Entrust Cryptographic Security Platform Timestamping Authority

nShield® HSM Integration Guide

2025-09-15

Table of Contents

Chapter 1. Introduction

The Entrust Cryptographic Security Platform (CSP) is a versatile and robust virtual appliance that streamlines and simplifies deployment across various environments of the following Entrust solutions: Certificate Authority, CA Gateway, Certificate Enrollment Gateway, Certificate Hub, Timestamping Authority, and Validation Authority. The Entrust CSP - Timestamping Authority (TSA) responds to timestamp requests to prove the existence of certain data before a given time. The Entrust nShield Hardware Security Module (HSM) securely store and manage the timestamp signing key. This document describes how to integrate the TSA with the HSM.

The HSM is available as an appliance or nShield as a Service (nSaaS). Throughout this guide, the term HSM refers to nShield Solo, nShield Connect, and nShield Edge products.

1.1. Product configuration

Entrust tested the integration with the following versions:

Product	Version
Entrust Timestamping Authority	v2.1.1
Entrust Deployment Manager	v2.0.1

1.2. Supported nShield hardware and software versions

Entrust has successfully tested with the following nShield hardware and software versions:

HSM	Security World Software	Firmware	Netimage
nShield 5c	13.6.8	13.4.5 (FIPS 140-3 certified)	13.6.7
Connect XC	13.6.8	12.72.3 (FIPS 140-2 certified)	13.6.7

1.3. Requirements

- Access to the Entrust TrustedCare Portal.
- A dedicated virtual appliance for the installation.

Familiarize yourself with:

- The Entrust Timestamping Authority Documents (use your TrustedCare credentials to log in).
- The nShield documentation.
- Your organizational Certificate Policy, Certificate Practice Statement, and a Security Policy or Procedure covering administration of the Entrust Timestamping Authority and HSM:
- Whether your Security World must comply with FIPS 140 Level 3 or Common Criteria standards. For more information see FIPS 140 Level 3 compliance:
  - The importance of a correct quorum for the Administrator Card Set (ACS) and the policy for managing these cards.
  - The importance of a correct quorum for the Operator Card Set (OCS) and the policy for managing these cards.
  - Key attributes such as key size, time-out, or needed for auditing key usage.
  - Whether to instantiate the Security World as recoverable or not.

Chapter 2. Deploy the Entrust Timestamping Authority

The TSA can be deployed as a stand-alone product, as well as part of the Entrust PKI Hub. This integration deploys and operates the TSA as a stand-alone virtual machine on VMware vSphere.

2.1. Deploy the Entrust Deployment Manager

1. Create a Entrust Deployment Manager (EDM) virtual machine on VMware vSphere. The ISO image is available in the **Software Downloads** tab at Entrust Deployment Manager. Follow the instructions in the **Entrust Deployment Manager 2.0.2 - Installation and Administration Guide** available in the **Documents** tab.
2. Configure the Entrust Deployment Manager (EDM) per section **Configuring the operating system**.
3. Continue with section **Starting up Entrust Deployment Manager**. This integration testing was configured as follows.

    - A single node cluster was deployed. The operation takes several minutes to execute.

      ```
      [sysadmin@timestamping-auth-edm ~]$ sudo clusterctl install --mode single-node
      [sudo] password for sysadmin:
      Installing  done ┤|                                                     |├ 100 %
      ```

    - The default TLS certificate was kept.
    - Both Management Console and Grafana default passwords were changed.

2.2. Deploy the Entrust Timestamping Authority

Two methods are available to deploy the TSA: through the Management console, and with the `clusterctl` command-line tool provided by the EDM. We utilized the `clusterctl` command-line tool in this integration.

1. Download the TSA files to the EDM virtual machine created above. The ISO image, CLI, and configuration file example are available in the **Software Downloads** tab at Entrust Timestamping Authority. Follow the instructions in the **Entrust Timestamping Authority 2.1 Deployment Guide** available in the **Documents** tab.

   Example of downloaded software before installation:

```
[sysadmin@timestamping-auth-edm ~]$ ls -al /usr/local/bin
total 48
drwxr-xr-x.  2 root     root  4096 May  7 19:24 .
drwxr-xr-x. 12 root     root  4096 May  7 14:44 ..
-rwxr-x---.  1 sysadmin edm  37344 May  7 18:33 tsactl

[sysadmin@timestamping-auth-edm ~]$ ls -al /home/sysadmin/Downloads/
total 2348404
drwxr-x---. 2 sysadmin edm         50 May  7 19:24 .
drwx------. 3 sysadmin edm        100 May  7 18:28 ..
-rw-r-----. 1 sysadmin edm 2404761327 May  5 18:23 tsa-2.1.1.sln
-rw-r-----. 1 sysadmin edm        430 May  7 18:46 tsa-config.json
```

2. Download the license file to the EDM virtual machine. See the **Entrust Timestamping Authority 2.1 Deployment Guide** available in the **Documents** tab for product licensing information.

```
[sysadmin@timestamping-auth-edm ~]$ ls -al /home/sysadmin/Downloads/
total 2348408
drwxr-x---. 2 sysadmin edm         74 May  8 13:49 .
drwx------. 3 sysadmin edm        100 May  7 14:28 ..
-rw-r-----. 1 sysadmin edm       3975 May  6 15:52 inttesttsa00.lic
-rw-r-----. 1 sysadmin edm 2404761327 May  5 14:23 tsa-2.1.1.sln
-rw-r-----. 1 sysadmin edm        430 May  7 14:46 tsa-config.json
```

