede-a230-02d14629603e |
| Name | utimacoTestNew |
| Created | 2025-06-03T12:15:24+00:00 |
| Status | ACTIVE |
| Content types | {'default': 'application/octet-stream'} |
| Algorithm | aes |
| Bit length | 256 |
| Secret type | opaque |
| Mode | cbc |
| Expiration | None |
+-----+-----+
```

Figure 17 : Get secret without payload

4. Fetch the secret that was created with its value.

```
>_ Console

# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-
4cae-b1d9-a2a818ea30a0 --payload
```

```
[root@rhelserver user(keystone_admin)]# openstack secret get http://172.28.14.25:9311/v1/secrets/a2e72976-914e-4ede-a230-02d14629603e --payload
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+
| Field | Value |
+-----+
| Payload | password |
+-----+
```

Figure 18 : Get secret with payload

The secret is decrypted first and displayed.

5. You can also verify the decryption operation logging in PKCS11 log file cs\_pkcs11\_R3.log during secret retrieval as shown below.

```
05.01.2023 10:29:56.105 [00002946:00002946] C_OpenSession T: leave C_OpenSession()
05.01.2023 10:29:56.105 [00002946:00002946] C_GetSessionInfo T: enter C_GetSessionInfo(hSession: 0x00000004)
05.01.2023 10:29:56.105 [00002946:00002946] C_GetSessionInfo T: leave C_GetSessionInfo()
05.01.2023 10:29:56.105 [00002946:00002946] C_DecryptInit T: enter C_DecryptInit(hSession: 0x00000004, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000004)
05.01.2023 10:29:56.105 [00002946:00002946] C_DecryptInit T: leave C_DecryptInit()
05.01.2023 10:29:56.108 [00002946:00002946] C_Decrypt T: enter C_Decrypt(hSession: 0x00000004)
05.01.2023 10:29:56.109 [00002946:00002946] C_Decrypt T: leave C_Decrypt()
05.01.2023 10:29:56.109 [00002946:00002946] C_CloseSession T: enter C_CloseSession(hSession: 0x00000004)
05.01.2023 10:29:56.109 [00002946:00002946] C_CloseSession T: leave C_CloseSession()
```

Figure 19 : pkcs#11 Logs

## 6.4 Generating a symmetric key in Barbican

1. Generate a new 256-bit key using OpenStack order create and store it in Barbican.

```
>_ Console

# openstack secret order create --name app_key --algorithm aes --mode ctr
--bit-length 256 --payload-content-type=application/octet-stream key

[root@rhelserver user(keystone_admin)]# openstack secret order create --name app_key --algorithm aes --mode ctr --bit-length 256 --payload-content-type=application/octet-stream key
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+
| Field | Value |
+-----+
| order href | http://172.28.14.25:9311/v1/orders/42ee7b6d-6c31-49ef-b77d-305ba04a6b79 |
| Type | Key |
| Container href | N/A |
| Secret href | None |
| Created | None |
| Status | None |
| Error code | None |
| Error message | None |
+-----+
```

Figure 20 : Symmetric Key Generation

2. You can also verify the encryption operation logging in the PKCS11 log file cs\_pkcs11\_R3.log during secret generation, as shown below.

```

05.01.2023 10:44:20.024 [00002941:00002941] C_OpenSession T: leave C_OpenSession()
05.01.2023 10:44:20.024 [00002941:00002941] C_GetSessionInfo T: enter C_GetSessionInfo(hSession: 0x00000003)
05.01.2023 10:44:20.024 [00002941:00002941] C_GetSessionInfo T: leave C_GetSessionInfo()
05.01.2023 10:44:20.028 [00002941:00002941] C_GenerateRandom T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[20
])
05.01.2023 10:44:20.028 [00002941:00002941] C_GenerateRandom T: leave C_GenerateRandom()
05.01.2023 10:44:20.031 [00002941:00002941] C_GenerateRandom T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[10
])
05.01.2023 10:44:20.031 [00002941:00002941] C_GenerateRandom T: leave C_GenerateRandom()
05.01.2023 10:44:20.031 [00002941:00002941] C_EncryptInit T: enter C_EncryptInit(hSession: 0x00000003, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000003)
05.01.2023 10:44:20.031 [00002941:00002941] C_EncryptInit T: leave C_EncryptInit()
05.01.2023 10:44:20.034 [00002941:00002941] C_Encrypt T: enter C_Encrypt(hSession: 0x00000003)
05.01.2023 10:44:20.034 [00002941:00002941] C_Encrypt T: leave C_Encrypt()
05.01.2023 10:44:20.034 [00002941:00002941] C_CloseSession T: enter C_CloseSession(hSession: 0x00000003)
05.01.2023 10:44:20.034 [00002941:00002941] C_CloseSession T: leave C_CloseSession()
    
```

Figure 21 : pkcs#11 Logs

- View the details of the order to identify the location of the generated key, which is shown here as the Secret href value.

