Oracle

WebLogic Server

14.1.1

Integration Guide

CryptoServer HSM (using SunPKCS11)

4.50.0.2

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	2026-06-19
Status	PUBLISHED
Document No.	IG-2026-0047

Table of Contents

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 integrate an Utimaco SecurityServer Hardware Security Module (HSM) with Oracle WebLogic Server. 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 Oracle WebLogic Server 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	Select **Details** and click on **Properties** button
`Monospaced`	Code that is given for explanation or as an example, file paths	`certreq.exe -new request.inf IISCertRequest.csr`
Italic	References and important terms	Operating system listed in *Tested Versions*

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
EC	Elliptic Curve
GUI	Graphical User Interface
HSM	Hardware Security Module

Abbreviation	Meaning
IP	Internet Protocol
JDK	Java Development Kit
JSPs	JavaServer Pages
LAN	Local Area Network
MBK	Master Backup Key
P11CAT	PKCS#11 CryptoServer Administration Tool
PCIe	PCI Express Interface
PIN	Personal Identification number
RSA	Rivest-Shamir-Adleman
SSL	Secure Socket Layer
URL	Uniform Resource Locator
VM	Virtual Machine

Table 2: List of abbreviations

2 Overview

2.1 Oracle WebLogic Server

Oracle WebLogic Server is a unified and extensible platform for developing, deploying and running enterprise applications, such as Java, for on-premises and in the cloud. WebLogic Server offers a robust, mature, and scalable implementation of Java Enterprise Edition (EE) and Jakarta EE.

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 Oracle WebLogic Server.

Operating System	Oracle WebLogic Server Version	JAVA	Utimaco Security Server Version	Utimaco HSM
Rhel 8	14.1.1	Java 11 Java 8	SecurityServer V4.50.0.2	CryptoServer CSe-Series/Se-Series

Table 3: List of tested versions

3.2 Software Requirements

Software	Software Requirements
HSM Interfaces	SecurityServer PKCS#11 configured
JDK 8	1.8.0_361
JDK 11	11.0.16
Host OS	Redhat 8
HSM software	Utimaco SecurityServer Software 4.50.0.2

Software	Software Requirements
p11tool2	p11tool2 from product package Utimaco SecurityServer 4.50.0.2
Oracle WebLogic Server	Oracle WebLogic Server version 14.1.1

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 4.50.0.2 or higher
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.50.0.2 or higher

Table 5: List of hardware requirements

Set up 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:

- The SecurityServer is set up and configured. Refer to the SecurityServer documentation to set up the HSM.
- The SecurityServer Default Admin is replaced with a new admin user.
- The MBK has been created and stored onto each HSM. Refer to the SecurityServer documentation to set up the MBK.
- The operating system used is listed in Tested Versions.

- The SecurityServer used is listed in Tested Versions.
- You familiarize yourself with the Oracle WebLogic Server documents and setup process.
- You have set up an admin user for installing software on Oracle WebLogic Server.
- Port 7002 is allowed 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 Oracle WebLogic 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 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`, `ADMIN.key` and `p11tool2` files from Utimaco SecurityServer software to `/opt/utimaco/bin` directory and make both the files executable.

›_ Console

```
# cd ~/path_to_application_folder

# cp csadm ADMIN.key p11tool2 /opt/utimaco/bin

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

4.2 SecurityServer 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 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 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.

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

›_ Console

```
# useradd oracle # passwd oracle
```

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 Security Officer (SO) and using the 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=<ask>

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

```
[root@weblserver jdk-11.0.6]# /opt/utimaco/bin/p11tool2 slot=0 Label=weblogic Login=ADMIN,/opt/utimaco/bin/ADMIN.key InitToken
=ask
Enter SO PIN:
Repeat SO PIN:
[root@weblserver jdk-11.0.6]# /opt/utimaco/bin/p11tool2 slot=0 LoginSO=ask InitPin=ask
Enter SO PIN:
Enter normal user PIN:
Repeat normal user PIN:
[root@weblserver jdk-11.0.6]#
```

Figure 1 : Slot Initialization output

5 Oracle WebLogic Server Download and Installation

To install Oracle WebLogic server:

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

›_ Console

```
# dnf -y update
```

2. Install/Open Oracle JDK.

Refer to the Oracle WebLogic support matrix for the version of java compatible with WebLogic 12c and 14c.

For java 8:

›_ Console

```
# tar -xvf jdk-8u361-linux-x64.tar.gz

# export PATH =/usr/lib/java/jdk1.8.0_361/
```

For java 11:

›_ Console

```
# tar -xvf jdk-11.0.6_linux-x64_bin.tar.gz

# export PATH=/home/oracle/jdk-11.0.6/bin:$PATH
```

3. Download the Generic Installer for Oracle WebLogic server from the Oracle official site https://www.oracle.com/middleware/technologies/weblogic-server-installers-downloads.html.

4. Pre-installation tasks:

›_ Console

```
# groupadd -g 1001 oinstall

# usermod -u 1001 -g oinstall oracle
```

5. Create a directory to install the WebLogic software and set the necessary permissions.

›_ Console

```
# mkdir -p /u01/app/oracle/product/<WebLogic_Version> # chown -R
oracle:oinstall /u01/app

# chmod -R 775 /u01
```

6. Log in to Oracle user.

7. Edit .bash_profile of Oracle user in vim text editor and add following environment variables in this file.

›_ Console

```
export JAVA_HOME=/usr/lib/java/jdk1.8.0_361/ export JRE_HOME=/usr/lib/java/
jdk1.8.0_361/jre export PATH=$PATH:/usr/lib/java/jdk1.8.0_361/ export
ORACLE_BASE=/u01/app/oracle

export MW_HOME=$ORACLE_BASE/product/14.1.1 export WLS_HOME=$MW_HOME/wlserver

export WL_HOME=$WLS_HOME

export DOMAIN_BASE=$ORACLE_BASE/config/domain export DOMAIN_HOME=$DOMAIN_BASE/
admin/admindomain
```

Change the values according to your system configuration.

8. Source/Execute the **.bash_profile** script to set environment variables for the current Linux shell.

›_ Console

```
$ . ~/.bash_profile
```

9. Execute the **unzip** command to extract the WebLogic Generic Installer.

›_ Console

```
[oracle@orcl-weblogic ~]$ unzip fmw_<version>_wls_lite_Disk1_1of1.zip
```

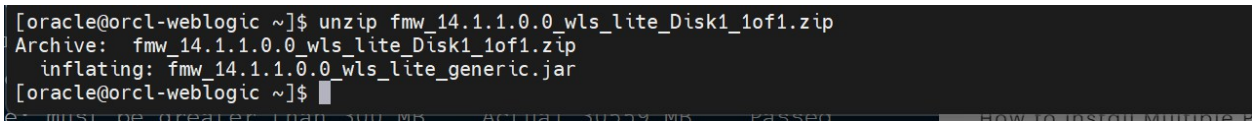

Figure 2 : Unzipping the Installer

10. Execute the Generic Installer JAR file by using the following **java** command with Oracle user.

›_ Console

