Apache Tomcat

10.0.27

Integration Guide

CryptoServer HSM

SecurityServer v4.45.5

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH
	Krefelder Straße 220
	52070 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-03-13
Status	PUBLISHED
Document No.	IG-2026-0003

Table of Contents

1 Introduction

This guide is part of the 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 integrate an Utimaco CryptoServer Hardware Security Module (HSM) with Apache Tomcat. Utimaco HSM securely stores the private key for SSL and offloads the cryptographic operations to the HSM.

1.1.1 Target Audience for This Guide

This guide is intended for Apache Tomcat and Utimaco HSM administrators.

1.1.2 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Click **Add** button
`Monospaced`	Code that is given for explanation or as an example, file paths	`./p11tool2 LoginUser=12345678 GetSlotInfo`
Italic	References and important terms	Visit the official *Utimaco Portal*.

Table 1: Document Conventions

We use special icons to highlight the most important notes and information.

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

Here you will find additional notes or supplementary information.

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

1.1.3 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
CA	Certificate Authority
CD	Compact Disc
CSADM	CryptoServer Command-line Administration Tool
CSR	Certificate Signing Request
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol

Abbreviation	Meaning
JDK	Java Development Kit
LAN	Local Area Network
MBK	Master Backup Key
P11CAT	PKCS#11 CryptoServer Administration Tool
PCIe	PCI Express Interface
PIN	Personal Identification Number
PKCS#11	Public-Key Cryptography Standard #11
RSA	Rivest-Shamir-Adleman
SSL	Secure Socket Layer
SO	Security Officer
URL	Uniform Resource Locator

Table 2: List of abbreviations

2 Overview

2.1 Apache Tomcat

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. The Apache Tomcat software is an open-source implementation of the Jakarta Servlet, Jakarta Server Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications. These specifications are part of the Jakarta EE platform.

2.2 Utimaco SecurityServer HSM

SecurityServer is a hardware security module developed by Utimaco IS GmbH. SecurityServer is a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage as well as 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 the system environment you will be using meets the following hardware and software requirements.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM with Apache Tomcat.

Operating System	Apache Tomcat	JAVA	Utimaco Security Server Version	Utimaco HSM
Rhel 8	10.0.27	Java 8 Java 11	SecurityServer V4.45.5	CryptoServer CSe-Series/Se-Series

Table 3: List of tested versions

3.2 Software Requirements

Software	Software Requirements
HSM Interfaces	CryptoServer PKCS 11 configured
JDK 8	1.8.0_342
JDK 11	11.0.16
Host VM	Host machine Operating System: Redhat 8 and above
HSM software	Utimaco Crypto Server Software 4.45.5
P11tool2	p11tool2 (3.1.1) from product package Utimaco SecurityServer 4.45.5

Software	Software Requirements
Tomcat	Tomcat version 10.0.27

Table 4: List of software requirements

3.3 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 4.45.5 or higher
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.45.5 or higher

Table 5: List of hardware requirements

Setup an account on the Utimaco support portal and request download access at the following URL: https://support.hsm.utimaco.com/.

3.4 Prerequisites

Before you begin, please ensure that you have:

- CryptoServer setup and configured. Refer to the CryptoServer documentation to setup the HSM.
- CryptoServer Default Admin replaced with a new admin user.
- MBK created and stored onto each HSM. Refer to the CryptoServer documentation to set up the MBK.
- The operating system listed in Tested Versions.
- SecurityServer as listed in Tested Versions.
- Familiarized yourself with the Apache Tomcat documents and setup process.

- The admin user for installing software on the Apache Tomcat server.
- Allowed port 443 through the firewall.

4 Installing and Configuring Utimaco SecurityServer Software

4.1 Download and Install Utimaco Software

If you have not already done so, please create and request an Utimaco Support Portal Account. This will allow you to download the software components needed for this installation.

1. Copy the downloaded software at the appropriate location on the Apache Tomcat Server.
2. Create 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 CryptoServer 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 CryptoServer software to /opt/utimaco/bin directory and make both the files executable.

›_ Console

```
# cd ~/path_to_application_folder

# cp csadm p11tool2 /opt/utimaco/bin

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

4.2 CryptoServer PKCS#11 Configuration

1. Create the directory /etc/utimaco. 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 cs_pkcs11_R3.cfg into /etc/utimaco directory.

›_ Console

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/x86-64/Crypto_APIs/PKCS11_R3/sample # cp
cs_pkcs11_R3.cfg /etc/utimaco # cd /etc/utimaco
```

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 = 1
Keepalive = true
# Set the Device to connect with
[CryptoServer]
# Device specifier
Device = <HSM_IP>
```

For more information regarding the commands and command parameters please check the CryptoServer documentation. The device may be a CryptoServer (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 Hardware (PCIe) HSM

To make your testing easier, it would be good to enable the PKCS#11 log file.

That can be enabled by editing the **Logging** Loglevel. Set the **LogPath** and Logging **Loglevel** to **1**.

For testing you may want to increase it 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 **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 p11tool2 command initialize the Slot that you want to use, and the slot user as shown below.

›_ Console

```
# ./p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key
InitToken=<SO_PIN>

# ./p11tool2 slot=<slot_no> LoginSO=<SO_PIN> InitPin=<CryptoUser_PIN>
```

```
root@tomcat bin]#
root@tomcat bin]# ./p11tool2 Slot=0 Label=tomact Login=ADMIN,ADMIN.key InitToken=123456
root@tomcat bin]#
root@tomcat bin]# ./p11tool2 Slot=0 LoginSO=123456 InitPin=123456
root@tomcat bin]#
```

Figure 1 : Slot initialization output

4.4 Create pkcs11.cfg at /etc/utimaco/

Create a file /etc/utimaco/pkcs11.cfg and add below contents to it.

›_ Console

```
name=CryptoServer
library=/opt/utimaco/lib/libcs_pkcs11_R3.so
slotListIndex=0
attributes=compatibility
attributes(*,*,*) = {
CKA_TOKEN = true
}
```

This file will be used by SunPKCS11 provider to perform cryptographic operation on Utimaco HSM.

Specify correct library path and slot index.

5 Apache Tomcat Download and Installation

To install Apache Tomcat:

1. (Optional) It is recommended to update the system with latest security patch.

›_ Console

```
# dnf -y update
```

2. Install OpenJDK For java 8:

›_ Console

```
# dnf install java-1.8.0-openjdk java-1.8.0-openjdk-devel
```

For java 11:

›_ Console

```
# dnf -y install java-11-openjdk java-11-openjdk-devel
```

3. Create a non-root user and set its password.

›_ Console

```
# useradd tomcat
# passwd tomcat
```

4. Download Tomcat 10.

›_ Console

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.0.27/bin/apache-tomcat-10.0.27.tar.gz
```

5. Create a directory.

›_ Console

```
# mkdir -p /opt/tomcat
```

6. Extract the archived file to /opt/tomcat directory.

›_ Console

```
# tar -xvf apache-tomcat-10.0.27.tar.gz -C /opt/tomcat --strip-components=1
```

7. Change ownership of the /opt/directory to tomcat user.

›_ Console

```
# chown -R tomcat:tomcat /opt/tomcat
```

8. Set executable permissions to scripts.

›_ Console

```
# chmod +x /opt/tomcat/bin/*.sh
```

9. Create Apache Tomcat Systemd file /etc/systemd/system/tomcat.service to manage tomcat service through systemctl and add below lines.

›_ Console

```
[Unit]
Description=Apache Tomcat Web Application Container
Wants=network.target
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/jre

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Change the values according to your system configuration.