```

>_ Console

# openstack secret order get http://barbican:9311/v1/orders/fb007366-
1965-49b0-97b6-e1306d103a73

[root@rhelserver user(keystone_admin)]# openstack secret order get http://172.28.14.25:9311/v1/orders/9faf4691-5f71-424c-bd0e-9242bd0ccb3c
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+
| Field | Value |
+-----+-----+
| Order href | http://172.28.14.25:9311/v1/orders/9faf4691-5f71-424c-bd0e-9242bd0ccb3c |
| Type | Key |
| Container href | N/A |
| Secret href | http://172.28.14.25:9311/v1/secrets/9ad5dd8e-d14e-4b4b-a234-a50da3fc81b2 |
| Created | 2025-06-03T12:21:16+00:00 |
| Status | ACTIVE |
| Error code | None |
| Error message | None |
+-----+-----+

[root@rhelserver user(keystone_admin)]# openstack secret order list
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----+-----+-----+-----+-----+-----+-----+-----+
| Order href | Type | Container href | Secret href | Created | Status | Error code | Error message |
+-----+-----+-----+-----+-----+-----+-----+-----+
| http://172.28.14.25:9311/v1/orders/b77ca54d-8f6c-4b54-b33a-75eff312a1c1 | Key | N/A | http://172.28.14.25:9311/v1/secrets/b01b087b-bec8-424b-bddd-d4cfb6474c0b | 2025-06-03T12:19:15+00:00 | ACTIVE | None | None |
| http://172.28.14.25:9311/v1/orders/42ee7b6d-6c31-49ef-b77d-305ba04a6b79 | Key | N/A | http://172.28.14.25:9311/v1/secrets/363e8da5-0d63-413f-b3c5-297dca051274 | 2025-06-03T12:20:37+00:00 | ACTIVE | None | None |
| http://172.28.14.25:9311/v1/orders/9faf4691-5f71-424c-bd0e-9242bd0ccb3c | Key | N/A | http://172.28.14.25:9311/v1/secrets/9ad5dd8e-d14e-4b4b-a234-a50da3fc81b2 | 2025-06-03T12:21:16+00:00 | ACTIVE | None | None |
+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

Figure 22 : Details of the Order and the order list

- Retrieve the details of the secret.

```

>_ Console
    
```

```
# openstack secret get http://barbican:9311/v1/secrets/731068e6-87834efc-b937-34002224aff6
```

```
[root@controller bin(keystone_admin)]# openstack secret get http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6
```

Field	Value
Secret href	http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6
Name	app_key
Created	2022-12-12T11:54:44+00:00
Status	ACTIVE
Content types	{'default': 'application/octet-stream'}
Algorithm	aes
Bit length	256
Secret type	symmetric
Mode	ctr
Expiration	None

Figure 23 : Retrieve the secret details

- Alternatively, you can list the symmetric key that has been generated by the command below.

```
> _ Console
```

```
# openstack secret list
```

```
[root@controller bin(keystone_admin)]# openstack secret list
```

Secret href	Name	Created	Status	Content
http://barbican:9311/v1/secrets/8ac7918d-36ce-4cae-b1d9-a2a818ea30a0	utimaco123	2022-12-06T06:38:56+00:00	ACTIVE	{'default': 'application/octet-stream'}
http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6	app_key	2022-12-12T11:54:44+00:00	ACTIVE	{'default': 'application/octet-stream'}

Figure 24 : Secret listing

## 6.5 Storing Public Key, Private Key and Certificate in OpenStackBarbican

- Create a self-signed certificate using the command below.

```
> _ Console
```

```
# openssl req -x509 -newkey rsa:4096 -keyout private.pem -out cert.pem
-sha256 -days 365 -nodes

[root@controller ~(keystone_admin)]# openssl req -x509 -newkey rsa:4096 -keyout private.pem -out cert.pem -sha256 -days 365 -nodes
Generating a RSA private key
.....++++
writing new private key to 'private.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:IN
State or Province Name (full name) []:MH
Locality Name (eg, city) [Default City]:Pune
Organization Name (eg, company) [Default Company Ltd]:utimaco.com
Organizational Unit Name (eg, section) []:utimaco
Common Name (eg, your name or your server's hostname) []:test.utimaco.com
Email Address []:support@utimaco.com
```

Figure 25 : Create a self-signed certificate



You can generate a key and a certificate by using other utilities as well.