```
$ java -jar fmw_<version>_wls_lite_generic.jar
```

11. A window will appear asking for installation related information. Follow the prompt and finish the installation.

12. Open http://<oracle_weblogic_-server-ip>:7001 in any web browser and verify if the Oracle WebLogic page is visible. Log in with the Administrative user which was created in the installation and configuration steps.

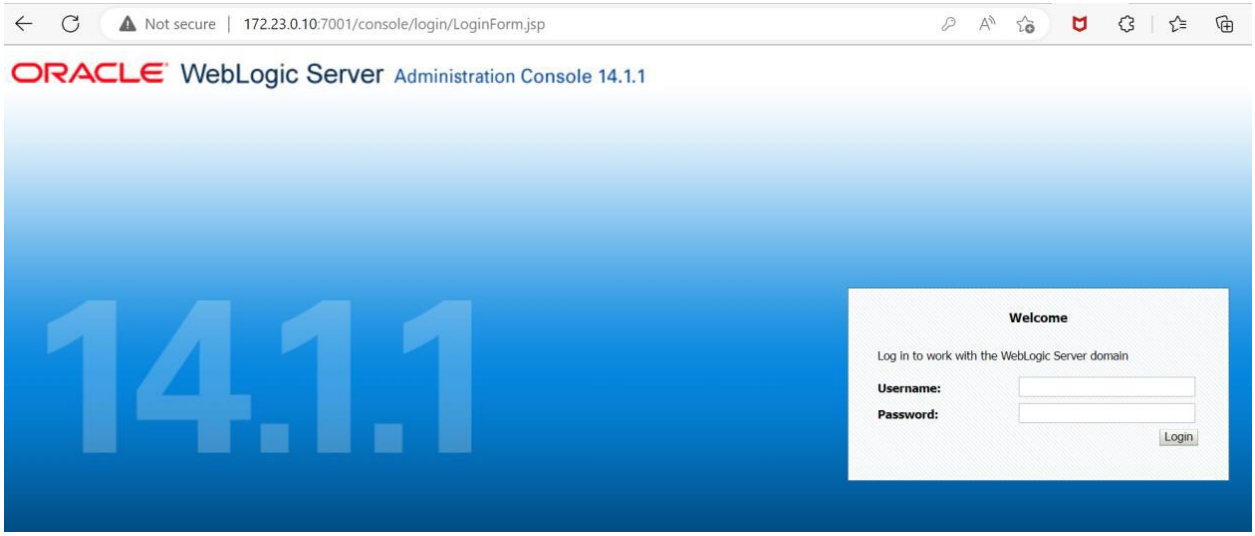

Figure 3 : Browser output over page 7001

6 Java Configuration to use Utimaco HSM

6.1 Create a pkcs11 File

1. Create a file `/u01/app/oracle/config/domain/admindomain/pkcs11` and add the 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 the SunPKCS11 provider to perform cryptographic operations on the Utimaco HSM.

2. Obtain the below jurisdiction (unlimited strength) policy files from Oracle for your country and Java version:

    a. US_export_policy.jar

    b. local_policy.jar

The unlimited policy files are required only for JDK 8 updates earlier than 8u161. On those and later versions, the stronger cryptographic algorithms are available by default.

5. Copy these jurisdiction policy files into the directory `<java-home>/lib/security`.

›_ Console

```
# cp US_export_policy.jar <java_home>/lib/security # cp local_policy.jar
<java_home>/lib/security
```

6.2 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/java/jdk1.8.0_361/jre/lib/security/
```

2. Edit the `java.security` configuration file to add the 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

/u01/app/oracle/config/domain/admindomain/pkcs11
```

Specify the correct provider number for SunPKCS11 Provider and path for the pkcs11 file.

6.3 Update java.security File to use Utimaco HSM for JDK11

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

›_ Console

```
# cd /home/oracle/jdk-11.0.6/conf/security/
```

2. Edit the `java.security` configuration file to add the 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

/u01/app/oracle/config/domain/admindomain/pkcs11
```

Specify the correct provider number for the SunPKCS11 Provider and path for pkcs11 file.

7 SSL Setup for Oracle WebLogic Server on Utimaco HSM

7.1 For OpenJDK8 with RSA Key

1. Log in as root user, and generate a keypair on Utimaco HSM.

›_ Console

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

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.
- weblkey is the key name that will be generated on Utimaco HSM.

```
[root@weblogic-sunpkcs11 ~]# keytool -genkey -alias weblkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11 -storep
ass 12345678 -providername SunPKCS11-CryptoServer -v
What is your first and last name?
  [Unknown]:  rsa 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=rsa demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN correct?
  [no]:  yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days
        for: CN=rsa demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
[Storing keystore]
[root@weblogic-sunpkcs11 ~]#
```

Figure 4 : Key generation using keytool command

Self-sign certificate doesn't work with WebLogic server.

2. Verify that the keys have been generated using the `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's name.

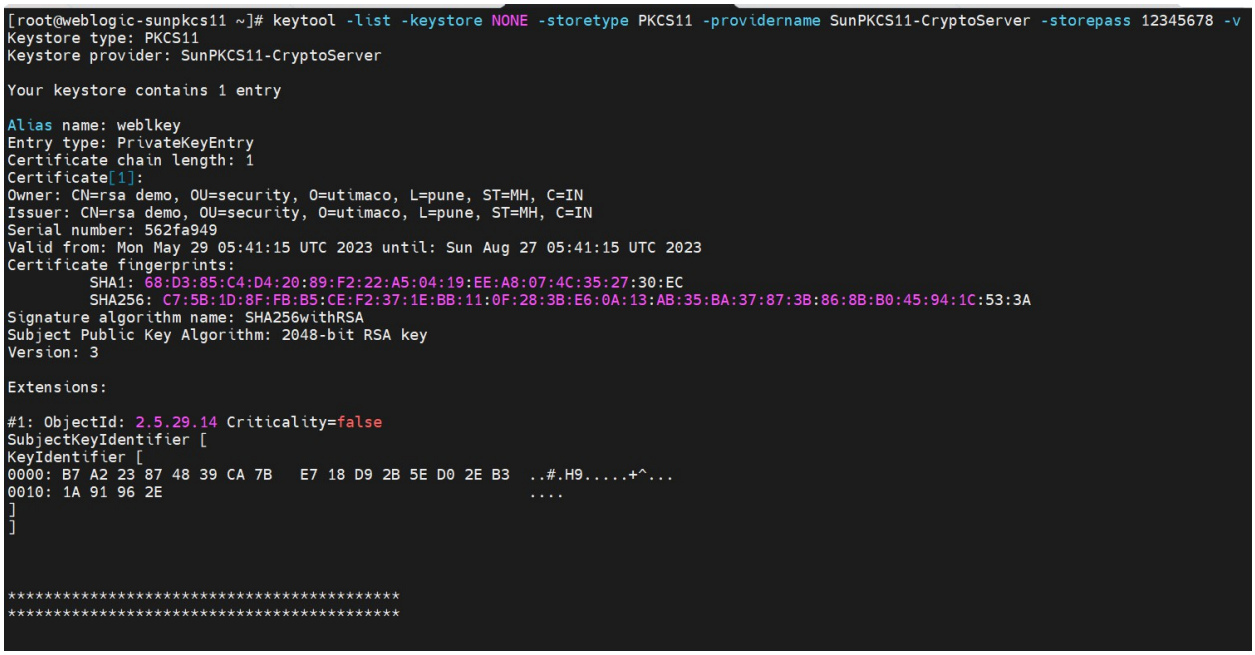

Figure 5 : List keys output

3. List the keys using `p11tool2`.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=<slot_PIN> ListObjects
```

```
[root@weblogic-sunpkcs11 ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE           = CKC_X_509
  CKA_UNIQUE_ID                  = AAB9C98C-790E-4878-A184-67C733335B30
  CKA_LABEL                      = weblkey
  CKA_ID                         = 0x7765626C 6B6579 (weblkey)
  CKA_SUBJECT                    =
               0x3061310B 30090603 55040613 02494E31 |0a1 0   U    IN1|
                 0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                 06035504 07130470 756E6531 10300E06 |  U    pune1 0  |
                 0355040A 13077574 696D6163 6F311130 | U    utimaco1 0|
                 0F060355 040B1308 73656375 72697479 |   U    security|
                 3111300F 06035504 03130872 73612064 |1 0   U    rsa d|
                 656D6F                              |emo             |


CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_UNIQUE_ID                  = 404ED210-9BD6-4D59-90EB-CFE7FDAB8665
  CKA_LABEL                      =
  CKA_ID                         =


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_UNIQUE_ID                  = EDA95275-F6F1-467D-85A1-DDED10F5FCAF
  CKA_SENSITIVE                  = CK_FALSE
  CKA_EXTRACTABLE                = CK_TRUE
  CKA_LABEL                      =
  CKA_ID                         = 0x7765626C 6B6579 (weblkey)
```

Figure 6 : List keys output using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias weblkey -file rsa.csr -storetype PKCS11 -keystore NONE
-v
```

Provide the keystore password when prompted.

Here:

- NONE is the keystore for HSM.
- PKCS11 is the storetype.
- weblkey is the key name.
- rsa.csr is the CSR file name that will be generated.

5. Get this CSR signed by CA.
6. Copy the signed certificate on the WebLogic server.
7. Copy the cacerts from `<java_installation_path>/jdk1.8.0_361/jre/lib/security/` to `/u01/app/oracle/config/domain/admindomain/`.
8. Import the signed certificate reply using the command below.