3. Register the TSA.

```
[sysadmin@timestamping-auth-edm ~]$ sudo clusterctl solution register --file
/home/sysadmin/Downloads/inttesttsa00.lic
[sudo] password for sysadmin:
tsa registered
```

4. Uploads the TSA to the Management Console endpoint.

```
[sysadmin@timestamping-auth-edm ~]$ sudo clusterctl solution upload -i tsa -f /home/sysadmin/Downloads/tsa-
2.1.1.sln
[sudo] password for sysadmin:
Uploading  done ┤║                                                              ║├ 100 %
Processing  done ┤║                                                              ║├ 100 %
tsa: version 2.1.1 uploaded. A redeploy of the tsa solution is required for changes to take effect
```

2.3. Configure the TSA

1. Open the following ports.

```
# sudo firewall-cmd --zone=public --permanent --add-port=323/udp
[sudo] password for sysadmin:
success
# sudo firewall-cmd --zone=public --permanent --add-port=80/tcp
success
# sudo firewall-cmd --reload
success
```

2. Edit the `/etc/chrony.conf` file following the instructions in the *Entrust Timestamping Authority 2.1 Deployment Guide*, section *Configuring chrony*.

```
[sysadmin@timestamping-auth-edm ~]$ cat /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
pool 2.rhel.pool.ntp.org iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

...

# TSA configuration
bindcmdaddress <IP of Entrust Entrust Deployment Manager virtual machine>
cmdallow all
```

3. Restart the chrony service.

```
[sysadmin@timestamping-auth-edm ~]$ sudo systemctl restart chronyd.service
```

Chapter 3. Install and configure the Entrust nShield HSM

3.1. Install the Entrust nShield HSM

Install the HSM locally, remotely, or remotely via the serial console. Condensed instructions are available in the following Entrust nShield Support articles.

- How To: Locally Set up a new or replacement nShield Connect.
- How To: Remotely Setup a new or replacement nShield Connect.
- How To: Remotely Setup a new or replacement nShield Connect XC Serial Console Model.

For detailed instructions see the nShield v13.6.8 Hardware Install and Setup. Guides.

3.2. Install the Security World software and create a Security World

1. Install the Security World software. For detailed instructions see the nShield Security World Software v13.6.8 Installation Guide.
2. Add the Security World utilities path to the system path. This path is typically `/opt/nfast/bin`:

   ```
   # sudo vi /etc/profile.d/nfast.sh
   ```

   Add the following info to `nfast.sh` and save:

   ```
   # Entrust Security World path variable
   export PATH=$PATH:/opt/nfast/bin
   ```

3. Open firewall port 9004 for the HSM connections:

   ```
   # sudo firewall-cmd --permanent --add-port=9004/tcp
   [sudo] password for sysadmin:
   success
   # sudo firewall-cmd --reload
   success
   ```

4. If using remote administration, open firewall port 9005 for the Entrust nShield Trusted Verification Device (TVD). Otherwise skip this step.

```
# sudo firewall-cmd --permanent --add-port=9005/tcp
[sudo] password for sysadmin:
success
# sudo firewall-cmd --reload
success
```

5. Open a command window and run the following utility to confirm that the HSM is **operational**:

```
>enquiry
Server:
 enquiry reply flags  none
 enquiry reply level  Six
 serial number        8FE1-B519-C5AA
 mode                 operational
...
Module #1:
 enquiry reply flags  UnprivOnly
 enquiry reply level  Six
 serial number        8FE1-B519-C5AA
 mode                 operational
...
```

6. Create your Security World if one does not already exist or copy an existing one. Follow your organization’s security policy for this. For more information see Create a new Security World.

   ACS cards cannot be duplicated after the Security World is created. You may want to create extras in case of a card failure or a lost card.

This is an example of the steps to copy an existing world from another server. The `world` and module files were first copied to the `/home/sysadmin/Download` directory from an external machine. Then these were copied to the `/opt/nfast/kmdata/local` directory. Notice the ownership.

```
[sysadmin@timestamping-auth-edm local]$ ls -al /home/sysadmin/Downloads/world
-rw-r-----. 1 sysadmin edm 40860 Nov  6  2024 /home/sysadmin/Downloads/world
[sysadmin@timestamping-auth-edm local]$ ls -al /home/sysadmin/Downloads/module_8FE1-B519-C5AA
-rw-r-----. 1 sysadmin edm 3716 Apr  2 15:14 /home/sysadmin/Downloads/module_8FE1-B519-C5AA

[sysadmin@timestamping-auth-edm local]$ sudo cp /home/sysadmin/Downloads/world /opt/nfast/kmdata/local/.
[sudo] password for sysadmin:
[sysadmin@timestamping-auth-edm local]$ sudo cp /home/sysadmin/Downloads/module_8FE1-B519-C5AA
/opt/nfast/kmdata/local/.
[sudo] password for sysadmin:

[sysadmin@timestamping-auth-edm local]$ ls -al /opt/nfast/kmdata/local/
total 52
drwxrwsr-x. 2 nfast nfast  4096 May  9 09:52 .
drwxrwsr-x. 7 nfast nfast  4096 May  8 17:13 ..
-rw-r-----. 1 root  nfast  3716 May  9 09:52 module_8FE1-B519-C5AA
-rw-r-----. 1 root  nfast 40860 May  9 09:40 world
```

1. Confirm that the Security World is "Usable*:

```
> nfkminfo
World
 generation  2
 state       0x3737000c Initialised Usable ...
 ...
Module #1
 generation 2
 state      0x2 Usable
 ...
```

== Select the protection method

The OCS or Softcard and associated passphrase will be used to authorize access to specific keys protected by the HSM.