2. Verify that the private key and certificate file are generated.

```
>_ Console

# ls

[root@controller ~(keystone_admin)]# ls
24.11.22.conf cert.pem cxl.log key.pem keystonec_admin private.pem public.pem
[root@controller ~(keystone_admin)]# █
```

Figure 26 : File listing

3. Generate a public key from a private key.

```
[root@controller ~]# ll
total 84
-rw----- 1 root root 51310 Dec  2 16:02 24.11.22.conf
-rw-r--r-- 1 root root  2143 Dec 12 18:29 cert.pem
-rw-r--r-- 1 root root  9792 Dec  2 10:23 cxi.log
-rw----- 1 root root  3272 Dec 12 18:18 key.pem
-rw----- 1 root root   361 Nov 29 11:48 keystone_admin
-rw----- 1 root root  3272 Dec 12 18:28 private.pem
-rw-r--r-- 1 root root   800 Dec 12 18:29 public.pem
[root@controller ~]#
```

Figure 27 : Creation of Public Key

4. Store the public key in OpenStack Barbican.

```
>_ Console

# openssl rsa -in private.pem -out public.pem -pubout

# openstack secret store --algorithm rsa --secret-type public --payload-
dcontent-type application/octet-stream --payload-content-encoding base64 -
-payload "$(base64 < public.pem)" --bit-length 2048 --name pubtest

[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --secret-type public --payload-content-type applicatio
n/octet-stream --payload-content-encoding base64 --payload "$(base64 < public.pem)" --bit-length 2048 --name pubtest
+-----+-----+
| Field | Value |
+-----+-----+
| Secret href | http://barbican:9311/v1/secrets/85c202ff-05b6-4923-9ce6-a6cee3cde2d9 |
| Name | pubtest |
| Created | None |
| Status | None |
| Content types | {'default': 'application/octet-stream'} |
| Algorithm | rsa |
| Bit length | 2048 |
| Secret type | public |
| Mode | cbc |
| Expiration | None |
+-----+-----+
```

Figure 28 : Store Public Key in Openstack Barbican

5. You can also verify the encryption operation logging in the PKCS11 log file cs\_pkcs11\_R3.log during public secret generation, as shown below.

```

05.01.2023 11:05:46.315 [00002944:00002944] C_OpenSession          T: leave C_OpenSession()
05.01.2023 11:05:46.315 [00002944:00002944] C_GetSessionInfo      T: enter C_GetSessionInfo(hSession: 0x00000003)
05.01.2023 11:05:46.316 [00002944:00002944] C_GetSessionInfo      T: leave C_GetSessionInfo()
05.01.2023 11:05:46.319 [00002944:00002944] C_GenerateRandom      T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[10
])
05.01.2023 11:05:46.319 [00002944:00002944] C_GenerateRandom      T: leave C_GenerateRandom()
05.01.2023 11:05:46.319 [00002944:00002944] C_EncryptInit         T: enter C_EncryptInit(hSession: 0x00000003, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000003)
05.01.2023 11:05:46.319 [00002944:00002944] C_EncryptInit         T: leave C_EncryptInit()
05.01.2023 11:05:46.322 [00002944:00002944] C_Encrypt             T: enter C_Encrypt(hSession: 0x00000003)
05.01.2023 11:05:46.322 [00002944:00002944] C_Encrypt             T: leave C_Encrypt()
05.01.2023 11:05:46.323 [00002944:00002944] C_CloseSession       T: enter C_CloseSession(hSession: 0x00000003)
05.01.2023 11:05:46.323 [00002944:00002944] C_CloseSession       T: leave C_CloseSession()
    
```