›_ Console

```
# keytool -importcert -alias weblkey -file /home/rsa_demo.p7b -storetype PKCS11
-keystore NONE -providername SunPKCS11-CryptoServer -storepass 12345678
```

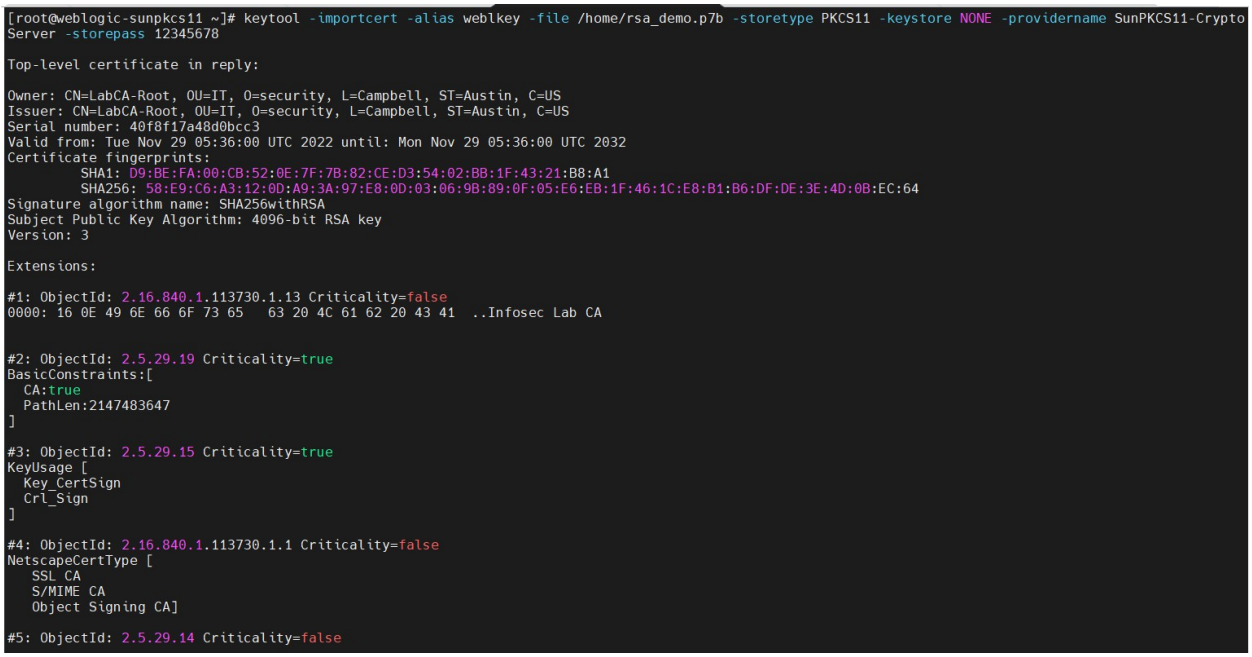

Figure 7 : Import user certificate into keystore

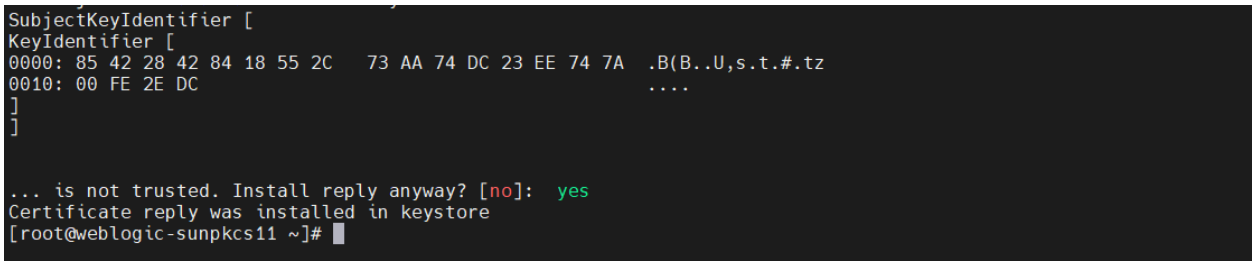

Figure 8 : Import user certificate into keystore

9. Verify that the `keytool` command shows the signed 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@weblogic-sunpkcs11 ~]# 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: weblkey
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=rsa 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: 9ea4b6952218a2c
Valid from: Mon May 29 05:47:00 UTC 2023 until: Wed May 29 05:47:00 UTC 2024
Certificate fingerprints:
         SHA1: C2:4A:01:FE:0A:7E:AF:D5:13:47:2C:43:94:07:90:5E:52:C6:55:10
         SHA256: 7C:11:02:E4:D0:0E:70:A9:7A:46:4B:6F:25:58:4B:A0:F8:22:24:92:D7:8B:06:FA:56:18:90:B8:09:F4:D6:75
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  codeSigning
]

#2: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B7 A2 23 87 48 39 CA 7B   E7 18 D9 2B 5E D0 2E B3  ..#.H9.....+^...
0010: 1A 91 96 2E                                        ....
]
]
```

Figure 9 : Keytool list output

```
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

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                                        ....
]
]



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

Figure 10 : Keytool list output

7.2 For OpenJDK8 with EC Key

1. Generate an EC keypair on Utimaco HSM.

›_ Console

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

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.
- webleckey is the key name that will be generated on Utimaco HSM.

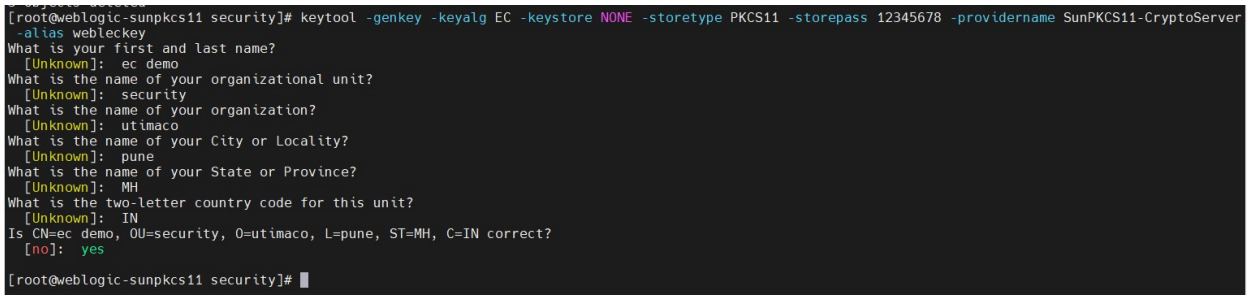

Figure 11 : Key generation using keytool command output

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@weblogic-sunpkcs11 security]# 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: webleckey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=ec demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Issuer: CN=ec demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Serial number: 11a37b15
Valid from: Thu May 25 06:54:17 UTC 2023 until: Wed Aug 23 06:54:17 UTC 2023
Certificate fingerprints:
         SHA1: 81:9C:F7:50:7E:7F:0E:9E:9B:1B:C1:05:7E:7B:EB:46:A9:F2:CA:29
         SHA256: 08:2B:92:65:F7:44:E1:FF:FE:A3:B8:15:84:B7:53:AE:B3:5B:78:61:6B:B6:D6:F7:79:2C:EB:35:0D:A2:B8:5A
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: C9 D3 0C 98 A1 6E 98 44   D1 69 63 18 41 B5 99 AF  .....n.D.ic.A...
0010: 40 25 47 54                                        @%GT
]
]



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


[root@weblogic-sunpkcs11 security]#
```

Figure 12 : List keys output

3. List the keys using `p11tool2`.

›_ Console

```
# p11tool2 Slot=0 LoginUser=<slot_PIN> ListObjects
```

```
[root@weblogic-sunpkcs11 security]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = 2971BD1A-52EE-4C58-AF6E-CA33D79D9D5C
  CKA_LABEL                     = webleckey
  CKA_ID                        =
                0x7765626C 65636B65 79               |webleckey       |

  CKA_SUBJECT                   =
                0x3060310B 30090603 55040613 02494E31 |0`1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130470 756E6531 10300E06 |  U    pune1 0  |
                  0355040A 13077574 696D6163 6F311130 | U    utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  3110300E 06035504 03130765 63206465 |1 0   U    ec de|
                  6D6F                               |mo              |


CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = D74CEDE0-4A3E-4B53-9554-C25DC4AD0D1D
  CKA_LABEL                     =
  CKA_ID                        =


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = B814FB58-D887-4BBC-9F08-4BF9D3C30AB4
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
                0x7765626C 65636B65 79               |webleckey       |

[root@weblogic-sunpkcs11 security]#
```

Figure 13 : List keys output using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias webleckey -file ec.csr -storetype PKCS11 -keystore
NONE -v
```

Provide the keystore password when prompted.

Here:

- NONE is the keystore for HSM.
- PKCS11 is the storetype.
- webleckey is the key name.

- ec.csr is the CSR file name that will be generated.

5. Get this CSR signed by the CA.
6. Copy the signed certificate on the WebLogic server.
7. Copy the cacerts from `<java_installation_path>/jdk1.8.0_361/jre/lib/security/` to `/u01/app/oracle/config/domain/admindomain/`.
8. Import the signed certificate using the command below.