- Operator Cards Set (OCS) are smartcards that are presented to the physical smartcard reader of an HSM. For more information on OCS use, properties, and K-of-N values, see Operator Card Sets (OCS).
- Softcards are logical tokens (passphrases) that protect they key and authorize its use. For more information on Softcards use see Softcards.

Follow your organization's security policy to select an authorization access method.

1. Create file `/opt/nfast/cknfastrc` containing the nShield PKCS #11 library environment variables per the selection above.

```
# Enable Softcard protection
CKNFAST_LOADSHARING=1

# OCS Preload file location and card set state
#NFAST_NFKM_TOKENSFILE=/tmp/preloadtoken
#CKNFAST_NONREMOVABLE=1

# PKCS #11 log level
CKNFAST_DEBUG=10
```

The Kubernetes implementation of the TSA puts some restrictions in the location of the `preloadtoken` file. Also the PKCS #11 log traditionally written to `/opt/nfast/log/pkcs11.log` will now be available as described in Troubleshoot. Some log info is also written to `/var/log/entrust/tsa/tsactl.log`. These log contains both the PKCS #11 and TSA log info intertwined.

1. Change group ownership of `/opt/nfast/cknfastrc` to `nfast`.

```
[sysadmin@timestamping-auth-edm log]$ sudo chown root:nfast /opt/nfast/cknfastrc
```

3.3. Create the Operator Card Set (OCS) or Softcard

Typically, an organization’s security policies dictate the use of one or the other.

3.3.1. Create the OCS

> After an OCS card set has been created, the cards cannot be duplicated. You may want to create extras in case of a card failure or a lost card.
>
> Add the **-p** (persistent) option to the `createocs` command to be able to encrypt/decrypt the database after the OCS card has been removed for safe storage from either the HSM front panel slot or from the TVD. See the Preload Utility for more information.
>
> Recovering from a power failure requires the OCS to be inserted in the HSM or the TVD.
>
> The authentication provided by the OCS as shown in the command in this section is non-persistent and only available while the OCS card is inserted in the HSM front panel slot or the TVD. If the TVD loses connection to the Remote Administration client the HSM will be inaccessible.

The Entrust Timestamping Authority maps one protecting token to one stored passphrase. It can store information for only one token at a time. Therefore an OCS card set quorum K must be one.

1. Edit file `/opt/nfast/kmdata/config/cardlist` to add the serial number of the card(s) to be presented or the wildcard value.
2. Run the `createocs` utility as described below. Enter a passphrase or password at the prompt. Use the same passphrase for all the OCS cards in the set (one for each person with access privilege, plus the spares). In this example note that slot 2, remote via a TVD, is used to present the card.

```
> createocs -m1 -s2 -N testOCS -Q 1/1

FIPS 140-2 level 3 auth obtained.

Creating Cardset:
 Module 1: 0 cards of 1 written
 Module 1 slot 0: Admin Card #1
 Module 1 slot 2: blank card
 Module 1 slot 3: empty
 Module 1 slot 2:- passphrase specified - writing card
Card writing complete.

cardset created; hkltu = edb3d45a28e5a6b22b033684ce589d9e198272c2
```

3. Verify the OCS created:

```
> nfkminfo -c
Cardset list - 1 cardsets:  (P)ersistent/(N)ot, (R)emoteable/(L)ocal-only
 Operator logical token hash               k/n timeout  name
 edb3d45a28e5a6b22b033684ce589d9e198272c2  1/1  none-NL testOCS
```

3.3.2. Create the Softcard

The Entrust Timestamping Authority maps one protecting token to one stored passphrase. Softcards are singular and do not have a quorum, so the Entrust Timestamping Authority credential matches them quite well.

Unlike OCS protection, which requires a smart card and a passcode, a softcard does not require additional input for recovery after a power failure.

1. Verify the `/opt/nfast/cknfastrc` file exists and contains the following variable. Otherwise, create it.

```
# Enable Softcard protection
CKNFAST_LOADSHARING=1
```

2. Execute the following command. Enter a passphrase at the prompt.

```
> ppmk -n testSC

Enter new pass phrase:
Enter new pass phrase again:
New softcard created: HKLTU 925f67e72ea3c354cae4e6797bde3753d24e7744
```

3. Verify the Softcard created:

```
> nfkminfo -s
SoftCard summary - 1 softcards:
 Operator logical token hash               name
 925f67e72ea3c354cae4e6797bde3753d24e7744  testSC
```

The `rocs` utility shows the OCS and Softcard created:

```
> rocs
`rocs' key recovery tool
Useful commands: `help', `help intro', `quit'.
rocs> list cardset
No. Name                     Keys (recov) Sharing
  1 testOCS                  0 (0)        1 of 5
  2 testSC                   0 (0)        (softcard)
rocs> quit
```

3.4. Prepare files to be loaded into TSA

1. Copy the `/opt/nfast/cknfastrc` file to `/opt/nfast/kmdata/`.

```
[sysadmin@timestamping-auth-edm ~]$ sudo cp /opt/nfast/cknfastrc /opt/nfast/kmdata/.
```