Figure 29 : pkcs#11 Logs

6. Get the value of the public key.

```

> _ Console

# openstack secret get -p -c Payload -f value
http://barbican:9311/v1/secrets/85c202ff-05b6-4923-9ce6-a6cee3cde2d9

[root@controller ~](keystone_admin)# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/85c202ff-05b6-4923-9ce6-a6cee3cde2d9
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAGEAyoyaYwpbZSLC9+xCvUWpN
RpTxBiuf2ln2L0rGmxTfrK8se3CX0n3ZvMHsaBDrKfi6hNF0ppgaGLIBQ/F85IE1
hVxjVHBS7NYEFZ11VG6pHi00WmCP1W0GvoZBPSHwYuq2Mcib78PaIC3Yn/IjASy9
BcUrD2JTaZ2YppDHgbGuWd7/mmc9TKB0n5M5SFiqfjNXeHvIYzlvxg4teWMrdGdT
v11IzJqHFFLIncQ0ZbkMS0ZoobsDW9suNC7oyeKMX01LqWib1sT2PfTYXcYLZG4q
/3ogZ69Doox8T/RjFKRlq0Jvj47EzmAoJs9ZkZMA5rAdN/rnkEnB6E/UPWrhSv
yqJWtIrmJ501sFRWHfV0R55KXjgsqKKK+LCFMVZB//uzV+KiJTD9e68+8tknx0Y
0GSX/gk2xk1HAupq5Iae7/7KDGruWpS/JcDw7o4zDZSgIhZZ1wlv735fHDn+o01U
818u0gCKHR0fCaGcvYhaGD1JBNx5dH9VxWUxMBzFrdrMIPm0SnA0s+TgAg+Fw8F+
CMAR0UfC3U4oGu6ibtVYi/QAzxaRlzMaw712EAN5QH7dYhSpH3jNwmhfvXaXB55d
sw45fjWB7R9Dz/pVhva608s4RX0Bf0zDMg6oNsrzI8pr8gyNSFT2MvzTguIv66IU
Isy/c8KSay/t3QbXt0j9hsUCAwEAAQ=
-----END PUBLIC KEY-----
    
```

Figure 30 : Get Public Key from Openstack Barbican

7. Store the Private key in OpenStack Barbican.

```

> _ Console

# openstack secret store --algorithm rsa --secret-type private --payload-
content-type application/octet-stream --payload-content-encoding base64 -
-payload "$(base64 < private.pem)" --bit-length 2048 --name privatekeytest
    
```

```
[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --secret-type private --payload-content-type application/octet-stream --payload-content-encoding base64 --payload "$(base64 < private.pem)" --bit-length 2048 --name privatekeytest
```

Field	Value
Secret href	http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4c-9cb3-2caebfb88b11
Name	privatekeytest
Created	None
Status	None
Content types	{'default': 'application/octet-stream'}
Algorithm	rsa
Bit length	2048
Secret type	private
Mode	cbc
Expiration	None