›_ Console

```
# keytool -importcert -alias webleckey -file /home/ec_demo.p7b -storetype PKCS11
-keystore NONE -providername SunPKCS11-CryptoServer -storepass 12345678
```

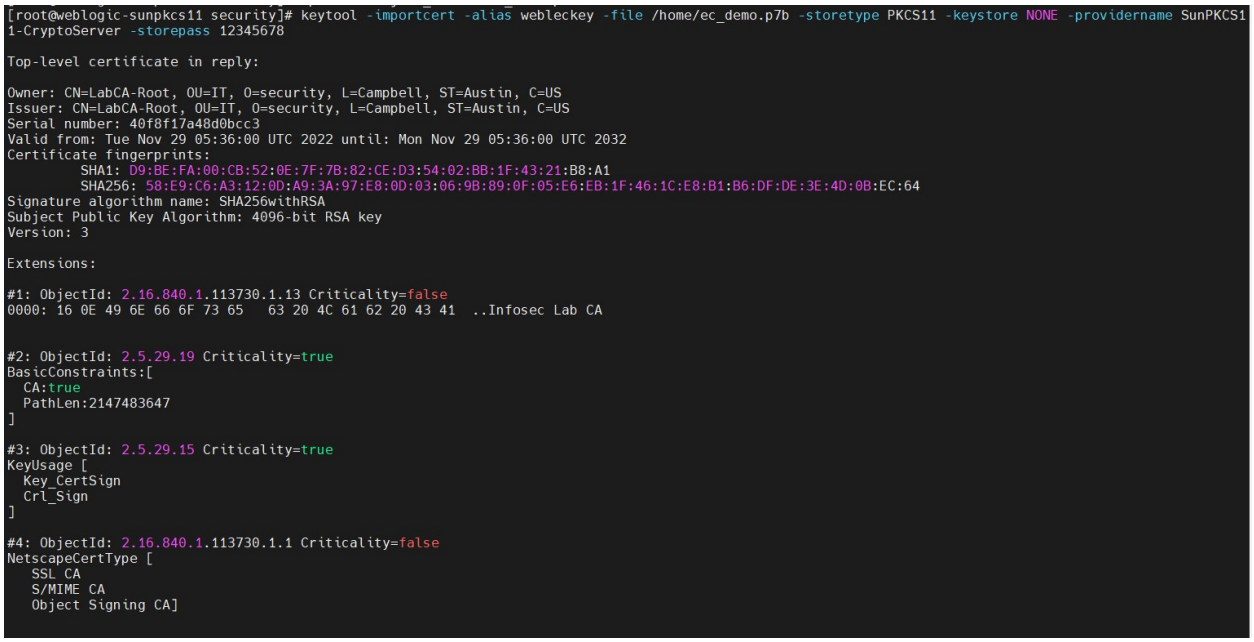

Figure 14 : Import user certificate into keystore

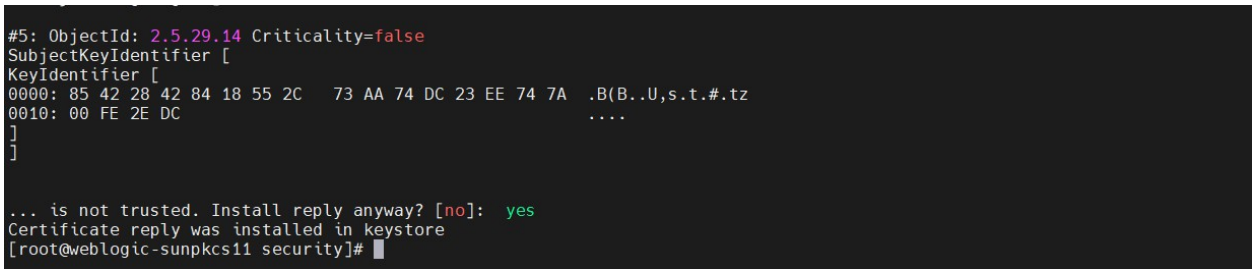

Figure 15 : Import user certificate into keystore

9. Verify that the `keytool` command shows the signed 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@weblogic-sunpkcs11 security]# 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: webleckey
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=ec 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: 303db41deaec2d5b
Valid from: Thu May 25 07:05:00 UTC 2023 until: Sat May 25 07:05:00 UTC 2024
Certificate fingerprints:
         SHA1: 7D:00:A5:4E:EF:CF:CD:11:8E:A2:74:7B:E2:86:50:14:71:7C:93:D7
         SHA256: 64:E3:6C:A9:66:6C:BD:47:8F:55:DB:19:B6:85:C6:BB:58:39:7E:61:F1:AD:F1:16:A9:9F:1E:70:16:65:32:A0
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  codeSigning
]

#2: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: C9 D3 0C 98 A1 6E 98 44   D1 69 63 18 41 B5 99 AF  .....n.D.ic.A...
0010: 40 25 47 54                                        @%GT
]
]
```

Figure 16 : Keytool list output

```
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

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                                        ....
]
]



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

Figure 17 : Keytool list output

7.3 For OpenJDK11 with RSA Key

1. Generate an RSA Keypair on Utimaco HSM.

›_ Console

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

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.
- webrsa is the key name that will be generated on Utimaco HSM.

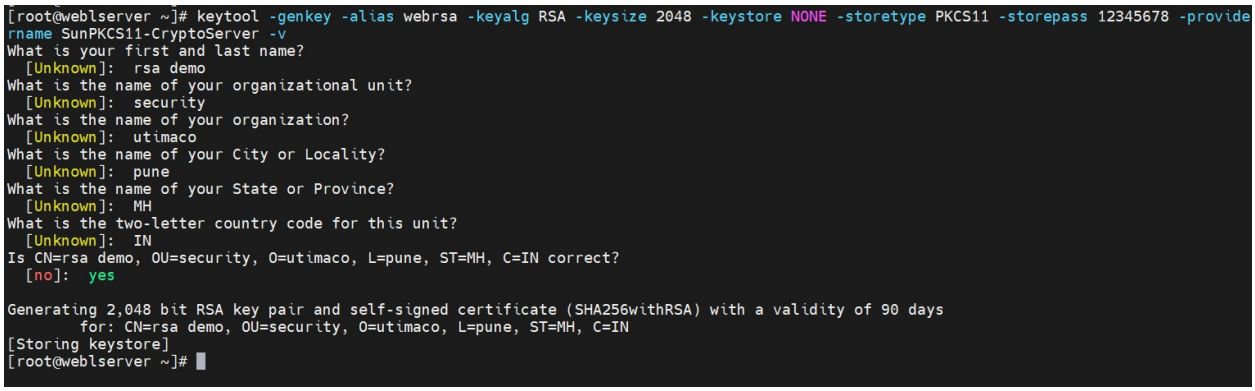

Figure 18 : Key generation using keytool command

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@weblserver ~]# 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: webrsa
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=rsa demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Issuer: CN=rsa demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Serial number: 3ef8be2f
Valid from: Mon May 29 06:03:24 UTC 2023 until: Sun Aug 27 06:03:24 UTC 2023
Certificate fingerprints:
         SHA1: B8:45:6A:B0:D1:59:C6:5A:A0:C1:20:BE:FD:43:39:9F:05:7A:EA:05
         SHA256: 10:91:27:C7:BE:96:AE:A1:6F:AB:98:52:EF:8A:B2:9C:8B:99:DA:02:27:54:58:E7:D2:70:DE:1C:F4:AE:4B:CD
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: 79 00 6B C2 C9 DC DD 16   24 64 3E A8 12 97 46 75  y.k.....$d>...Fu
0010: 66 CE 5B 8D                                        f.[.
]
]



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

Figure 19 : List keys output

3. List the keys using `p11tool2`.

›_ Console

```
# p11tool2 Slot=0 LoginUser=<slot_PIN> ListObjects
```

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


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE           = CKC_X_509
  CKA_UNIQUE_ID                  = 94C0ADCD-A47C-43DC-8258-DE95A096FC4D
  CKA_LABEL                      = webrsa
  CKA_ID                         = 0x77656272 7361 (webrsa)
  CKA_SUBJECT                    =
               0x3061310B 30090603 55040613 02494E31 |0a1 0   U    IN1|
                 0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                 06035504 07130470 756E6531 10300E06 |  U    pune1 0  |
                 0355040A 13077574 696D6163 6F311130 | U    utimaco1 0|
                 0F060355 040B1308 73656375 72697479 |   U    security|
                 3111300F 06035504 03130872 73612064 |1 0   U    rsa d|
                 656D6F                              |emo             |



CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_UNIQUE_ID                  = 179163A3-FA39-48A9-A566-966D04F404B1
  CKA_LABEL                      =
  CKA_ID                         =



CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                   = CKK_RSA
  CKA_UNIQUE_ID                  = 0F1F01B9-966E-4F4D-93ED-FEC069505D79
  CKA_SENSITIVE                  = CK_FALSE
  CKA_EXTRACTABLE                = CK_TRUE
  CKA_LABEL                      =
  CKA_ID                         = 0x77656272 7361 (webrsa)
[root@weblserver ~]#
```

Figure 20 : List keys output using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias webrsa -file rsa.csr -storetype PKCS11 -keystore NONE
-v
```

Provide the keystore password when prompted.