```
[root@tomcat ~]# vim /etc/systemd/system/tomcat.service
[Unit]
Description=Apache Tomcat Web Application Container
Wants=network.target
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/jre

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Figure 2 : /etc/systemd/system/tomcat.service file output

10. Reload the daemon using:

›_ Console

```
# systemctl daemon-reload
```

11. Start and Enable Tomcat Service.

›_ Console

```
# systemctl start tomcat
# systemctl enable tomcat
```

12. Confirm Tomcat status that it is running using:

›_ Console

```
# systemctl status status tomcat
```

```
[root@tomcat ~]# systemctl status status tomcat
Unit status.service could not be found.
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2022-09-26 07:41:41 UTC; 3h 31min ago
 Main PID: 1059 (java)
    Tasks: 50 (limit: 49676)
   Memory: 222.8M
   CGroup: /system.slice/tomcat.service
           └─1059 /usr/lib/jvm/jre/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava

Sep 26 07:41:41 tomcat.example.com systemd[1]: Starting Apache Tomcat Web Application Container...
Sep 26 07:41:41 tomcat.example.com startup.sh[1036]: Tomcat started.
Sep 26 07:41:41 tomcat.example.com systemd[1]: Started Apache Tomcat Web Application Container.
```

Figure 3 : Tomcat service status

13. Open http://<apache_tomcat_-server-ip>:8080 in any web browser and verify if Apache tomcat page is visible.

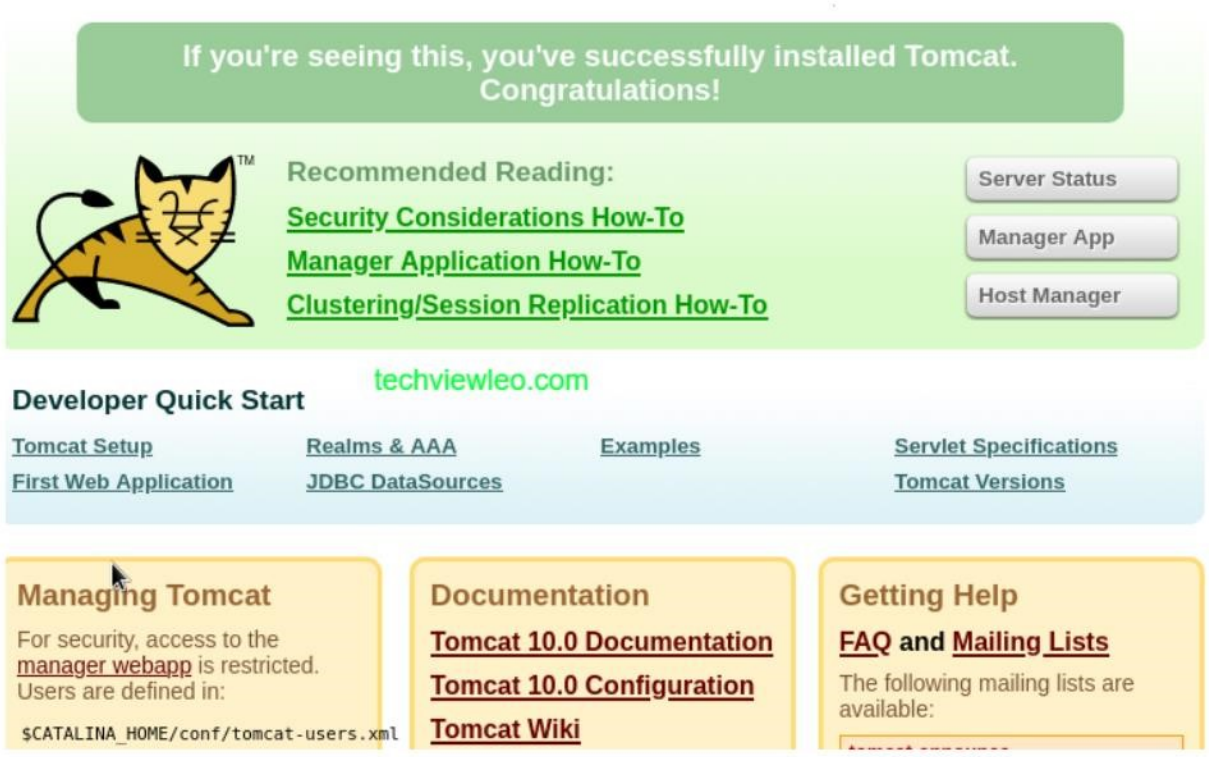


Figure 4 : Browser output over page 8080

6 Java Configuration to Use Utimaco HSM

6.1 Update java.security file to Use Utimaco HSM for JDK8

1. Go to the <JDK_Installation_directory>/jre/lib/security directory.

›_ Console

```
# cd /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b092.el8_1.x86_64/jre/lib/
security/
```

2. Edit the java.security configuration file to add SunPKCS11 provider as highlighted below.

›_ Console

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC

security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC

security.provider.10=sun.security.pkcs11.SunPKCS11 /etc/utimaco/pkcs11.cfg
```

Specify correct provider number and path for pkcs11.cfg file.

6.2 Update java.security file to Use Utimaco HSM for JDK11

1. Go to the <JDK_Installation_directory> conf/security directory.

›_ Console

```
# cd /usr/lib/jvm/java-11-openjdk-11.0.16.0.8-1.el8_2.x86_64/conf/security/
```

2. Edit the java.security configuration file to add SunPKCS11 provider.

›_ Console

```
security.provider.1=SUN security.provider.2=SunRsaSign
security.provider.3=SunEC security.provider.4=SunJSSE
security.provider.5=SunJCE security.provider.6=SunJGSS
security.provider.7=SunSASL security.provider.8=XMLDSig
security.provider.9=SunPCSC security.provider.10=JdkLDAP
security.provider.11=JdkSASL
security.provider.12=SunPKCS11 /etc/utimaco/pkcs11.cfg
```

Specify correct provider number and path for pkcs11.cfg file.

7 Generate SSL Key and Certificate for Apache Tomcat on Utimaco HSM

7.1 Generating CA Signed SSL Certificate

7.1.1 For OpenJDK8 with RSA Key

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 123456  -providername SunPKCS11-CryptoServer -alias tomcatsslkey
```

Provide information when prompted here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatsslkey is the key name that will be generated on Utimaco HSM

```
[root@tomcat2 bin]# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11 -storepass 123456  -providernam
e SunPKCS11-CryptoServer -alias tomcatsslkey
What is your first and last name?
  [Unknown]:  tomcat demo
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes
```

Figure 5 : Key generation using Keytool command

2. Verify that the keys have been generated using keytool command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 123456 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat2 bin]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 123456 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 7b81657e
Valid from: Wed Oct 19 09:50:22 UTC 2022 until: Tue Jan 17 09:50:22 UTC 2023
Certificate fingerprints:
         SHA1: D2:EA:4C:3F:59:66:CB:19:84:CA:59:BB:34:D8:73:8D:53:D1:F1:66
         SHA256: 6A:D9:19:45:9F:8B:E0:5B:84:2C:B1:FB:6B:FC:4A:70:6B:F4:97:1A:41:7E:EC:C6:AF:69:6F:26:02:18:7D:34
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 55 BA 08 59 E4 38 F4 51   55 58 F6 98 10 80 E6 88  U..Y.8.QUX......
0010: 7B 6E F9 94                                        .n..
]
]



*******************************************
*******************************************
```

Figure 6 : Listkeys output

3. List the keys using p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=123456 ListObjects
```

```
[root@tomcat2 bin]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=123456 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_LABEL                     = tomcatsslkey
  CKA_ID                        =
               0x746F6D63 61747373 6C6B6579          |tomcatsslkey    |

  CKA_SUBJECT                   =
               0x3064310B 30090603 55040613 02494E31 |0d1 0   U    IN1|
                 0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                 06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                 0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                 0F060355 040B1308 73656375 72697479 |   U    security|
                 31143012 06035504 03130B74 6F6D6361 |1 0   U    tomca|
                 74206465 6D6F                       |t demo          |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_LABEL                     =
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
               0x746F6D63 61747373 6C6B6579          |tomcatsslkey    |

[root@tomcat2 bin]#
```

Figure 7 : List keys output using p11tool2

4. Generate a CSR using Keytool command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 123456
 providername SunPKCS11-CryptoServer -alias tomcatsslkey -file tomcatssl.csr
```

Here:

- NONE is the keystore for HSM

- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatsslkey is the key name
- tomcatssl.csr is the CSR file name that will be generated

5. Get this CSR signed by CA.
6. Copy the signed certificate along with root CA certificate chain on the tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomcatsslkey -file /home
/tomcat_demo.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS11-
CryptoServer -storepass 123456
```

```
[root@tomcat2 ~]# keytool -importcert -trustcacerts -alias tomcatsslkey -file /home/tomcat_demo.p7b -storetype PKCS11 -keyst
ore NONE -providername SunPKCS11-CryptoServer -storepass 123456

Top-level certificate in reply:

Owner: CN=LabCA - Root, OU=IT, O=Security, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA - Root, OU=IT, O=Security, L=Pune, ST=MH, C=IN
Serial number: 3358f9d071b38bbc
Valid from: Tue Sep 20 11:43:00 UTC 2022 until: Mon Sep 20 11:43:00 UTC 2032
Certificate fingerprints:
         SHA1: 42:18:CA:74:C9:2C:06:55:29:C7:42:9F:D8:C1:0E:0C:A3:31:35:00
         SHA256: 1A:BC:63:DA:0E:A0:52:AE:BB:E3:02:B0:C2:7A:D8:4E:A6:D1:11:09:33:A6:88:00:3C:40:9C:D7:13:E6:B2:D4
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0C 53 65 63 75 72 69   74 79 20 4C 61 62        ..Security Lab


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
```

Figure 8 : Import user certificate into keystore

```
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 23 0E 2F D2 3E 30 A5 89   91 84 FD F6 D9 87 EA 05  #./.>0..........
0010: DF 71 F7 BF                                        .q..
]
]


... is not trusted. Install reply anyway? [no]:  yes
Certificate reply was installed in keystore
```

Signed certificate must also contain certificate chain.