Figure 31 : Store Private Key in Openstack Barbican

8. Get the value of the Private key.

```
>_ Console

# openstack secret get -p -c Payload -f value

http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4c-9cb3-2caebfb88b11
```

```
[root@controller ~](keystone_admin)# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4c-9cb3-2caebfb88b11
-----BEGIN PRIVATE KEY-----
MIIEJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkqAgEAAoICAQDKhpjCltlIsL37
EK9Rak1GLPEGK7R/awfYs6sabFN+sryx7cJfSfdnBYexoE0sp+LqE0XSmbBoaUgFD
8XzkgTWFxGmUcFLs1gQVnWVUbqkELTRYxw/VY4a+hkE9I fBi6rYxyIHvw9ogLdif
8iMBLL0fXsSPYLNpnZtmkMeBsa5Z3v+aZz1MoHSfkzliWkp+M1d4e8h jOW/GD115
Yyt0Z10/WUjMnocUUsidxA5LuQxLm thuwNb2y40Lu jJ4oxFTUupYhvWxPY99Nhd
xgtkbir/eIbnr00ijHxP9GmUpGWRQkm+PjsTOYCGmz1mRkwdmsB03+ueR0ScHoT9
Q9auFKJ2o1a0iuYnnSwwFYd9XRhnkpeOCyooor6UIUxVkh/+7NX4qILMP17rz7
y2Sf5jQZJf+CTbGTUC6mrkhp7v/soMatTCLL8lwNbu j jMNLKAIFmXWcv/vf18c
0f6g6VtzXy46AIdE58JoZy9iFoYUkE3Hl0f1XFZTEwHmMt2swg+bRkCA6z50AC
D4XdwX4IwBHRRLdtiga7qJu1vIL9ADPFpGXmxbvXYQA3lAft1lFKkfeM3Cef+9
dpcHLJ2zDjL+NYHtH0Nn+LWG9rrTyzhFc4EU7MMYDqg2yvmvYDI1IIVPy/NOC
4i/rohQizL9zwpJrL+3dBte06P2GxQIDAQABAoICACaUJR1MJU/V2xqsPvk/SqEb
Vh5azPjXOPGtd0+rVkBKzlpLdEZAeu5/fM0GDAXev4j1bUcDdzfSIZrJeNRSA5x
ykoZC3LaAcUsSlUfD1AURFh08usvHkfeW0C16mitVS9+hmHp608gTJelRGfA4fDr
chxoGo0P5g09AwckJ3GAOKkr+Sc3BpZrRu+6BtKJGtrC0z9z/DnWns5DNolpMra
eaeF+ANDEbwb0LJH4b01IEFNAoLI6ba/D1EFur+ctcD6U1tCABYjoeObc1LDY+Vu
m1WmGTgpr+SvVRdlYhqBcodKJUs5HezmqqUOX40SIL69KMD6E/k8GheN9LJvLC
wlogv930300AE4pI8wvXigPMLNBESgQlAPCJhsGJlQ7kw++4SYGV10BR+gx/DJ7S
kMv44vUgxf1YREJR5RftwNatTQakb/vognZaRsJvPrb8LAgTYm5nEvEjcbCpnEq
eX9ELM3K98agTjC6BVuCZ+FrhERb6mq0jCERtzjgevYCYVSjL+EB0/Pj/7RvDX7
S3shh0ebSeAwyo3FavLqno+05Wypcfm00asfeMb0kctCto4eieHkKhG83ZqnA8p
1UgUgFfnj+Y0dhXFI/tBo05X7gFkxj/ouXka+T25RShyEF+T85cWLD/sFCARyh