Here:

- NONE is the keystore for HSM.
- SunPKCS11 is the storetype.
- SunPKCS11-CryptoServer is the provider name.
- webrsa is the key name.
- rsa.csr is the CSR file name that will be generated.

5. Get this CSR signed by the CA.
6. Copy the signed certificate on the WebLogic server.
7. Copy the cacerts from `<java_installation_path>/jdk-11.0.6/lib/security/` to `/u01/app/oracle/config/domain/admindomain/`.
8. Import the signed certificate using the command below.

›_ Console

```
# keytool -importcert -alias webrsa -file /home/rsa_demo.p7b -storetype PKCS11

-keystore NONE -providername SunPKCS11-CryptoServer -storepass 12345678
```

```
[root@weblserver ~]# keytool -importcert -alias webrsa -file /home/rsa_demo.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS11-CryptoServer -s
torepass 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: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]

#5: ObjectId: 2.5.29.14 Criticality=false
```

Figure 21 : Import user certificate into keystore

```
#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@weblserver ~]#
```

Figure 22 : Import user certificate into keystore

9. Verify that the `keytool` command shows the signed 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@weblserver ~]# 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: webrsa
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=rsa 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: 46bc22b2720c699a
Valid from: Mon May 29 06:14:00 UTC 2023 until: Wed May 29 06:14:00 UTC 2024
Certificate fingerprints:
         SHA1: FE:51:04:5E:80:12:6C:FA:5A:2F:12:3C:54:27:9E:2D:B2:F5:EA:07
         SHA256: 61:8B:B6:FF:B1:D4:D4:C9:00:F9:EB:8C:A8:C2:5A:04:76:18:34:6D:B9:2B:42:CD:A0:26:FB:BA:37:0F:B5:99
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  codeSigning
]

#2: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 79 00 6B C2 C9 DC DD 16   24 64 3E A8 12 97 46 75  y.k.....$d>...Fu
0010: 66 CE 5B 8D                                        f.[.
]
]
```

Figure 23 : Keytool list output

```
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

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                                        ....
]
]



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

Figure 24 : Keytool list output

7.4 For OpenJDK11 with EC Key

1. Generate an EC keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias weblogiceckey -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.
- weblogiceckey is the key name that will be generated on Utimaco HSM.

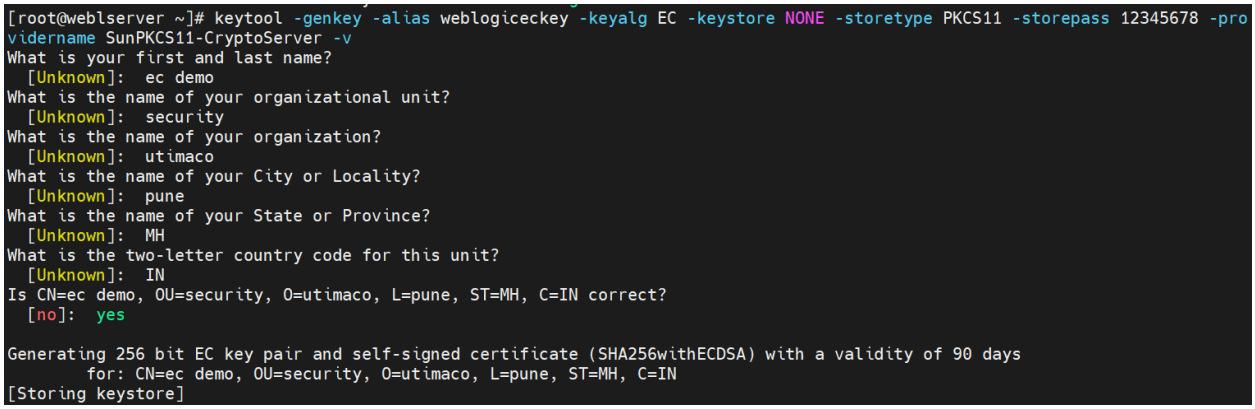

Figure 25 : Key generation using keytool command

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@weblserver ~]# 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: weblogiceckey
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=ec demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Issuer: CN=ec demo, OU=security, O=utimaco, L=pune, ST=MH, C=IN
Serial number: 43b782aa
Valid from: Fri May 26 10:29:34 UTC 2023 until: Thu Aug 24 10:29:34 UTC 2023
Certificate fingerprints:
         SHA1: 1C:15:98:9D:1A:56:14:28:45:4B:19:C0:02:17:B5:55:3A:CA:7E:35
         SHA256: C8:8A:4A:01:8C:AC:BF:20:6E:35:01:E7:31:9E:19:50:6F:04:A9:DB:3A:97:06:66:E0:0C:65:B8:93:4D:DD:88
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: D2 A2 62 BA 95 D7 4F 84   26 09 18 66 C8 5B A0 4E  ..b...O.&..f.[.N
0010: 53 E4 8A 6B                                        S..k
]
]



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

Figure 26 : List keys output

3. List the keys using `p11tool2`.

›_ Console

```
# p11tool2 Slot=0 LoginUser=<slot_PIN> ListObjects
```

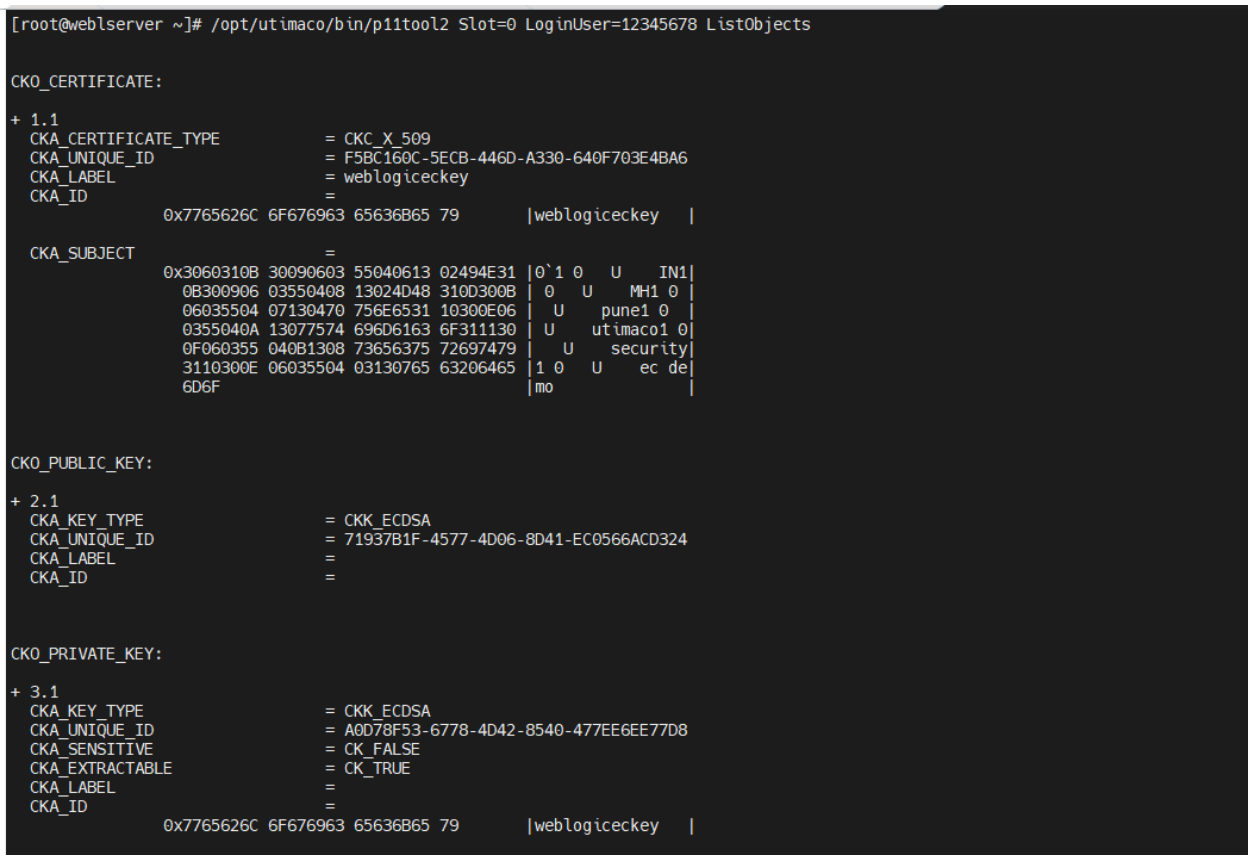

Figure 27 : List keys output using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias weblogiceckey -file ec.csr -storetype PKCS11 -
keystore NONE -v
```

Provide the keystore password when prompted.

Here:

- NONE is the keystore for HSM.
- PKCS11 is the storetype.
- weblogiceckey is the key name.

- ec.csr is the CSR file name that will be generated.

5. Get this CSR signed by the CA.
6. Copy the signed certificate on the WebLogic server.
7. Copy the cacerts from `<java_installation_path>/jdk-11.0.6/lib/security/` to `/u01/app/oracle/config/domain/admindomain/`.
8. Import the signed certificate using the command below.

›_ Console