8. Verify that the keytool command shows the signed certificate as well as root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 123456 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat2 ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 123456 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA - Root, OU=IT, O=Security, L=Pune, ST=MH, C=IN
Serial number: 10ce3f255a8e5c87
Valid from: Wed Oct 19 10:05:00 UTC 2022 until: Thu Oct 19 10:05:00 UTC 2023
Certificate fingerprints:
         SHA1: 95:A1:DE:76:11:12:3E:27:64:5F:AB:84:E1:46:9A:32:6C:27:71:2F
         SHA256: 64:CD:0B:74:99:EC:7A:13:AB:56:00:73:11:7A:79:20:AD:D4:28:41:5C:54:75:6D:F7:2D:17:4A:0C:0E:FE:A6
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 55 BA 08 59 E4 38 F4 51   55 58 F6 98 10 80 E6 88  U..Y.8.QUX......
0010: 7B 6E F9 94                                        .n..
]
]

Certificate[2]:
Owner: CN=LabCA - Root, OU=IT, O=Security, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA - Root, OU=IT, O=Security, L=Pune, ST=MH, C=IN
Serial number: 3358f9d071b38bbc
Valid from: Tue Sep 20 11:43:00 UTC 2022 until: Mon Sep 20 11:43:00 UTC 2032
Certificate fingerprints:
```

Figure 9 : Keytool list output

```
         SHA1: 42:18:CA:74:C9:2C:06:55:29:C7:42:9F:D8:C1:0E:0C:A3:31:35:00
         SHA256: 1A:BC:63:DA:0E:A0:52:AE:BB:E3:02:B0:C2:7A:D8:4E:A6:D1:11:09:33:A6:88:00:3C:40:9C:D7:13:E6:B2:D4
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0C 53 65 63 75 72 69   74 79 20 4C 61 62        ..Security Lab


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 23 0E 2F D2 3E 30 A5 89   91 84 FD F6 D9 87 EA 05  #./.>0..........
0010: DF 71 F7 BF                                        .q..
]
]



*******************************************
```

7.1.2 For OpenJDK8 with EC Key

1. Generate an EC keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomsslec -keyalg EC -keystore NONE -storetype PKCS11
-storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted Here:

- EC is the key algorithm
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomsslec is the key name that will be generated on Utimaco HSM

```
[root@tomcat security]# keytool -genkey -alias tomsslec -keyalg EC -keystore NONE -storetype PKCS11 -storepass 12345678 -provi
dername SunPKCS11-CryptoServer -v
What is your first and last name?
  [Unknown]:  test test
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes

Generating 256 bit EC key pair and self-signed certificate (SHA256withECDSA) with a validity of 90 days
        for: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
[Storing keystore]
```

Figure 10 : Key generation using keytool command output

2. Verify that the keys have been generated.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomsslec
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 2cea96b1
Valid from: Thu Jan 12 07:07:21 UTC 2023 until: Wed Apr 12 07:07:21 UTC 2023
Certificate fingerprints:
         MD5:  54:F3:B7:5D:34:08:BE:62:EE:56:87:83:73:90:6C:D3
         SHA1: 6A:8D:24:24:24:C1:B8:43:9A:DC:47:01:CD:BB:D6:84:05:69:BE:0F
         SHA256: 05:26:FF:29:1C:73:00:B0:EE:DE:69:AA:A1:EA:B3:54:91:4C:57:47:56:87:68:12:1A:27:85:7D:19:EB:6D:8D
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: FC 39 7A E8 13 40 42 5A   8C 95 C2 34 D0 0D E9 2B  .9z..@BZ...4...+
0010: B7 C7 22 F8                                        .."
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 11 : Listkeys output

3. List the keys using p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = 98C5B923-4164-44B6-B82F-4666B8284933
  CKA_LABEL                     = tomsslec
  CKA_ID                        = 0x746F6D73 736C6563 (tomsslec)
  CKA_SUBJECT                   =
               0x3063310B 30090603 55040613 02494E31 |0c1 0   U    IN1|
                 0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                 06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                 0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                 0F060355 040B1308 73656375 72697479 |   U    security|
                 31133011 06035504 03130A74 65737420 |1 0   U    test |
                 74657374 20                         |test            |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = D3658141-0D75-441F-8FB7-4D3FCA03907A
  CKA_LABEL                     =
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = FABFA4DE-876A-401C-930F-5A115DBBAB80
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        = 0x746F6D73 736C6563 (tomsslec)
[root@tomcat ~]#
```

Figure 12 : List Keys output using p11tool2

4. Generate a CSR using Keytool command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 12345678
 providername SunPKCS11-CryptoServer -alias tomsslec  -file tomcatec.csr
```

Here:

- NONE is the keystore for HSM

- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomsslec is the key name
- tomcatec.csr is the CSR file name that will be generated

5. Get this CSR signed by CA.
6. Copy the signed certificate along with root CA certificate chain on the tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomsslec -file
/home/tomcat/test_ec.p7b -storetype PKCS11 -keystore NONE -providername
SunPKCS11-CryptoServer -storepass 12345678
```

```
[root@tomcat ~]# keytool -importcert -trustcacerts -alias tomsslec -file /home/tomcat/test_ec.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS
11-CryptoServer -storepass 12345678

Top-level certificate in reply:

Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         MD5:  80:E7:CE:91:84:D6:E9:D9:03:53:DB:F3:11:84:F3:34
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]
```

Figure 13 : Import user certificate into keystore

```
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]


... is not trusted. Install reply anyway? [no]:  yes
Certificate reply was installed in keystore
[root@tomcat ~]#
```

Signed certificate must also contain certificate chain.

8. Verify that the keytool command shows the signed certificate as well as root CA certificate in console put below cmd.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomsslec
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=test test, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 181c921c02828bcc
Valid from: Thu Jan 12 07:28:00 UTC 2023 until: Fri Jan 12 07:28:00 UTC 2024
Certificate fingerprints:
         MD5:  62:7A:A0:7C:83:7A:E9:10:81:9A:23:D7:A5:1B:F3:81
         SHA1: 31:C6:79:BB:8F:E1:08:C1:4A:BC:16:83:C5:13:31:09:14:A8:D6:1B
         SHA256: 16:92:26:BB:D3:85:7D:5F:23:F6:26:EE:3E:42:4C:60:2A:68:B6:06:9F:BB:A9:EB:76:26:3B:FB:C6:59:BF:00
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: FC 39 7A E8 13 40 42 5A   8C 95 C2 34 D0 0D E9 2B  .9z..@BZ...4...+
0010: B7 C7 22 F8                                        ..".
]
]

Certificate[2]:
Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         MD5:  80:E7:CE:91:84:D6:E9:D9:03:53:DB:F3:11:84:F3:34
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
```

Figure 14 : Keytool list output

```
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]



*******************************************
*******************************************
```

7.1.3 For OpenJDK11 with RSA Key

1. Generate an RSA keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 12345678  -providername SunPKCS11-CryptoServer -alias tomcatrsa
```

Provide information when prompted Here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name that will be generated on Utimaco HSM

```
[root@tomcat ~]# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11 -storepass 12345678  -providername SunPKCS11-CryptoServer -a
lias tomcatrsa
What is your first and last name?
  [Unknown]:  utimaco demo
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes
```