LqoUH051cd+tsM3XwEqBAoIBA0QkwwxAlj7mF8GbdGRt+CSLVAxyxqsw5H9c3iW
Gwln/RlFUzJansvY4pXmMQP2L7M3k5oFvMutP6V8R2LgQjQGRqawx7r/fd6yAS
PRhSsmktQ5MqXhBNKIXcLZX11251T4sBaU30qDFism9jkaNGC3KfVGjvKDMAJuk
7LGkBRuMaN7XZLsuxh6cLFVNCblw6cK0Ql1jAdP5R9rE6lhtAT0IXcDnk4us18
9GHQ9ArBUSBlGFe9f6ctR/6rrV0Pob8vbSuobefw7ffdjIR61n7zc4L21jqRW02j
95q8YnFw5Za0WR7XVzCCMwiR1Ub0fBCs3hM6cotVwdGxf3qtAoIBA0Qio9Avq1Jd
7wRlWME0t9gOUHgg86nAKg3q3j1YiYpOTmHLQAWCjFA+2TX8STkT68PhZD7+FTEt
adDLkBTw7outXsa5xymFIVi1FEpT08zfej1USczPPEjzTawGNQ5Uja5LaVGaDn/
Ug4rYvqjvPceop009dlnqNOC1Ynu7gqvD0s7/zRRjnnfGL5qoB7F7NXjZeBAm2/Y
u1Ppfdonv8gmw0t+qA0z1LLMz6VwRt4AEnfnR6E7e6vSk6SblctZQyURnQmXvxQv
mEhr4goD63rWm1ee1/+q8tLy74f/VL4Na8csPzWejwnhwxFoG57ljumgObEdePDN
kIphq+nEcx5AoIBAERwkj7SgQewV6X4N/e35YbsDL+IbnjghsL/w+jsp9DDQbT
k8IKziQ6DeULNfgBtFE65WBAqDBFCNXR28Y6tXwybJhL9Hmgv0Bwct7iBl/fXpYb
LGSws1PSOMPvI30dTj+sJ4LZlVydF6juh2K7wLRPsoTtGLLjixvdZrNd2PugR5Cn
Y2PG5E0RpFkbHqIQr8dWUahda240ebW0b+rIE//Y0QRsk0/U5glx4b5rgsQ37uvN
61pDgSSBhZuFqYyFL1Z38/5KmEumIC9QTW4L35s08iEHXE/QIBxoY827xUzQosr
NPLbeNm2mCS46xGqU60JUh1ihjBElacgc0h8na3UCggEBAJ+SWaCxmGmY0KMFgnF
2WJit+DjU5jwChWgzvReQ662WerL3iGScVejo0PTK0yGB3AF/asJyH0aJo1
Uf0QWdkt6AbRLoW9boNnpLTmnU2+VGzI9lgQJQGN05Qm2VZfq+u6CtJ294iZ3Rn
tNqL/y8kQbHyRavmTYr34R/VCh8JdY3Cs4pTtCGAoS6tvurDhSPHFwM4UthsTrx
Zen0t1Yw/ow1K3TtuNsh0ud0GwYG4feeoKC4AVrmQfGMPApFth5Yk97a6nh0nUjp
pmgVkg0ocI+TpmD7zRYze18a5nY53wfpkFabYKmbZ2f+1XFRkesIeQtTLiU0Xh5K
JokCggEBAIt6n3QlBssMMKfmdS1JG2L3iWfVzCXLKVNza/zqkhr7hv0A4Acc
SBNx08sHtJKrXLRSH6kFu22RFywPk7s1By8rb5jIjNI16esrqsfcXhN4CvldRYKt
Wf1SgqacJAivfcvf9JYfB7ywiBDSXORg2L0qetQy2dXmqmRy+ynGz9dALj5K0
ZMe7Tp7YCLYwYwChG77JlqhrG666zrQk0Nc/2Q8Fdt15s7i0637+eRtPtufy7bE
Bnxbc0Ft1j33pjBPdCfhGLIP1wPjPJSURwU6TjD0B7cTzNn4G+bf/FCBNY6Dh
wULb12pZMGyrnXjr724ETQeP1z4vKc=
-----END PRIVATE KEY-----
```