```
# keytool -importcert -alias weblogiceckey -file /home/ec_demo.p7b -storetype
PKCS11 -keystore NONE -providername SunPKCS11-CryptoServer -storepass 12345678
```

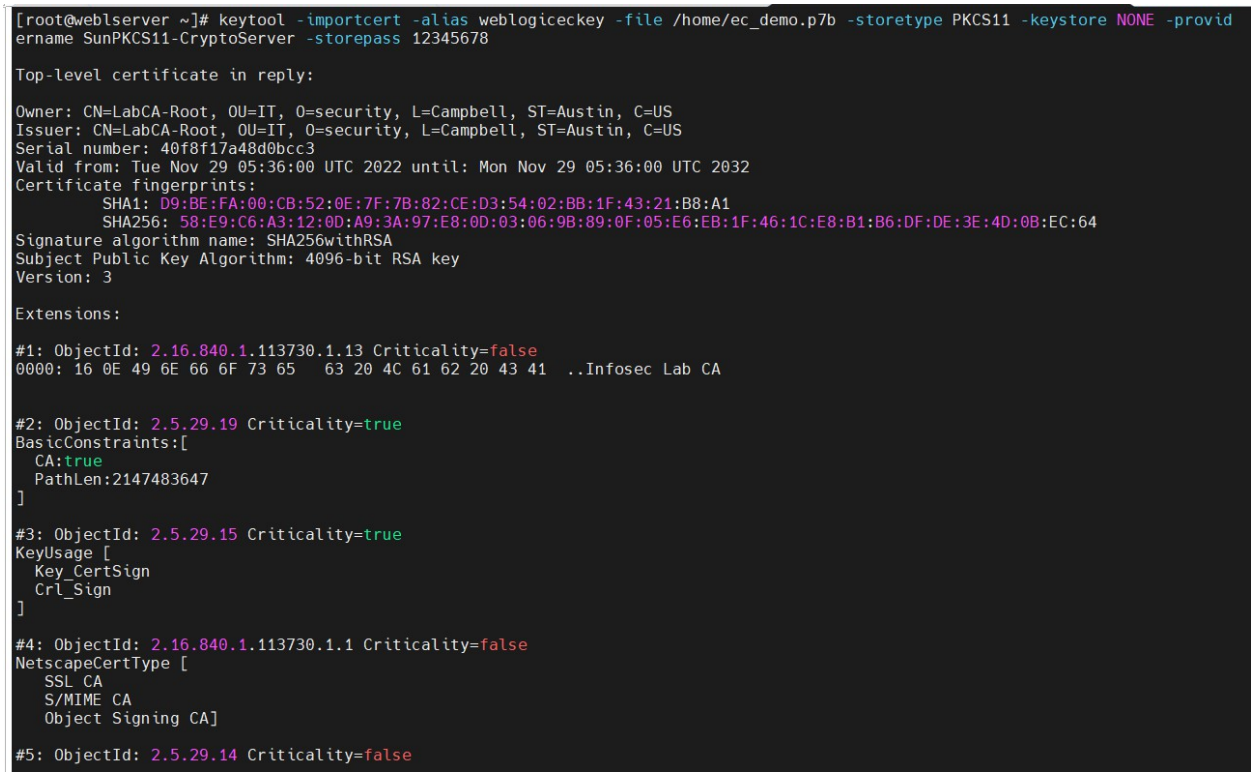

Figure 28 : Import user certificate into keystore

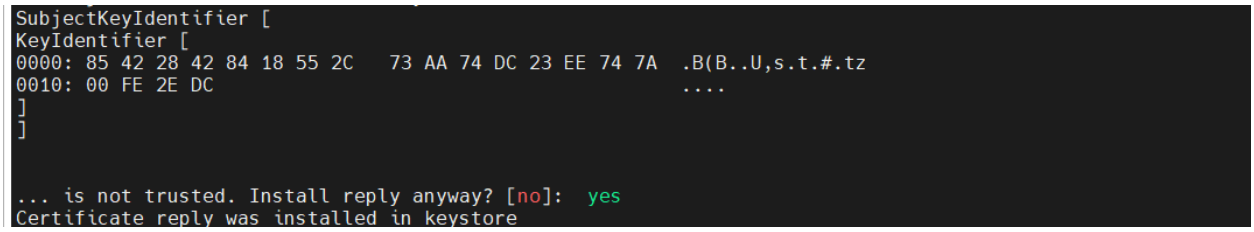

Figure 29 : Import user certificate into keystore

9. Verify that the `keytool` command shows the signed 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@weblserver ~]# 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: weblogiceckey
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=ec 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: 7303345f221b9a74
Valid from: Fri May 26 10:47:00 UTC 2023 until: Sun May 26 10:47:00 UTC 2024
Certificate fingerprints:
         SHA1: DF:31:3F:E9:24:A7:CE:63:B0:15:97:4F:3F:90:3F:57:63:58:8F:06
         SHA256: 5D:AB:3B:35:1F:8B:EC:36:6A:3D:06:B5:DA:30:E4:67:6E:ED:B9:2B:02:0A:07:75:7E:94:3F:25:94:84:86:E6
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  codeSigning
]

#2: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D2 A2 62 BA 95 D7 4F 84   26 09 18 66 C8 5B A0 4E  ..b...O.&..f.[.N
0010: 53 E4 8A 6B                                        S..k
]
]
```

Figure 30 : Keytool list output

```
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

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                                        ....
```

Figure 31 : Keytool list output

7.5 Update Domain Structure in Oracle Web Console to use HSM for SSL

1. Open the Administration Console using the link http://hostname:7001/console. Navigate to **Domain Structure** > **Environment**.
2. Click **Servers**.
3. Click **AdminServer** as shown in the figure.