Figure 15 : Key Generation using Keytool command

2. Verify that the keys have been generated.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatrsa
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 49204032
Valid from: Mon Jan 23 10:43:41 UTC 2023 until: Sun Apr 23 10:43:41 UTC 2023
Certificate fingerprints:
         SHA1: E8:65:ED:A5:1D:2C:36:5C:6C:4B:7C:9B:19:A6:65:49:53:69:1D:31
         SHA256: 80:D2:C9:FA:63:6C:21:E5:3C:14:2C:30:32:11:56:AD:FD:39:27:60:B8:3B:1A:64:4C:9E:20:0F:E2:D0:D2:7B
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D0 C7 CE FD 85 77 6C ED   D5 19 9E A6 D4 DB 47 84  .....wl.......G.
0010: E8 58 8C 04                                        .X..
]
]
```

Figure 16 : Listkeys output

3. List the keys using p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects

CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = 3074A9E8-C4EF-4B4F-B614-599D4E6F1FE2
  CKA_LABEL                     = tomcatrsa
  CKA_ID                        =
                0x746F6D63 61747273 61               |tomcatrsa       |

  CKA_SUBJECT                   =
                0x3065310B 30090603 55040613 02494E31 |0e1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                  0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  31153013 06035504 03130C75 74696D61 |1 0   U    utima|
                  636F2064 656D6F                     |co demo         |


CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 8A22FF77-82ED-428A-B354-F2F343471FA3
  CKA_LABEL                     =
  CKA_ID                        =


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 541F5F6C-AD49-4018-AFAE-8B5AF9F663DD
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
                0x746F6D63 61747273 61               |tomcatrsa       |
```

Figure 17 : List Keys output using p11tool2

4. Generate a CSR using Keytool command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 12345678
 providername SunPKCS11-CryptoServer -alias tomcatrsa  -file tomcatrsa.csr
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name
- tomcatrsa.csr is the CSR file name that will be generated

5. Get this CSR signed by CA.
6. Copy the signed certificate along with root CA certificate chain on the tomcat server.
7. Import the signed certificate chain reply using the command belo.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomcatrsa -file /root/tomcatrsa.p7b
-storetype PKCS11 -keystore NONE -providername SunPKCS11-CryptoServer storepass
12345678
```

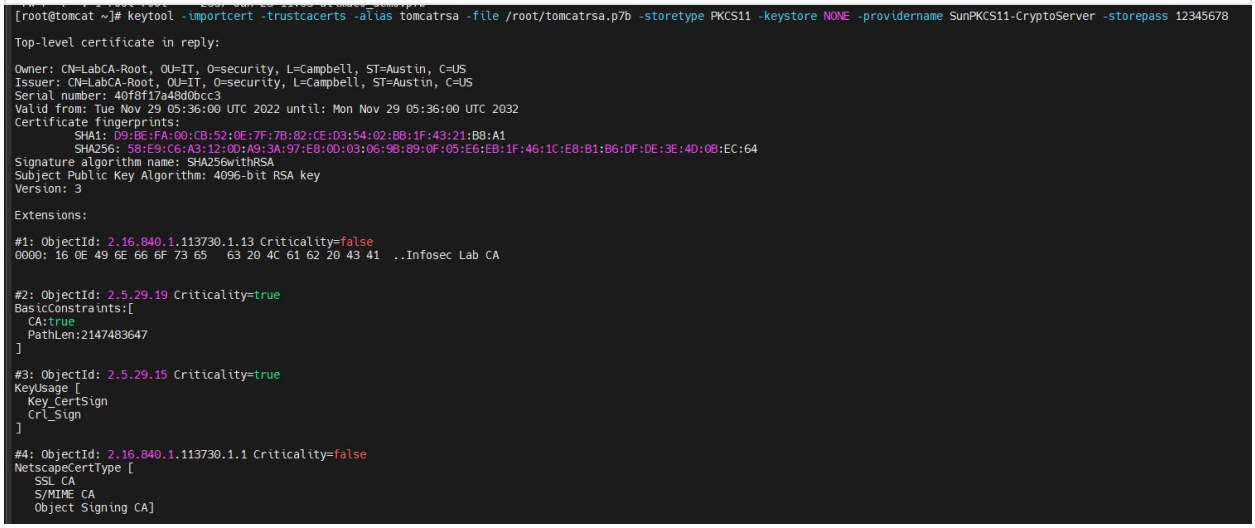
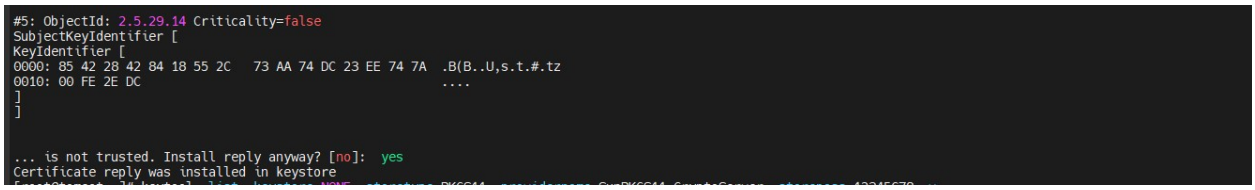


Figure 18 : Import user certificate into keystore



Signed certificate must also contain certificate chain.

8. Verify that the keytool command shows the signed certificate as well as root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-

CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatrsa
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 6a8e2b2c65363ed8
Valid from: Mon Jan 23 10:47:00 UTC 2023 until: Tue Jan 23 10:47:00 UTC 2024
Certificate fingerprints:
         SHA1: 3B:3A:7A:85:84:CA:2A:92:22:A8:39:F0:E8:C7:9D:DE:5D:97:CC:ED
         SHA256: 15:A8:AC:23:A4:F9:BD:8C:62:67:07:F9:1F:2F:0F:A5:64:36:D2:F2:18:63:37:E9:99:C8:C7:A2:84:1A:32:F9
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D0 C7 CE FD 85 77 6C ED   D5 19 9E A6 D4 DB 47 84  .....wl.......G.
0010: E8 58 8C 04                                        .X..
]
]

Certificate[2]:
Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3
```

Figure 19 : Keytool list output

```
Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

7.1.4 For OpenJDK11 with EC Key

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomssleckey -keyalg EC -keystore NONE -storetype

PKCS11 -storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted Here:

- EC is the key algorithm
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomssleckey is the key name that will be generated on Utimaco HSM

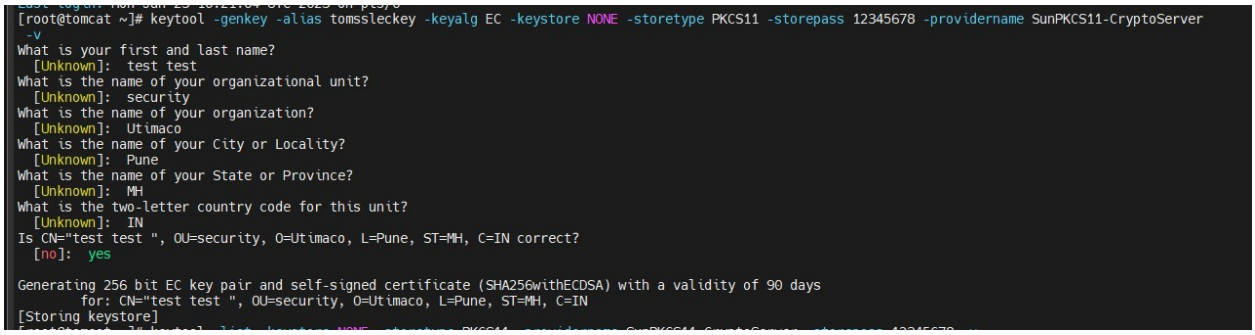


Figure 20 : Key generation using Keytool command

2. Verify that the keys have been generated.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider's name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomssleckey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 75fd19bd
Valid from: Mon Jan 23 11:33:53 UTC 2023 until: Sun Apr 23 11:33:53 UTC 2023
Certificate fingerprints:
         SHA1: 53:14:8E:28:B9:B9:C7:AE:4A:72:96:56:82:8E:1C:A8:65:C4:18:FF
         SHA256: E0:1C:21:90:62:53:51:00:8C:34:A1:73:ED:FA:47:74:35:9C:6D:5B:21:BF:0B:35:FB:0E:B1:26:8C:AE:A3:88
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: ED D7 40 2A 31 02 E8 CD   67 4F DB 12 BA 4B E4 2F  ..@*1...gO...K./
0010: B2 75 39 D6                                        .u9.
]
]



*******************************************
*******************************************
```