Figure 32 : Get Private key from Openstack Barbican

9. Store the certificate on OpenStack Barbican.

```
> _ Console

# openstack secret store --algorithm rsa --bit-length 2048 --secret-type
certificate --payload-content-type application/octet-stream --payload-
content-encoding base64 --payload "$(base64 < cert.pem)" --name
certificatetestutimaco
```

```
[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --bit-length 2048 --secret-type certificate --payload
-content-type application/octet-stream --payload-content-encoding base64 --payload "$(base64 < cert.pem)" --name certificatetestu
timaco
+-----+
| Field | Value |
+-----+
| Secret href | http://barbican:9311/v1/secrets/9bba64eb-c875-4d72-9058-2b30fa907c17 |
| Name | certificatetestutimaco |
| Created | None |
| Status | None |
| Content types | {'default': 'application/octet-stream'} |
| Algorithm | rsa |
| Bit length | 2048 |
| Secret type | certificate |
| Mode | cbc |
| Expiration | None |
+-----+
```

Figure 33 : Store certificate in OpenStack Barbican

10. Get the value of the certificate from OpenStack Barbican.

```
>_ Console

# openstack secret get -p -c Payload -f value

http://barbican:9311/v1/secrets/9bba64eb-c875-4d72-9058-2b30fa907c17

[root@controller ~(keystone_admin)]# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/9bba64eb-c875-4d7
2-9058-2b30fa907c17
-----BEGIN CERTIFICATE-----
MIIGATCCA+mgAwIBAgIUJH3q+N40CpMYdwB0n2DHVDuaSRwwDQYJKoZIhvcNAQEL
BQAwY8x CzA JBgNVBAYTAKL0MQswCQYDVQQIDAJNSDENMA sGA1UEBwwEUFVORTEQ
MA4GA1UECgwHVXRpbWVjZjzEQMA4GA1UECwwHVGVzdGluZzEzEcMBoGA1UEAwwTdGVz
dGluZy51dGltYWVlLnVnVbTEiMCAGCSqGSIb3DQEJARYTc3VvcG9ydEB1dGltYWVl
LnVnVbTAeFw0yMjE1MTIwMjE1MTIwMjE1MTIwMjE1MTIwMjE1MTIwMjE1MTIwMjE1
EwJjTjELMAkGA1UECAwCTUgxDTAuBgNVBAMcMBFBVTkUxEDA0BgNVBAoMB1V0aW1h
Y28xEDA0BgNVBAsMB1Rlc3RpbmcuXDAaBgNVBAMHE3Rlc3RpbmudXRpbWVjby5j
b20xIjAgBgqhkiG9w0BCQEW E3N1cHBvcnRAdXRpbWVjby5jb20wggE1iMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQDKhpjCltLIsL37EK9Rak1GLPEGK7R/awfY
s6sabFN+sryx7cJfSfdnBYexoE0sp+LqE0XSmBoaUgFD8XzkgTWFxGnuCFLs1gQV
nWVUbqkelTRYxw/VY4a+hkE9I fB i6rYxyIHvw9ogLd i f8 iMBLL0FxSsPYLNpnZ im
kHeBsa5Z3v+aZz1MoHSfkzLWkp+M1d4e8hjOW/GD i15Yyt0Z10/WUjMmocUU s id
xAS1uQxLRmihuwNb2y40LuJj4oxfTUupYhvwPY99Nhdxtgk b i r / e i B n r 0 0 i j H x P
9GMUpGwrQkm+Pj sTOYcgmz 1mRkwDmsB03+ueR0ScHoT9Q9auFK/J2o1a0 iuYnnSW
wVfYd9XRhnkpe0Cyooor6UIUxVKH/+7NX4qILMP17rz7y25fE5jOZJf+CTbGTUc
6mrkhp7v/soMatTCL8lwNbu j MNLKA iFLmXCW/vfL8c0f6g6VTzXy46AIodE58J
oZy9 iFoYOUKE3Hl0f1XFZTEwHMWt2swg+bRkC A6z50ACD4XDwX4IwBHRR8Ld i ga
7qJ9 iV1e9ADFPpGXmxbvXYQA3Lft1 iFKkfeM3CeF+9dpcHLJ2zDj l+NYHtH0Nn
+LWG9rrTyzhFc4EU7MMyDqg2yvMjymvyDI1IVPYy/NO4 i /rohQ izL9zwpJrL+3d
Bte06P2Gx0IDAQBo1HwUTAdBgNVHQ4EFg0UncuKAgvd4s i f k 6 2 Y g 0 f w 9 C r D z U Q w D w Y D V R 0 T A Q H / B A U w A w E B
/ z A N B g k q h k i G 9 w 0 B A Q s F A A 0 C A g E A D j 6 5 v / A + Y 9 X Y A S u C F 8 G I T K R 3 1 0 0 B O C 2 7 I 4 U g
7 v + G a o a D y n Z s v F + q y a F + D R Y X s 4 h E P u W A z y A 0 + U 3 L P 8 w w k z z U z h g 5 a + j v b T w x S Z L l
x t Q + r C e / i p x 0 K H l c q m v 3 0 G j x p 4 P s + d t z 9 C 2 1 j H w P w 3 e h U a P 0 F y e K P d R V Q x r v b 0 R u
A n P p + d H e R R I o s Q G + d 6 5 1 4 x R C G u v W K S G A 1 k m M q 8 0 F L C Y L 6 1 H j t b Y 1 D z J o h W A Y K C 0 0
z H d + G 4 U N M q h t 2 r r j A j 0 J x o h c 2 5 p q p 7 u B Q A B L d p u D 4 4 0 + / p E 1 b 7 K D P h W 6 S 4 v q 0 1 S
b l F N f 6 J M w 0 J E H + w 1 E q / 6 J v J y y o r i W k z s 0 e T g g b r x L q w 5 6 2 q r 3 7 y E n j j m N P X 8 k u T p 4
1 t K s d z t v 2 2 p B Y D Z + S X J d + a g g A q p S b f z q D k i k i v W Z x A 4 w k y y D h 0 5 H X u Q L f w o m H N t b
0 D e J 1 r 0 Z T d x L M S 0 C o j L W r G q a H m c c e / a 1 + d d 0 a i w b 6 0 E 1 L 0 8 5 5 S C Z P 1 S p o w Z + L N 6 5
3 Z Z j y K v H u 1 H R 3 G W Y o t j e o p g 6 m f y + m A Z V a f + i d n w z w E F 6 2 6 9 P 8 T n b w c y P N F A g K g H E
/ 1 h o w P 1 Q B R S 0 s f 8 U j V x P E t u L F N 2 F M l f 8 w o H V t 4 0 u 4 S h a q 4 s H n n 1 5 K w v L 5 K P 0 f V 1
e Z E w O U P 2 D S o X G Y w 9 N t + e v e C i T 1 G R a w h e 6 y Y s B z Q 1 y F b E n 0 L S B A p z z y + / d j J S i +
5 0 w E t T M =
-----END CERTIFICATE-----
```