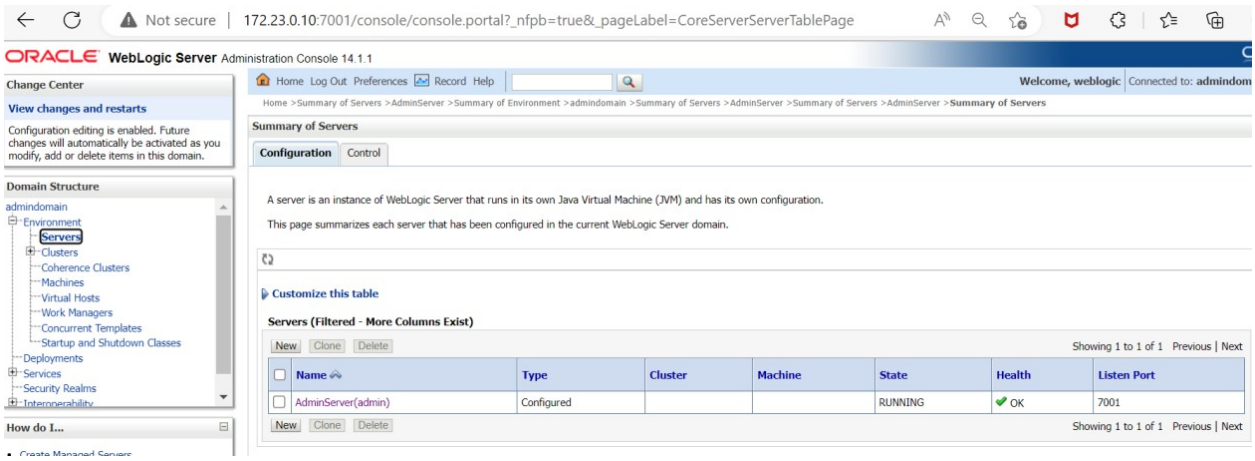

Figure 32 : Updating AdminServer

4. Select the **SSL Listen Port Enabled** checkbox and click **Save**.

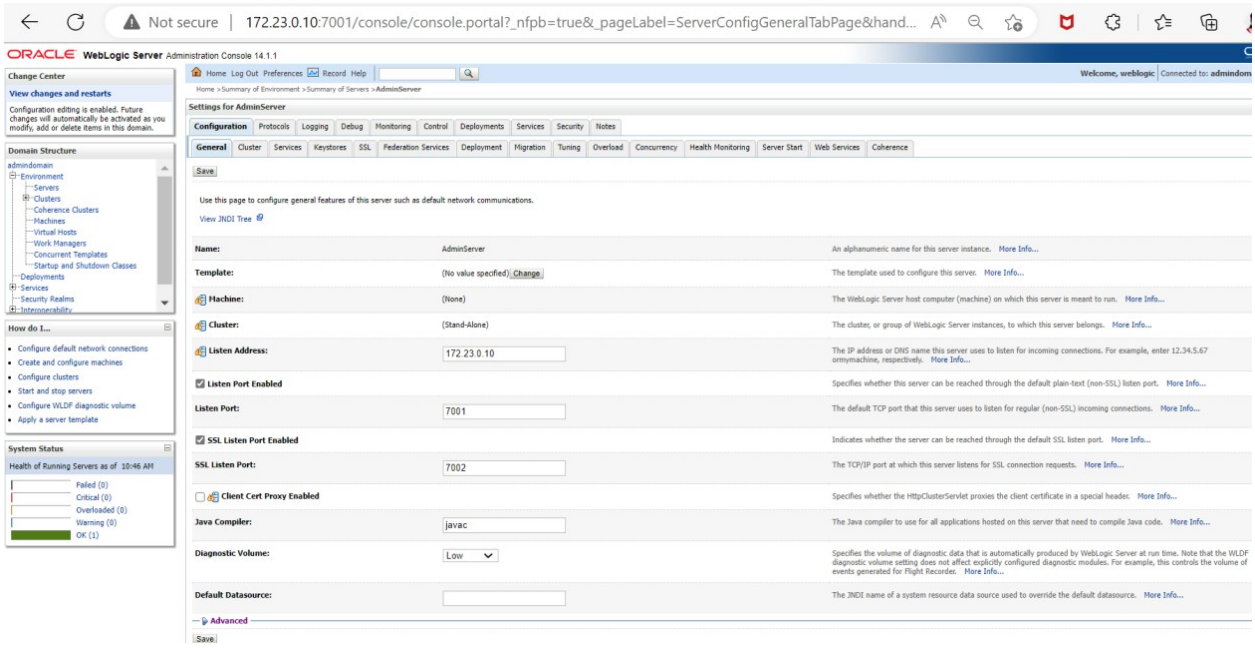

Figure 33 : Enabling SSL configuration

5. Open the **Keystores** tab and then complete the following steps:

   a. Click on the **Change** button.

   b. From the drop-down menu, select **Custom Identity and Custom Trust** and click **Save**.

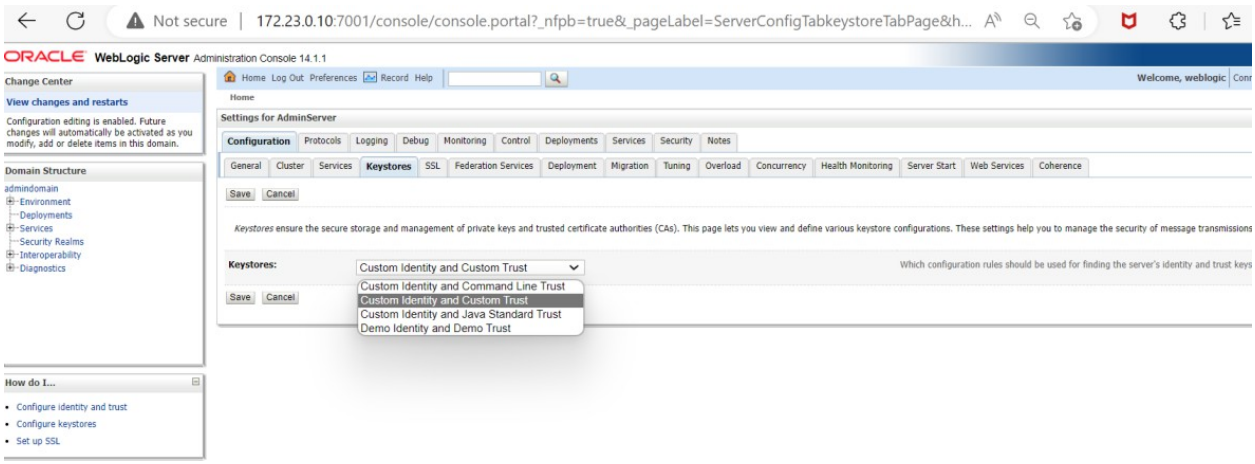

Figure 34 : Updating the Keystore configuration

c. In the **Custom Identity Keystore** field, enter pkcs11.

d. In the **Custom Identity Keystore Type** field, enter pkcs11.

e. In the **Custom Identity Keystore Passphrase** and **Confirm Custom Identity Keystore Passphrase** fields, enter the password for the HSM.

f. In the **Custom Trust Keystore** field, enter cacerts.

g. In the **Custom Trust Keystore Type** field, enter jks.

You cannot set the Custom Trust Keystore Type to pkcs11 as SunPKCS11 does not support the importing of root Certificate itself to HSM, so you have to use cacerts keystore for trusted root certificate authorities.

h. In the **Custom Trust Keystore Passphrase** and **Confirm Custom Trust Keystore Passphrase** fields, enter the password for the cacerts keystore as **changeit** and click **Save** as shown in the screenshot.

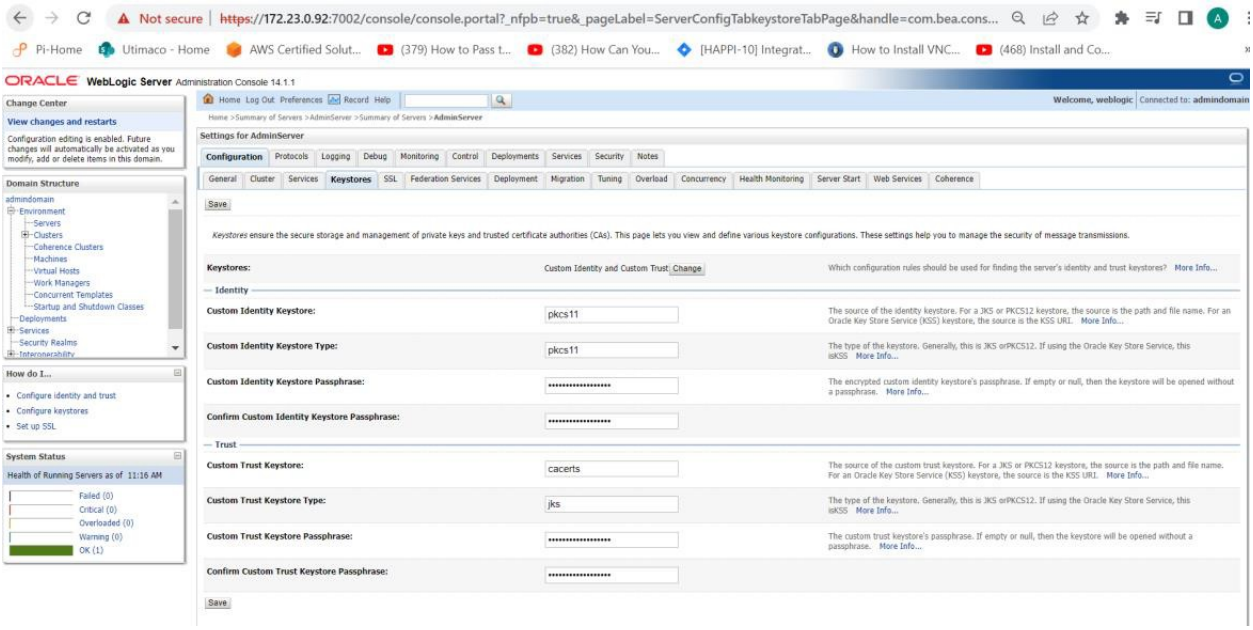

Figure 35 : Updating the Keystore configuration

6. Open the SSL tab.

   a. In the **Private Key Alias** field, enter the name of the SSL key generated on the HSM. For example: weblogiceckey.

   b. In the **Private Key Passphrase** field, provide the slot PIN and click on **Save**.