Figure 21 : Listkeys output

3. List the keys using p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = C8600964-25B8-469F-882C-41DDD865588B
  CKA_LABEL                     = tomssleckey
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |

  CKA_SUBJECT                   =
                0x3063310B 30090603 55040613 02494E31 |0c1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                  0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  31133011 06035504 03130A74 65737420 |1 0   U    test |
                  74657374 20                         |test           |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = 4307A902-5708-4DC5-B39E-24308E4A43E8
  CKA_LABEL                     =
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = BA833B57-8882-4CF7-84F3-B1F0C928CEDE
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |
```

Figure 22 : List Keys output using p11tool2

4. Generate a CSR using Keytool command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 12345678
 providername SunPKCS11-CryptoServer -alias tomssleckey  -file tomcateckey.csr
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomssleckey is the key name
- tomcateckey.csr is the CSR file name that will be generated

5. Get this CSR signed by CA.
6. Copy the signed certificate along with root CA certificate chain on the tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
 #keytool -importcert -trustcacerts -alias tomssleckey -file
/root/test_test.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS11-
CryptoServer -storepass 12345678
```

```
[root@tomcat ~]# keytool -importcert -trustcacerts -alias tomssleckey -file /root/test_test.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS11-CryptoServer -storepass 12345678

Top-level certificate in reply:

Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:B8:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]
```

```
... is not trusted. Install reply anyway? [no]:  yes
Certificate reply was installed in keystore
```

Signed certificate must also contain certificate chain.

8. Verify that the keytool command shows the signed certificate as well as root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- SunPKCS11-CryptoServer is the provider's name
- 12345678 is the slot PIN

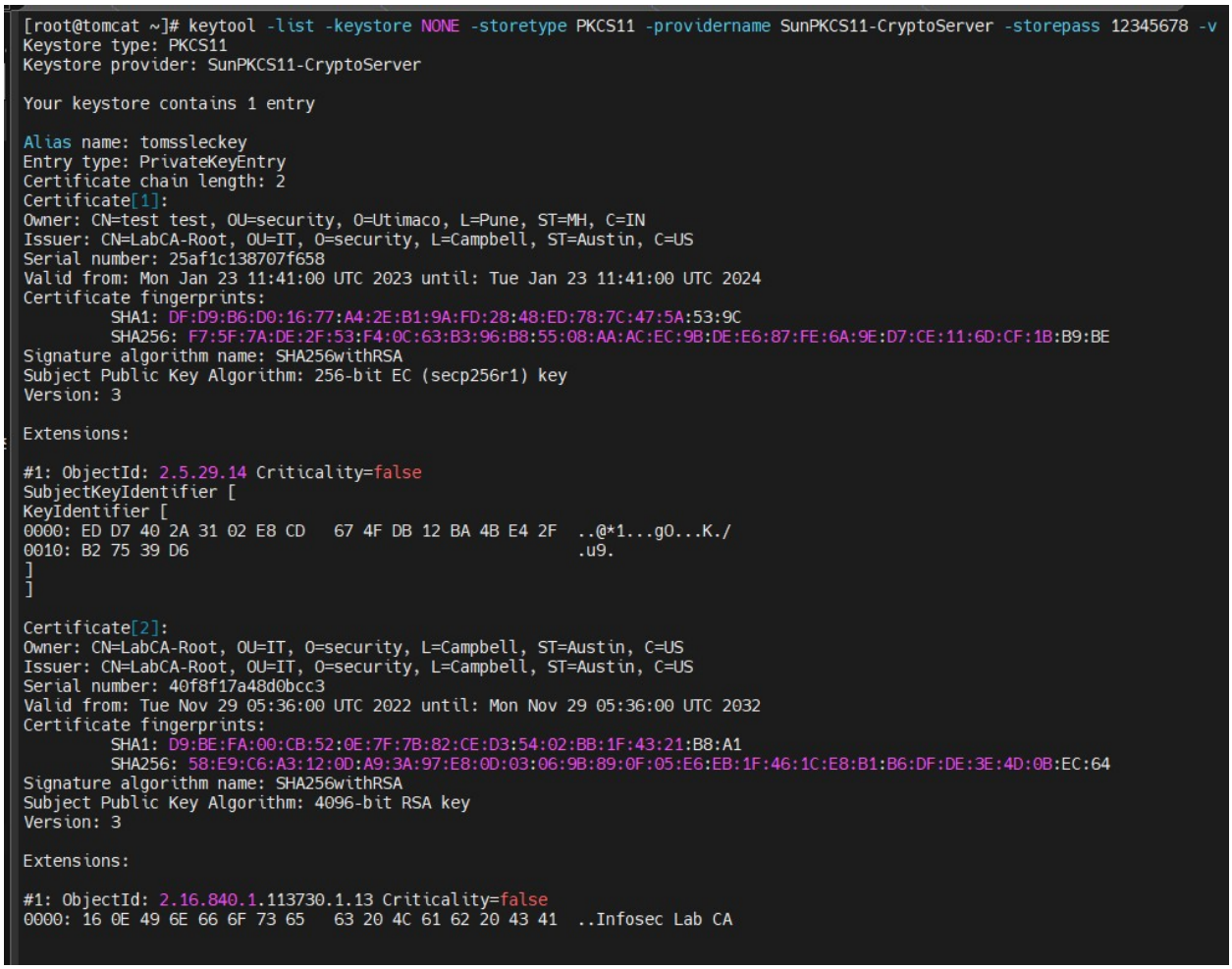


Figure 23 : Keytool list output

```
Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]



*******************************************
*******************************************
```

7.2 Using Self Sign Certificate

7.2.1 For OpenJDK8 with RSA Key Using Self Sign Certificate

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 123456  -providername SunPKCS11-CryptoServer -alias tomcatsslkey
```

Provide information when prompted here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM

- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatsslkey is the key name that will be generated on Utimaco HSM

```
[root@tomcat2 bin]# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11 -storepass 123456  -providernam
e SunPKCS11-CryptoServer -alias tomcatsslkey
What is your first and last name?
  [Unknown]:  tomcat demo
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes
```

Figure 24 : Keytool command to generate keys

It is recommended to use CA signed certificate for production environment.

2. Verify that the keys have been generated by p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=123456 ListObjects
```

```
[root@tomcat2 ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=123456 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE           = CKC_X_509
  CKA_LABEL                      = tomcatsslkey
  CKA_ID                         =
                0x746F6D63 61747373 6C6B6579          |tomcatsslkey    |

  CKA_SUBJECT                    =
                0x3065310B 30090603 55040613 02494E31 |0e1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                  0355040A 13077574 696D6163 6F311130 | U    utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  31153013 06035504 03130C75 74696D61 |1 0   U    utima|
                  636F2064 656D6F                     |co demo         |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_LABEL                      =
  CKA_ID                         =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_SENSITIVE                  = CK_FALSE
  CKA_EXTRACTABLE                = CK_TRUE
  CKA_LABEL                      =
  CKA_ID                         =
                0x746F6D63 61747373 6C6B6579          |tomcatsslkey    |
```

Figure 25 : Keytool list output using p11tool2

3. List the keys using keytool command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 123456 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 123456 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

```
[root@tomcat2 bin]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 123456 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=tomcat demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 7b81657e
Valid from: Wed Oct 19 09:50:22 UTC 2022 until: Tue Jan 17 09:50:22 UTC 2023
Certificate fingerprints:
         SHA1: D2:EA:4C:3F:59:66:CB:19:84:CA:59:BB:34:D8:73:8D:53:D1:F1:66
         SHA256: 6A:D9:19:45:9F:8B:E0:5B:84:2C:B1:FB:6B:FC:4A:70:6B:F4:97:1A:41:7E:EC:C6:AF:69:6F:26:02:18:7D:34
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 55 BA 08 59 E4 38 F4 51   55 58 F6 98 10 80 E6 88  U..Y.8.QUX......
0010: 7B 6E F9 94                                        .n..
]
]



*******************************************
*******************************************
```

Figure 26 : Keytool list output

7.2.2 For OpenJDK8 with EC Key Using Self Sign Certificate

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomsslec -keyalg EC -keystore NONE -storetype PKCS11
-storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted Here:

- EC is the key algorithm

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomsslec is the key name that will be generated on Utimaco HSM

```
[root@tomcat security]# keytool -genkey -alias tomsslec -keyalg EC -keystore NONE -storetype PKCS11 -storepass 12345678 -provi
dername SunPKCS11-CryptoServer -v
What is your first and last name?
  [Unknown]:  test test
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes

Generating 256 bit EC key pair and self-signed certificate (SHA256withECDSA) with a validity of 90 days
        for: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
```

Figure 27 : Keytool command to generate keys

It is recommended to use CA signed certificate for production environment.