2. Create the `/tmp/preloadtoken` file described in [install-entrust-hsm:::select-protection-method] `/opt/nfast/cknfastrc` file. Change ownership to `nfast`.

```
[sysadmin@timestamping-auth-edm ~]$ sudo touch /tmp/preloadtoken
[sudo] password for sysadmin:

[sysadmin@timestamping-auth-edm ~]$ sudo chown nfast:nfast /tmp/preloadtoken
```

3. Change the permissions of directory `/var/log/entrust/tsa` containing the PKCS #11 log info.

```
[sysadmin@timestamping-auth-edm entrust]$ sudo chmod 777 /var/log/entrust/tsa
```

4. Add the `sysadmin` user to the `nfast` group.

```
[sysadmin@timestamping-auth-edm nfast]$ sudo usermod -a -G nfast sysadmin
[sudo] password for sysadmin:
```

Chapter 4. Integrate the Entrust Timestamping Authority with the nShield HSM

4.1. Create a certificate request

1. Run the following commands in the EDM to load the HSM configuration.

```
[sysadmin@timestamping-auth-edm ~]$ sudo /usr/local/bin/tsactl import-nshield -f /opt/nfast/kmdata
[sudo] password for sysadmin:
If there is a kmdata in TSA, it will be overwritten. Created keys will be lost. Continue? [y/N]: y
Setting  done ┤|                    |├ 100 %
Secret(s) established. A redeploy of the tsa solution is required for changes to take effect
Importing nShield...                          Done
```

2. Present the OCS if using OCS protection. The OCS must be presented in slot 0 which means: insert the OCS in the HSM front panel reader, or map the TVD to slot 0 as described in Map dynamic slots to slot #0

```
[sysadmin@timestamping-auth-edm ~]$ /opt/nfast/bin/nfkminfo
World
 generation  2
 state       0x3737000c Initialised Usable

...

Module #1 Slot #0 IC 217
 generation    1
 phystype      SmartCard
 slotlistflags 0x2 SupportsAuthentication
 state         0x5 Operator
 flags         0x10000
 shareno       2
 shares        LTU(PIN) LTFIPS
 error         OK
Cardset
 name          "testOCS"
 k-out-of-n    1/5
 flags         NotPersistent PINRecoveryForbidden(disabled) !RemoteEnabled
 timeout       none
 card names    "" "" "" "" ""
 hkltu         edb3d45a28e5a6b22b033684ce589d9e198272c2
 gentime       2023-07-20 18:50:48
```

3. Create the key pair and the certificate signing request (CSR) of the certificate for signing TSA responses.

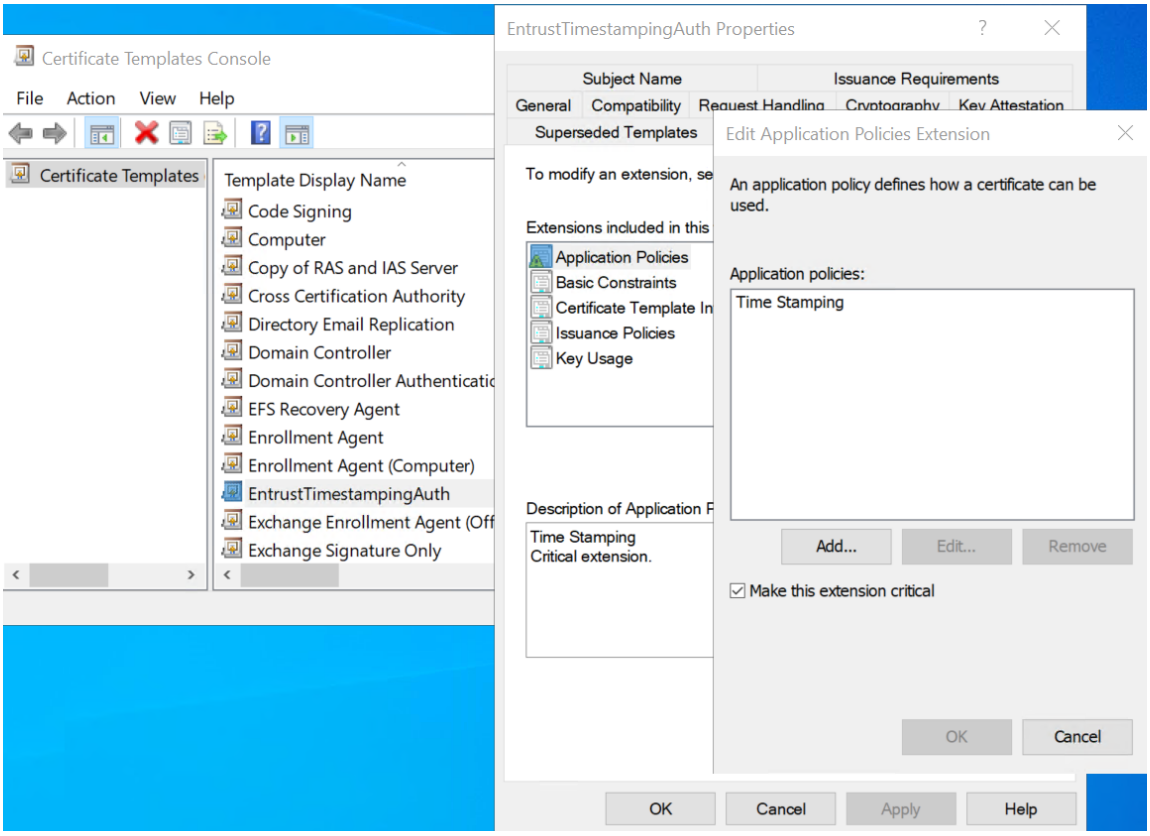
```
[sysadmin@timestamping-auth-edm ~]$ sudo /usr/local/bin/tsactl create-key -k RSA2048 -s "CN=TSA" -o
/tmp/tsa-cert-request.txt -t testOCS -v nshield
tsa-cert-request.txt file already exists. Do you want to overwrite it? [y/N]: y
Obtaining necessary components for TSA...     Done
```

```
Enter HSM PIN:
Starting PKCS #11 Manager...                    Done
2025-05-14 20:47:58 [0007]: pkcs11: 00000000 >>   C_GetFunctionList

...

2025-05-14 20:47:58 [0007] tc0b6ffd10d7f0000: pkcs11: 00000000 <    rv 0x00000000
Setting  done ┤|                       |├ 100 %
Secret(s) established. A redeploy of the tsa solution is required for changes to take effect
CSR written in path: tsa-cert-request.txt
```