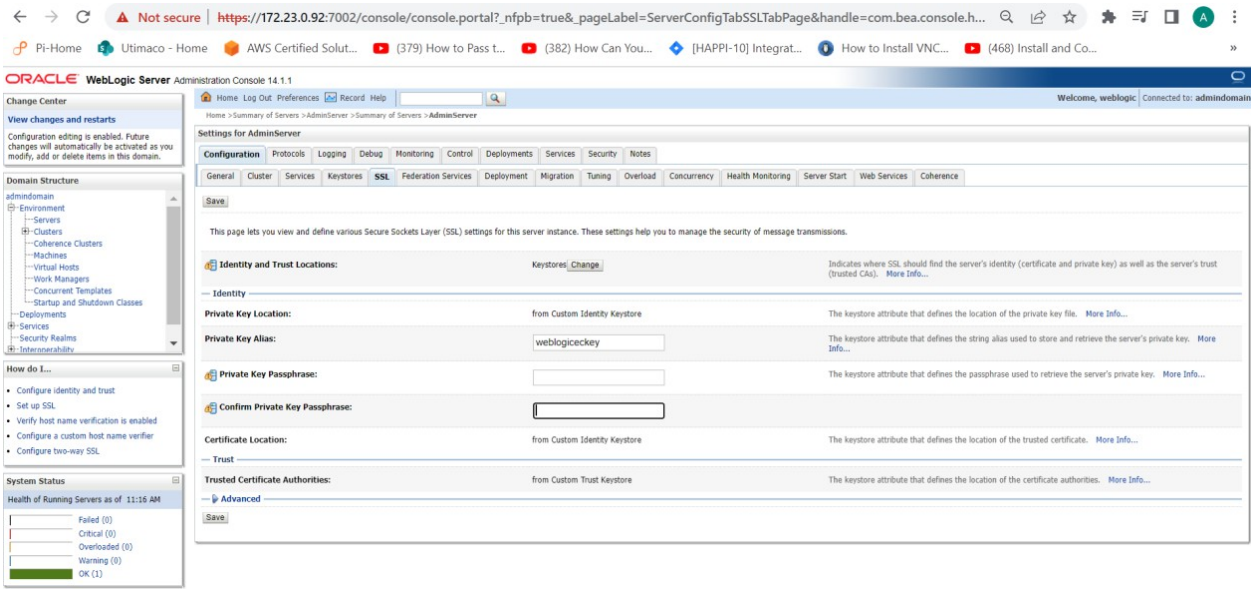

Figure 36 : Updating SSL Private Key

7. Restart the WebLogic server by clicking on **Environment** > **Servers**, click on **AdminServer** and click on **control tab**. Then, click on **restart the SSL**.

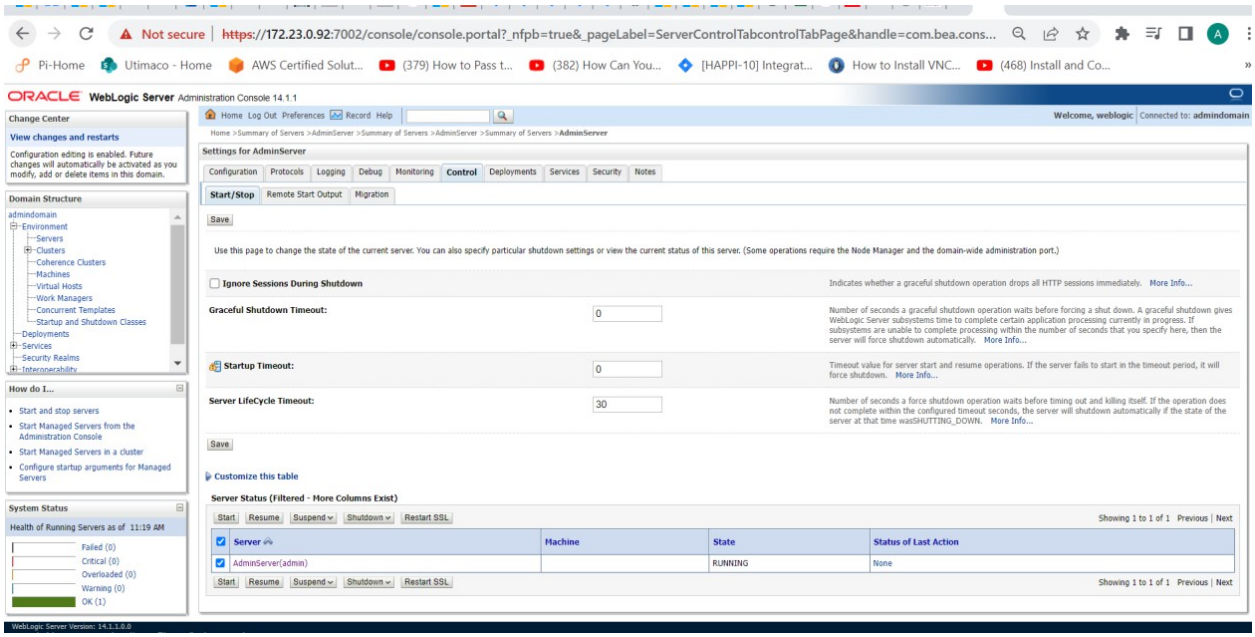

Figure 37 : Restarting SSL

8. After restarting the WebLogic server in the above step, access the **Administration console** over https using https://hostname:7002/console.

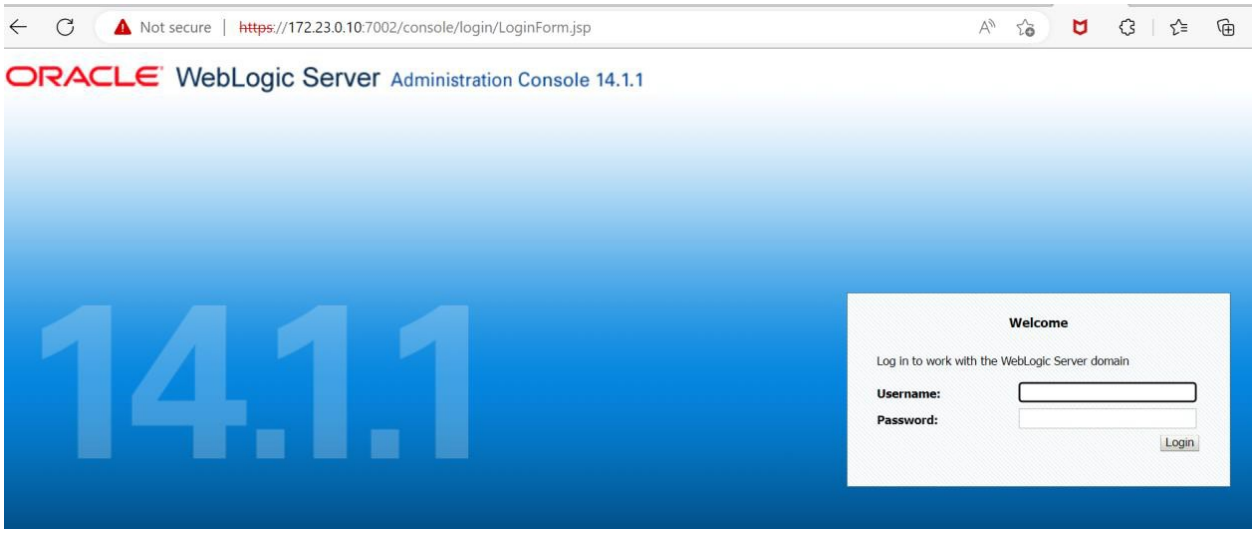

Figure 38 : WebLogic service status over https

This completes the integration of Oracle WebLogic Server with Utimaco SecurityServer using SunPKCS11.

8 Troubleshooting

Error	Diagnosis
After restarting weblogic service and loading the page over SSL, the key alias name is getting altered with some new random alias name. Because of this, whenever you will restart weblogic server next time you will not be able to load the page over SSL.	You need to replace the key alias name from UI with the newly altered key alias name, and restart the weblogic service.
<Error> <Server> <BEA-002606> <The server is unable to create a server socket for listening on channel "DefaultSecure[iiops][1]". The address 192.168.122.1 might be incorrect or another process is using port 7002: java.io.IOException: No identity key/certificate entry was found under alias weblogiceckey in keystore /u01/app/oracle/config/domain/admindomain/pkcs11 on server AdminServer.>	Update the correct alias name of the key from UI, to be used by the server.

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 Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

All CryptoServer product documentation is also available on 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
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-000

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.

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.