2. Verify that the keys have been generated by p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

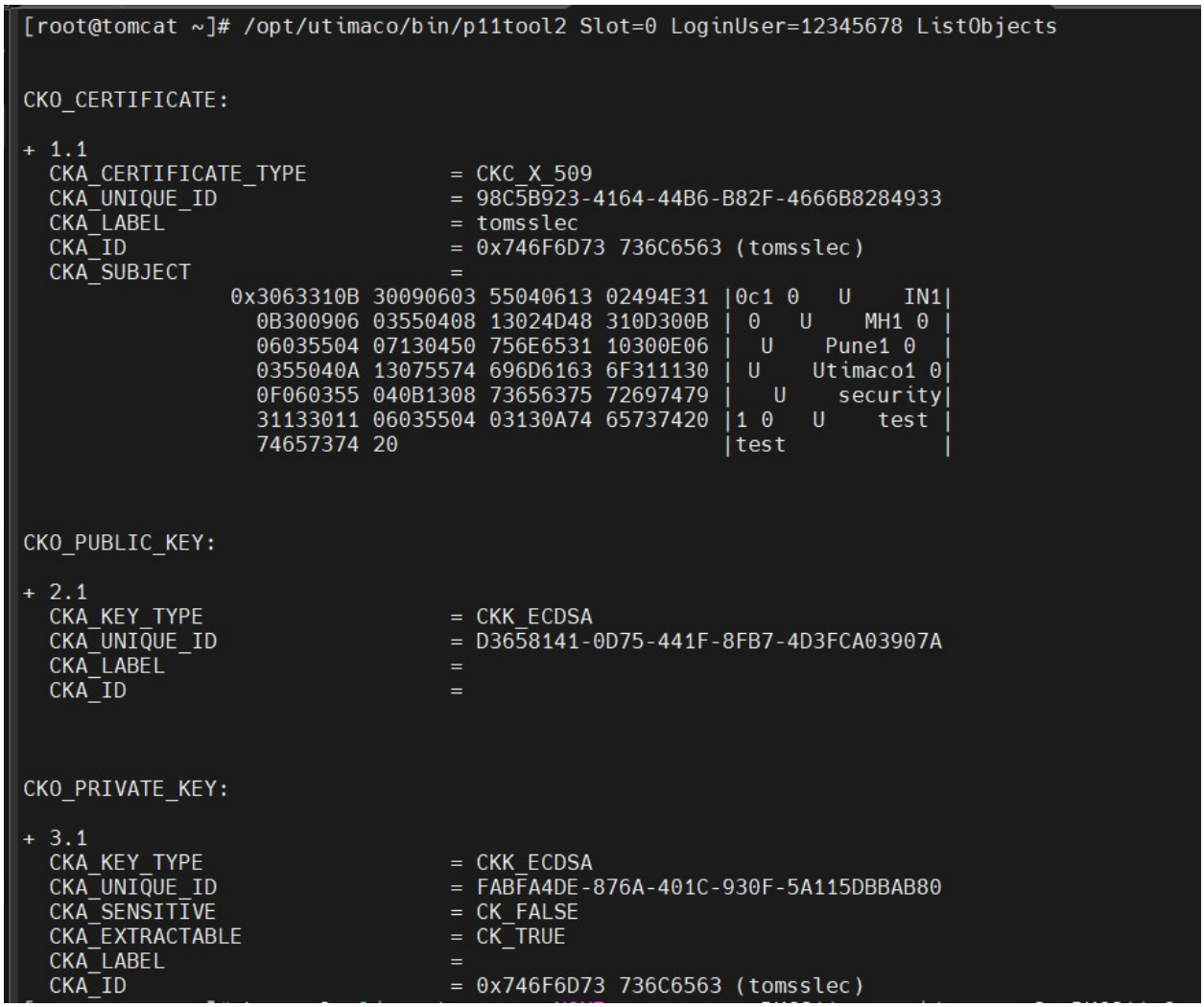


Figure 28 : List Keys output using p11tool2

3. List the keys using keytool command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype

- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

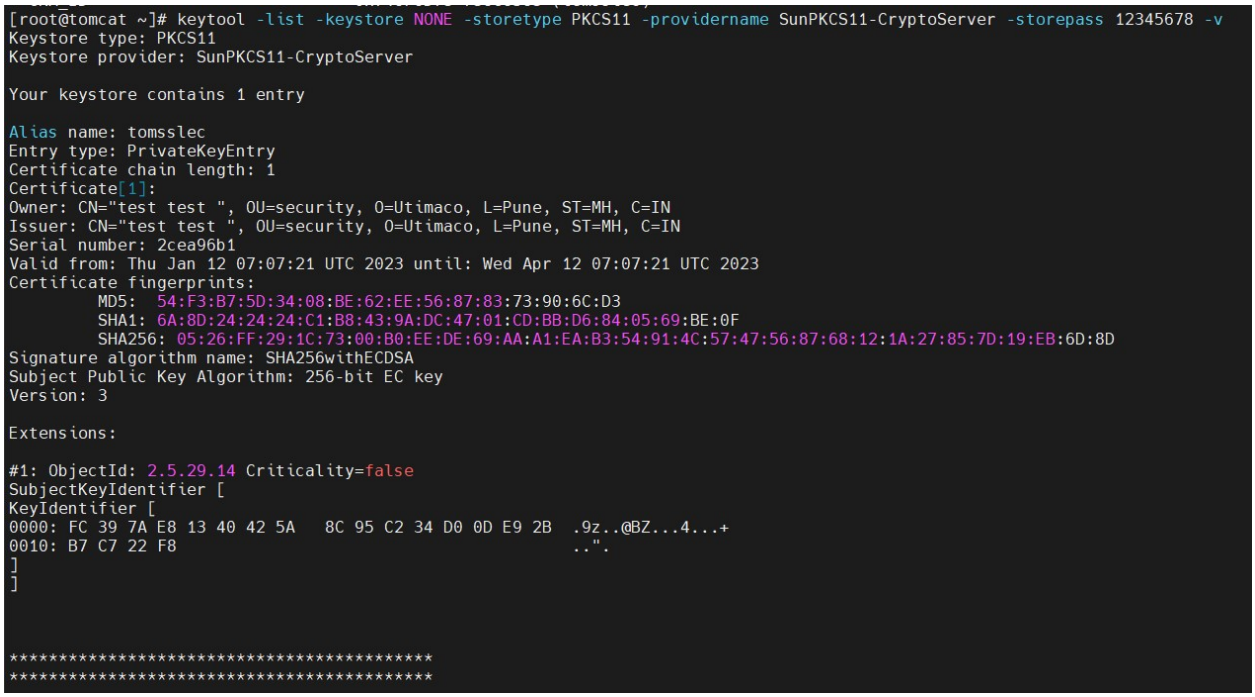


Figure 29 : Keytool List output

7.2.3 For OpenJDK11 with RSA Key Using Self Sign Certificate

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 12345678  -providername SunPKCS11-CryptoServer -alias tomcatrsa
```

Provide information when prompted Here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM
- PKCS11 is the storetype

- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name that will be generated on Utimaco HSM

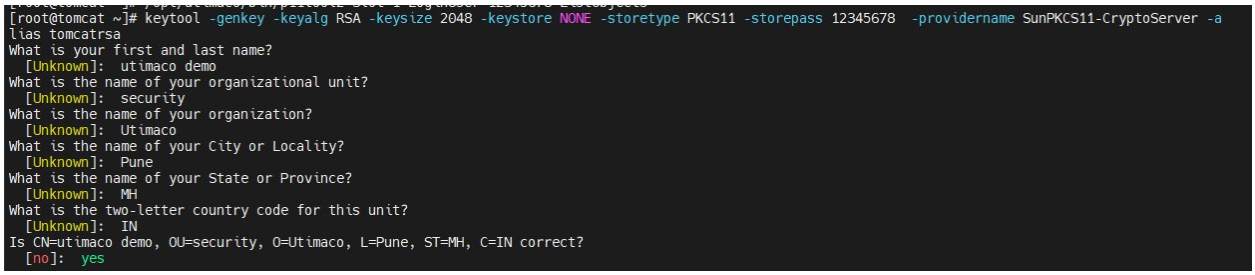


Figure 30 : Keytool command to Generate Keys

It is recommended to use CA signed certificate for production environment.

2. Verify that the keys have been generated by p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

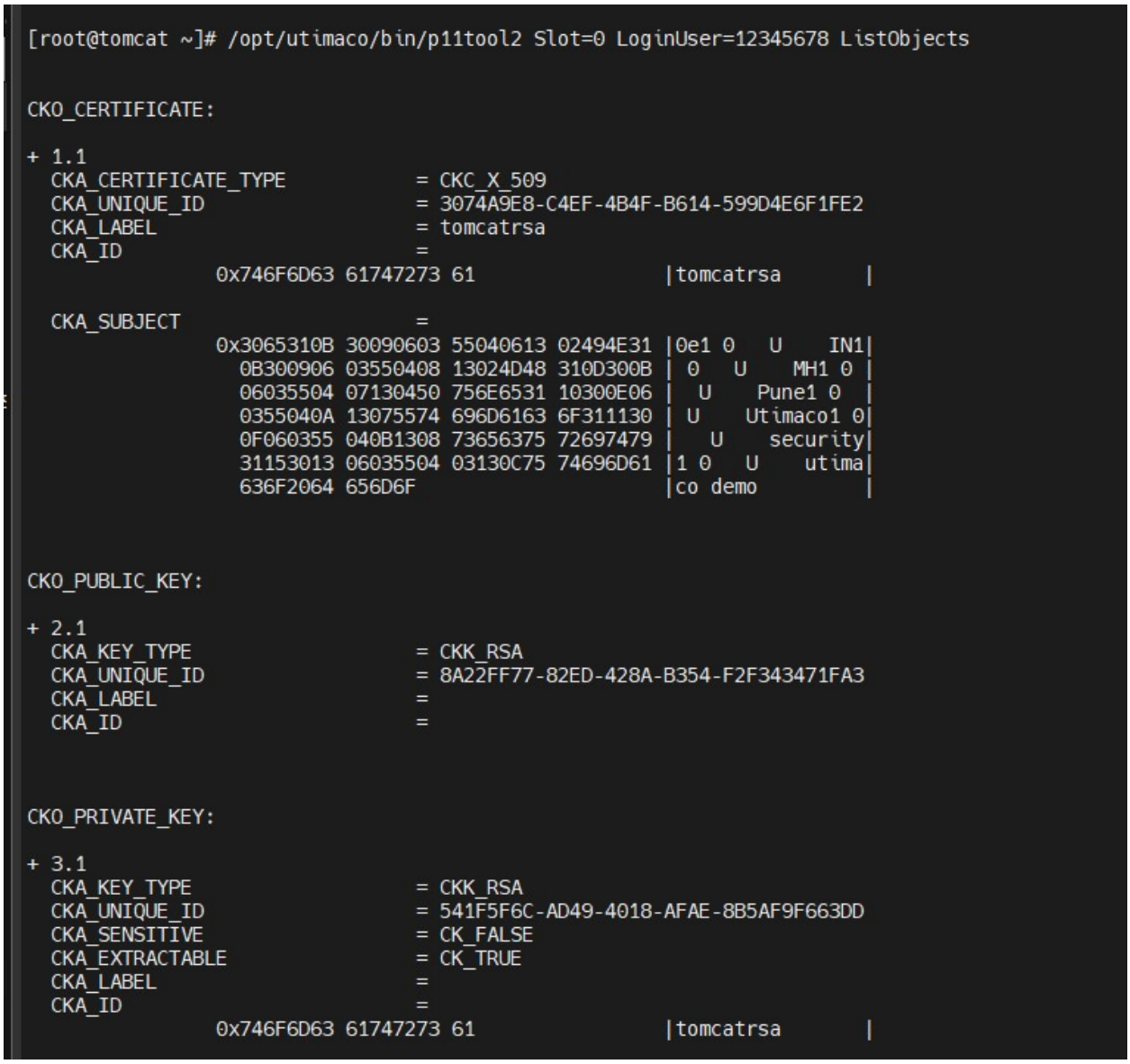


Figure 31 : List Keys output using p11tool2

3. List the keys using keytool command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- SunPKCS11-CryptoServer is the provider name