4.2. Sign the certificate request

1. Create a timestamping certificate template in your CA to sign the `/tmp/tsa-cert-request.txt` CSR created in Create a certificate request. A certificate template named **EntrustTimestapingAuth** was created by copying the **Web Server** certificate. The **Server Authentication** application policy was removed from the **Extensions**. The **Time Stamping** application policy was added to the **Extensions**. This extension was made critical.

   The PKI used in this testing consisted of a root CA and a subordinate CA. The certificate template create above and signing below were done at the subordinate CA.



Time Stamping
Critical extension.

An application policy defines how a certificate can be used.

Application policies:
Time Stamping

☑ Make this extension critical

2. Sign the `/tmp/tsa-cert-request.txt` CSR. The signed certificate file, `tsa-signed-cert.cer` in this example, must contain a certificate in PEM format and Base64

encoding.

```
C:\Users\Administrator.INTEROP\Downloads>certreq -submit -attrib
"CertificateTemplate:EntrustTimestampingAuth" tsa-cert-request.txt
Active Directory Enrollment Policy
  {96E14557-DDD4-48BD-BE1A-AA453F20D859}
  ldap:
RequestId: 17
RequestId: "17"
Certificate retrieved(Issued) Issued
```

3. Print the certificate in text form. Notice the extended key usage.

```
C:\Users\xxxxxxxx\Downloads>openssl x509 -text -noout -verify -in tsa-signed-cert.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            39:00:00:00:13:82:ce:22:2f:09:f3:52:7f:00:00:00:00:00:13
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC = local, DC = interop, CN = interop-INTEROP-SUB-CA-CA
        Validity
            Not Before: May 23 15:48:40 2025 GMT
            Not After : May 23 15:48:40 2027 GMT
        Subject: CN = TSA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d2:f1:6a:ec:9a:f0:f2:66:b9:f0:dd:21:f7:0a:

                    ...

                    46:23:e8:74:c7:8d:c5:e9:cb:87:77:d5:a2:16:25:
                    f6:e1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                E1:36:EF:65:3A:81:F8:AB:12:CF:B6:57:D6:50:DD:FA:21:80:F6:A5
            X509v3 Authority Key Identifier:
                DE:07:BB:92:75:2C:43:F4:BC:2F:9F:D5:3D:2C:00:79:C7:6A:27:B9
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:ldap:///CN=interop-INTEROP-SUB-CA-CA,CN=interop-sub-
ca,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=interop,DC=local?certificateRevocation
List?base?objectClass=cRLDistributionPoint
            Authority Information Access:
                CA Issuers - URI:ldap:///CN=interop-INTEROP-SUB-CA-
CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=interop,DC=local?cACertificate?base?ob
jectClass=certificationAuthority
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            1.3.6.1.4.1.311.21.7:
                0/.'+.....7.....x...X...(.......V.e...C...W..d...
            X509v3 Extended Key Usage: critical
                Time Stamping
            1.3.6.1.4.1.311.21.10: critical
                0.0
..+.......
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        8d:20:50:30:64:4b:4a:29:ae:63:cd:3f:4a:3e:75:87:d1:12:
```

```
        ...

        c4:db:29:cf
```

4. Save the certificate file `tsa-signed-cert.cer`.

4.3. Export your root CA certificate

Your certificate chain will be needed to verify the time stamp in Test the integration.