Figure 34 : Get Certificate from Openstack Barbican

11. You can verify all the secrets.



```
>_ Console

# su -s /bin/sh -c "barbican-manage hsm gen_hmac --library-path '/opt/
utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890 --slot-id 3 --
label 'hmac_utimaco1234' --length 32" barbican

[root@controller bin(keystone_admin)]# barbican-manage hsm gen_hmac --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphr
ase '123456' --slot-id 1 --label 'hmac_utimaco1234' --length 32
HMAC successfully generated!
```

Figure 37 : HMAC Key Generation

3. Verify that the keys are generated on the Utimaco HSM using the p11tool2 command.

```
>_ Console

#./p11tool2 slot=<slot_id> LOGINUSER=<Crypto_User_PIN> ListObjects

[root@controller bin(keystone_admin)]# ./p11tool2 slot=1 LOGINUSER=123456 ListObjects

CKO_SECRET_KEY:

+ 1.1
CKA_KEY_TYPE           = CKK_AES
CKA_UNIQUE_ID          = CK_UNAVAILABLE_INFORMATION
CKA_SENSITIVE          = CK_TRUE
CKA_EXTRACTABLE        = CK_FALSE
CKA_LABEL               = mkek_utimaco1234
CKA_ID                 =

+ 1.2
CKA_KEY_TYPE           = CKK_AES
CKA_UNIQUE_ID          = CK_UNAVAILABLE_INFORMATION
CKA_SENSITIVE          = CK_TRUE
CKA_EXTRACTABLE        = CK_FALSE
CKA_LABEL               = hmac_utimaco1234
CKA_ID                 =

[root@controller bin(keystone_admin)]# █
```

Figure 38 : Key Listing

4. Add a new label to the Barbican config file.

```
# Master KEK label (as stored in the HSM) (string value)
mkek_label = mkek_utimaco1234

# Master KEK length in bytes. (integer value)
mkek_length = 32

# Master HMAC Key label (as stored in the HSM) (string value)
hmac_label = hmac_utimaco1234
```

Figure 39 : Entries in barbican.conf file

- Restart OpenStack-Barbican-API and the https service.

```
> _ Console
```

```
# systemctl restart openstack-barbican-api.service
```

```
# systemctl restart httpd
```

- Run the rewrap\_pkek command to rewrap pkek with the new mkek.

```
> _ Console
```

```
# su -s /bin/sh -c "barbican-manage hsm rewrap_pkek" barbican
```

```
[root@controller bin(keystone_admin)]# barbican-manage hsm rewrap_pkek
2022-12-09 11:31:12.007 1222190 WARNING oslo_db.sqlalchemy.engines [-] MySQL SQL mode is 'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION', consider enabling TRADITIONAL or STRICT_ALL_TABLES
Retrieving all available projects
Retrieving KEKs for Project 4a60eab7-d73a-43f2-8837-9b6d1310d7b5
Post-change IV: b'7KfVkpSs/qYh2pjeAoByKw=', Wrapped Key: b'6VlxgtIS+FR9zjBCu5PfQ3d7AmsjqPl9cnLgs+fsptCXjSwvzAE+fW+lEsDspVvW'
```

Figure 40 : Key rewrap Output

- Restart the OpenStack-Barbican-API and httpd services.

```
> _ Console
```

```
# systemctl restart openstack-barbican-api.service
```

```
# systemctl restart httpd
```

8. Verify that you can get the secret generated earlier.

```
> _ Console
```

```
# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-
```

```
4cae-b1d9-a2a818ea30a0 --payload
```

```
[root@controller ~(keystone_admin)]# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-4cae-b1d9-a2a818ea30a0 --p
ayload
+-----+
| Field | Value |
+-----+
| Payload | password |
+-----+
[root@controller ~(keystone_admin)]#
```

Figure 41 : Get secret with payload



*This completes the integration of OpenStack Barbican and Utimaco HSM.*

## 7 Further Information

This document forms a part of the information and support that is provided by Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

All SecurityServer product documentation is also available at the Utimaco IS GmbH website:

<https://utimaco.com/>

### 7.1 References

<i>Reference</i>	<i>Title/Company</i>
[CSADMIN]	ustrust_Anchor_cHSM_Manual_Administrators.pdf