- 12345678 is the slot PIN

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatrsa
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 49204032
Valid from: Mon Jan 23 10:43:41 UTC 2023 until: Sun Apr 23 10:43:41 UTC 2023
Certificate fingerprints:
         SHA1: E8:65:ED:A5:1D:2C:36:5C:6C:4B:7C:9B:19:A6:65:49:53:69:1D:31
         SHA256: 80:D2:C9:FA:63:6C:21:E5:3C:14:2C:30:32:11:56:AD:FD:39:27:60:B8:3B:1A:64:4C:9E:20:0F:E2:D0:D2:7B
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D0 C7 CE FD 85 77 6C ED   D5 19 9E A6 D4 DB 47 84  .....wl.......G.
0010: E8 58 8C 04                                        .X..
]
]



*******************************************
*******************************************
```

Figure 32 : Keytool list output

7.2.4 For OpenJDK11 with EC Key Using Self Sign Certificate

1. Generate a keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomssleckey -keyalg EC -keystore NONE -storetype
PKCS11 -storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted Here:

- EC is the key algorithm

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomssleckey is the key name that will be generated on Utimaco HSM

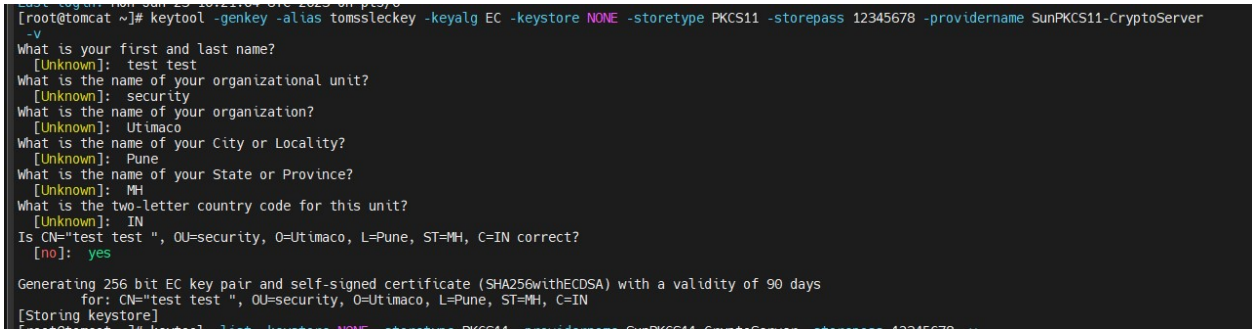


Figure 33 : Keytool command to Generate Keys

It is recommended to use CA signed certificate for production environment.