1. Export the certificates of your CAs in the chain. The exported certificates were copied to a server, first introduced now, which will be used later to request the time stamp.

```
C:\Users\Administrator\Downloads>dir interop-root-ca.cer interop-sub-ca.cer
 Volume in drive C has no label.
 Volume Serial Number is 86CD-3DFE

 Directory of C:\Users\Administrator\Downloads

05/27/2025  02:41 PM               793 interop-root-ca.cer

 Directory of C:\Users\Administrator\Downloads

05/27/2025  02:27 PM             1,360 interop-sub-ca.cer
               2 File(s)          2,153 bytes
               0 Dir(s)  51,288,801,280 bytes free
```

2. Convert these certificates to `pem` format.

```
C:\Users\Administrator\Downloads>certutil.exe -encode interop-sub-ca.cer interop-sub-ca.pem
Input Length = 1360
Output Length = 1930
CertUtil: -encode command completed successfully.
```

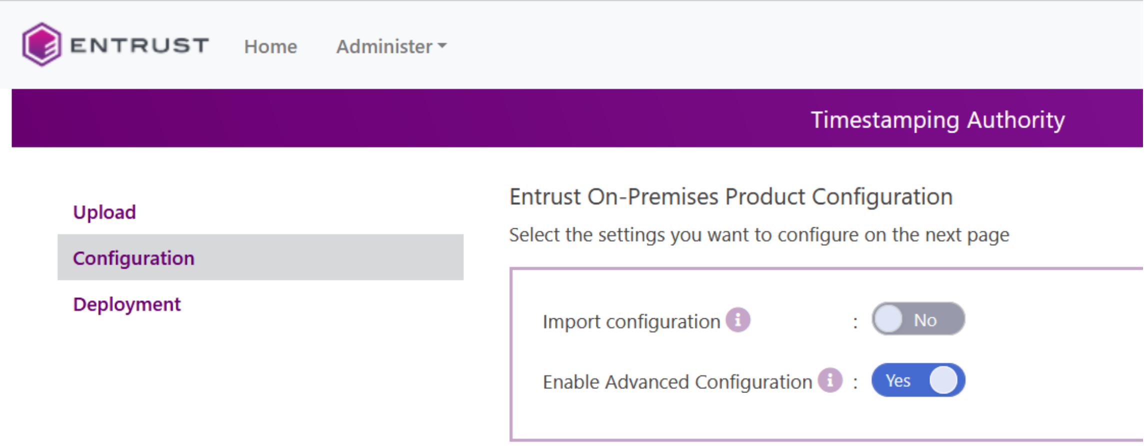
3. Create the certificate chain.

```
C:\Users\Administrator\Downloads>type interop-root-ca.pem interop-sub-ca.pem > chain.pem

interop-root-ca.pem

interop-sub-ca.pem
```

Chapter 5. Configure the Entrust Timestamping Authority

The following steps are described in detail in the **Entrust Timestamping Authority 2.1 Deployment Guide**, section **Configuring Entrust Timestamping Authority**.

1. Present the OCS to the HSM if using OCS protection. The OCS must be presented in slot 0 which means: insert the OCS in the HSM front panel reader, or map the TVD to slot 0 as described in Map dynamic slots to slot #0
2. On a web browser navigate to https://<machine>/management-console, where <machine> is the EDM hostname or IP address. Log in as a user with Timestamping Authority management permissions, for example **admin**.
3. On the **Timestamping Authority** icon, select **Manage Solution**.
4. Select **Configuration**.
5. Active **Enable Advance Configuration**. Then select **Next**.



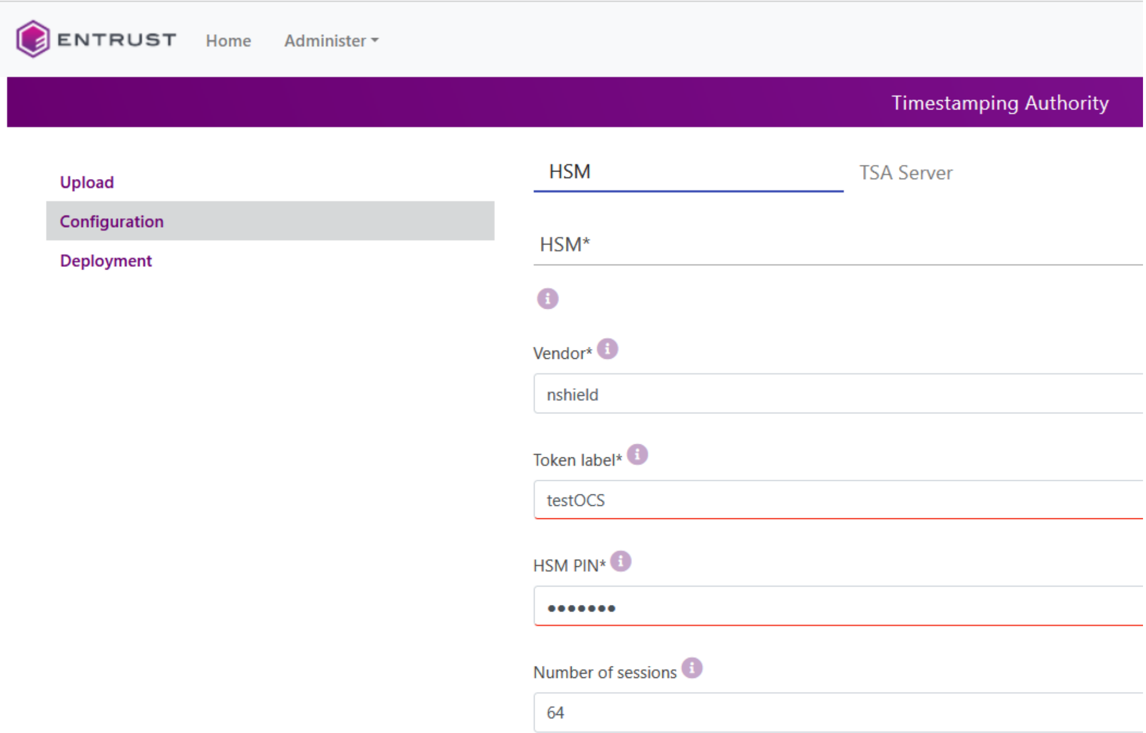
Entrust On-Premises Product Configuration

Select the settings you want to configure on the next page

Import configuration : No

Enable Advanced Configuration : Yes

6. In the **HSM** tab, in the vendor pull-down menu, select **nShield**. Enter the token name and passphrase. For **Number of sessions**, the default 64 was used. Then select **Next**.