2. Verify that the keys have been generated by p11tool2.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = C8600964-25B8-469F-882C-41DDD865588B
  CKA_LABEL                     = tomssleckey
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |

  CKA_SUBJECT                   =
                0x3063310B 30090603 55040613 02494E31 |0c1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                  0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  31133011 06035504 03130A74 65737420 |1 0   U    test |
                  74657374 20                         |test           |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = 4307A902-5708-4DC5-B39E-24308E4A43E8
  CKA_LABEL                     =
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = BA833B57-8882-4CF7-84F3-B1F0C928CEDE
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |
```

Figure 34 : List Keys output using p11tool2

3. List the keys using keytool command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomssleckey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 75fd19bd
Valid from: Mon Jan 23 11:33:53 UTC 2023 until: Sun Apr 23 11:33:53 UTC 2023
Certificate fingerprints:
         SHA1: 53:14:8E:28:B9:B9:C7:AE:4A:72:96:56:82:8E:1C:A8:65:C4:18:FF
         SHA256: E0:1C:21:90:62:53:51:00:8C:34:A1:73:ED:FA:47:74:35:9C:6D:5B:21:BF:0B:35:FB:0E:B1:26:8C:AE:A3:88
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: ED D7 40 2A 31 02 E8 CD   67 4F DB 12 BA 4B E4 2F  ..@*1...gO...K./
0010: B2 75 39 D6                                        .u9.
]
]



*******************************************
*******************************************
```

Figure 35 : Keytool list output

7.3 Update server.xml file for SSL Configuration

1. Open server.xml file.

›_ Console

```
# vi /opt/tomcat/conf/server.xml
```

2. Add the following entries to connector section for SSL.

›_ Console

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
            maxThreads="150" SSLEnabled="true">
        <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
        <SSLHostConfig>
            <Certificate certificateKeystoreFile=""
                  certificateKeystoreType="PKCS11"
                  certificateKeystoreProvider="SunPKCS11-CryptoServer"
                  certificateKeyAlias=" tomcatsslkey "
                  certificateKeystorePassword="123456"
                  type="RSA" />
        </SSLHostConfig>
    </Connector>
```

Here:

- certificateKeystoreFile is blank as HSM is being used
- certificateKeystoreType is pkcs11 keystore is being used
- certificateKeystoreProvider is SunPKCS11-CryptoServer
- certificateKeyAlias is the name of the key generated using keytool command
- certificateKeystorePassword is the password of the HSM keystore
- type is the key algorithm to use (RSA/EC)

3. Reload the daemon using:

›_ Console

```
# systemctl daemon-reload
```

4. Restart Tomcat Service using:

›_ Console

```
# systemctl restart tomcat
```

5. Confirm Tomcat status that it is running using:

›_ Console

```
# systemctl status tomcat
```

6. The below output shows it is running:

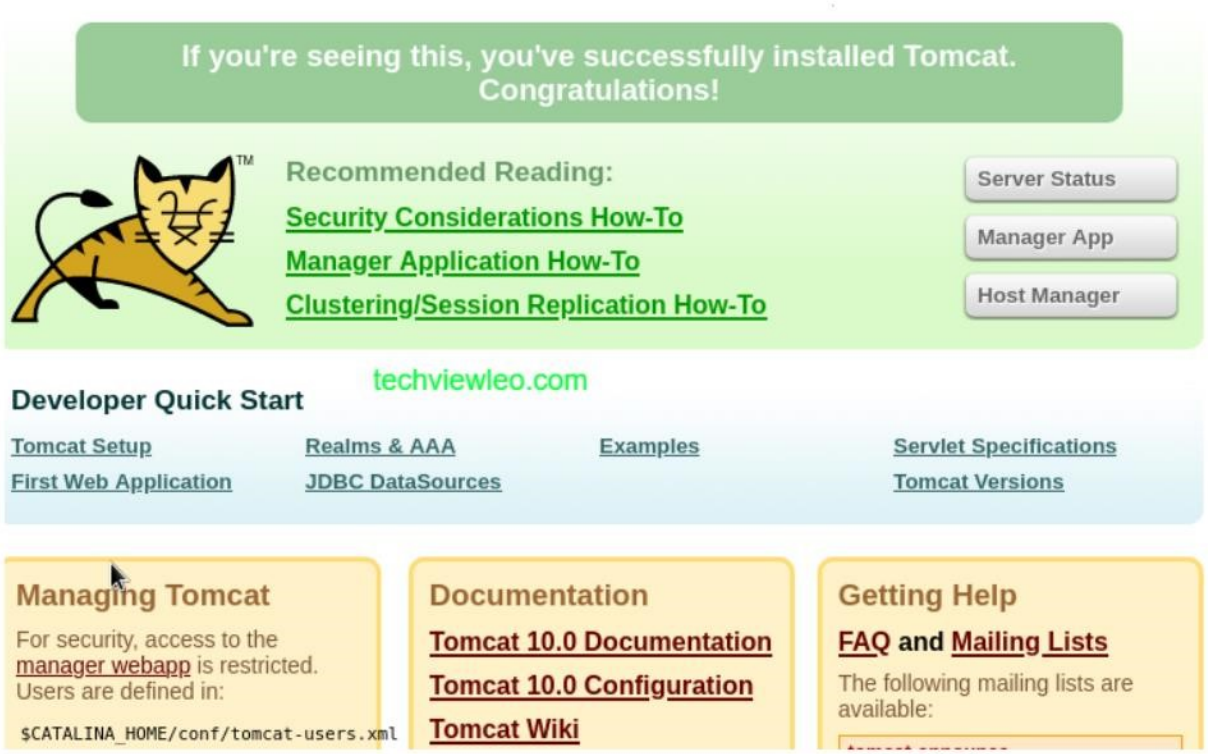


Figure 36 : Tomcat Service Status Output

7. Now access the page over https using https://172.23.0.40:8443.

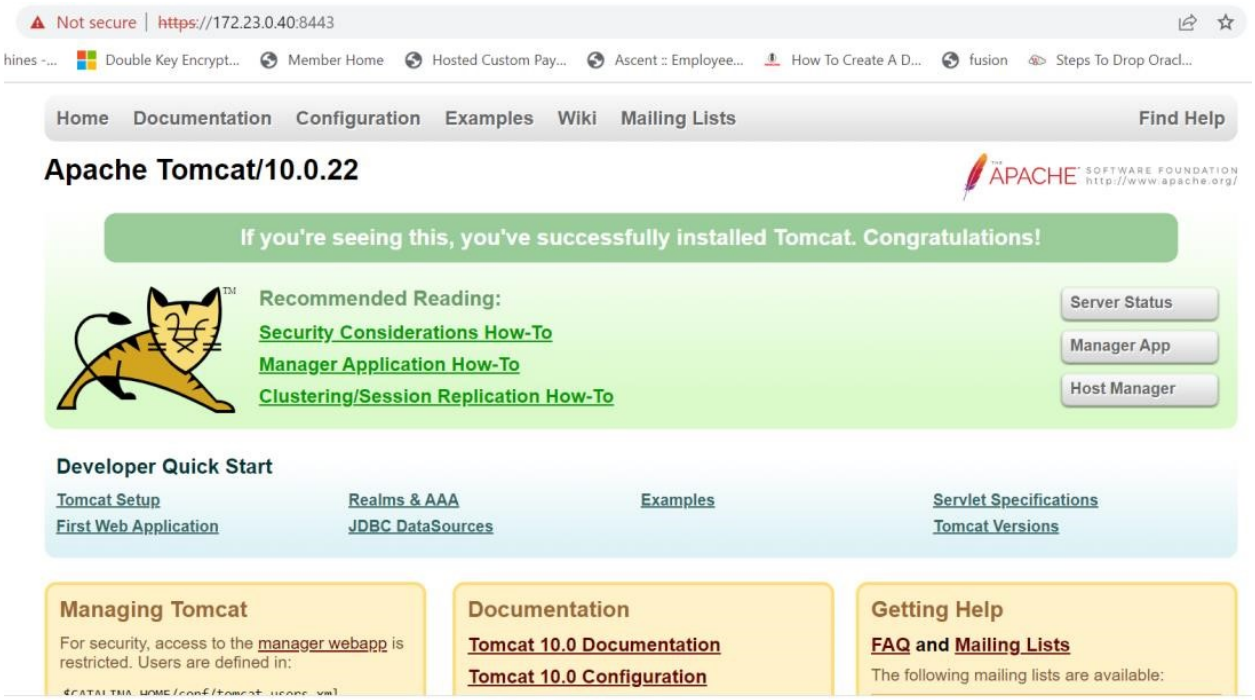


Figure 37 : Tomcat service status output

This completes the integration of Apache Tomcat with Utimaco HSM using SunPKCS11 security provider.

8 Troubleshooting

Error	Diagnosis
LoginUser= failed: 05.12.2021 23:45:45 src/p11adm_R2.c[429] p11_login: C_Login [type=1] returned Error 0x00000102 (CKR_USER_PIN_NOT_INITIALIZED)	PKCS#11 Slot is not initialized.
The CryptoServer PKCS#11 Library R3 is not initialized. Error CKR_CRYPTOKI_NOT_INITIALIZED occurred	PKCS#11 Slot is not initialized.

Table 6: List of Errors and their Diagnoses

9 Further Information

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

All CryptoServer product documentation is also available at the Utimaco IS GmbH website: https://utimaco.com/.

10 References

Reference	Title/Company	Document No.
[CSADMIN]	CryptoServer – csadm Manual/Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004

Table 7: References

11 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Str. 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.