HSM*

Vendor*

nshield

Token label*

testOCS

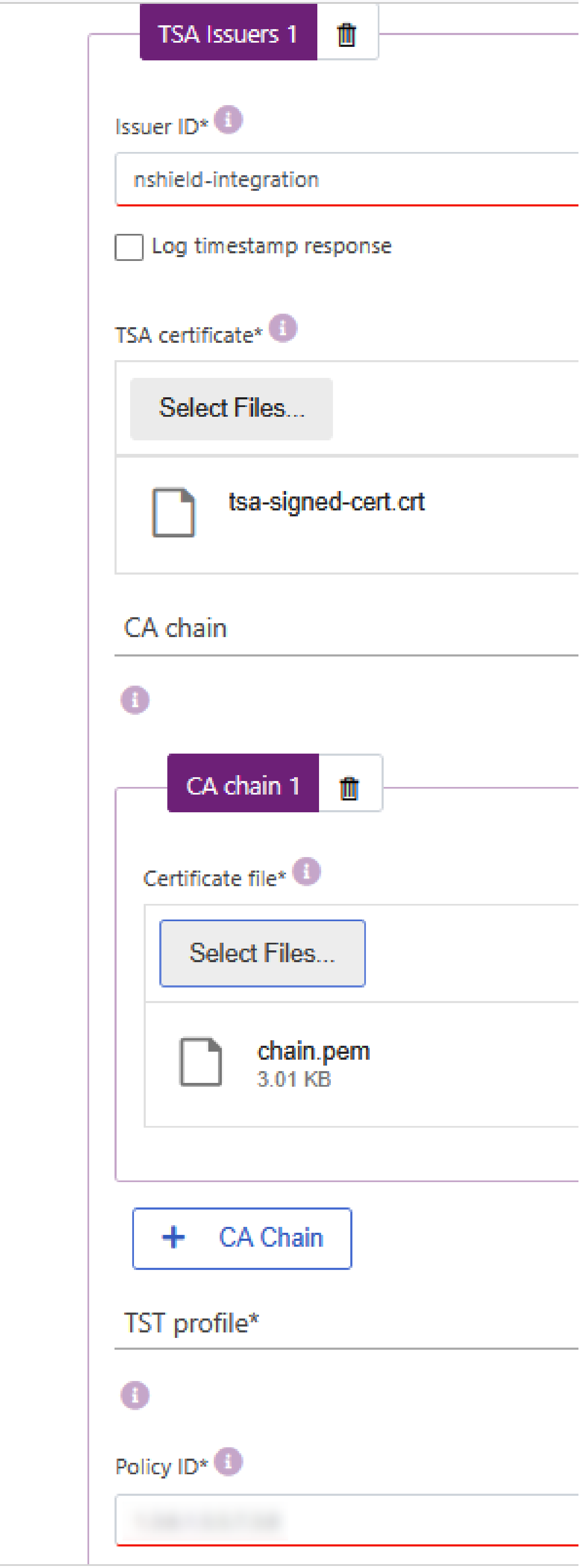
HSM PIN*

•••••••

Number of sessions

64

7. In the **TSA Server** tab, make you selection. Defaults were used. Then select **Next**.
8. In the **Clock Service** tab, make you selection. Defaults were used. Then select **Next**.
9. In the **TSA Issuers** tab, select the **+ TSE Issuers** icon.
10. Enter the issuer ID or name.
11. In **TSA certificate**, select the `tsa-signed-cert.cer` file created in Sign the certificate request.
12. In **CA Chain**, select the `chain.pem` file created in Export your root CA certificate.
13. In **Policy ID**, enter the RFC 3161 Time-Stamp Protocol (TSP) object identifier values of **1.3.6.1.5.5.7.3.8**.
14. The default values were selected for the other fields in this window. Then select **Validate**.



Issuer ID*: nshield-integration

□ Log timestamp response

TSA certificate*: tsa-signed-cert.crt

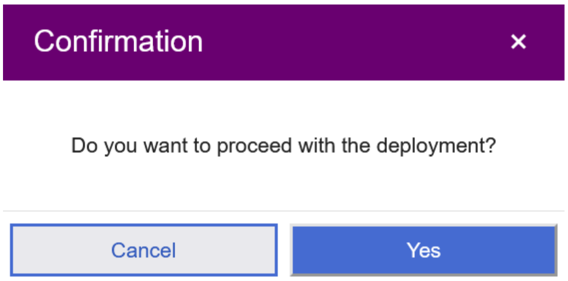
CA chain

Certificate file*: chain.pem 3.01 KB

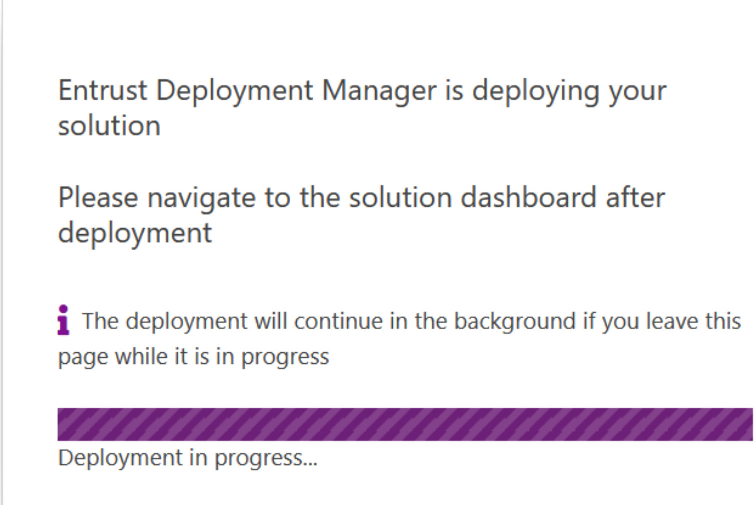
TST profile*

Policy ID*: [redacted]

15. Upon no validation errors, select **Submit**.
16. Upon no submissions errors, select **Deploy**. Then select **Yes**.



Do you want to proceed with the deployment?



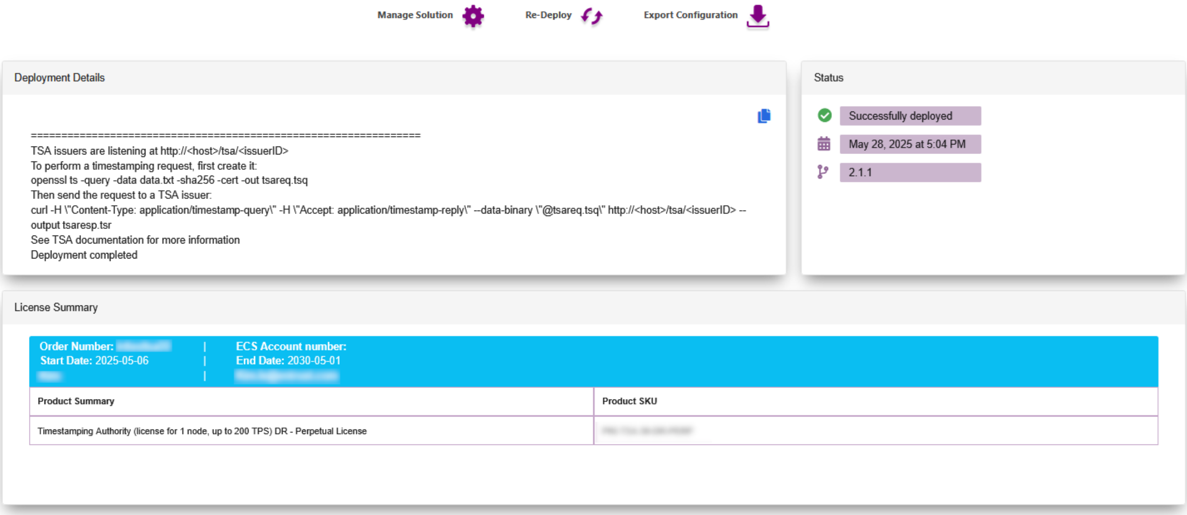
Entrust Deployment Manager is deploying your solution

Please navigate to the solution dashboard after deployment

ℹ The deployment will continue in the background if you leave this page while it is in progress

Deployment in progress...

Example of a successful deployment:



Deployment Details

```
================================================================
TSA issuers are listening at http://<host>/tsa/<issuerID>
To perform a timestamping request, first create it:
openssl ts -query -data data.txt -sha256 -cert -out tsareq.tsq
Then send the request to a TSA issuer:
curl -H \"Content-Type: application/timestamp-query\" -H \"Accept: application/timestamp-reply\" --data-binary \"@tsareq.tsq\" http://<host>/tsa/<issuerID> --
output tsaresp.tsr
See TSA documentation for more information
Deployment completed
```

Status

✓ Successfully deployed

May 28, 2025 at 5:04 PM

2.1.1

License Summary

Order Number: [redacted] | ECS Account number:
Start Date: 2025-05-06 | End Date: 2030-05-01
[redacted] | [redacted]

Product Summary	Product SKU
Timestamping Authority (license for 1 node, up to 200 TPS) DR - Perpetual License	[redacted]

Chapter 6. Test the integration

Testing consists of time stamping a file as described in the **Entrust Timestamping Authority 2.1 Deployment Guide**, section **Testing the timestamping service**. The time stamping request comes from the server first introduced in Export your root CA certificate.

1. Select a file to be time stamped.

```
C:\Users\Administrator\Downloads>type hello-world.txt
hello, world
```

2. Create a `tsq` (time stamp request) file, which contains a hash of the file created above to be signed.

```
C:\Users\Administrator\Downloads>openssl ts -query -data hello-world.txt -sha512 -cert -out hello-world.tsq
Using configuration from C:\Program Files\nCipher\nfast\\openssl\openssl.cnf
```

3. Validate the time stamp request file.

```
C:\Users\Administrator\Downloads>openssl ts -query -in hello-world.tsq -text
Using configuration from C:\Program Files\nCipher\nfast\\openssl\openssl.cnf
Version: 1
Hash Algorithm: sha512
Message data:
    0000 - 87 10 33 9d cb 68 14 d0-d9 d2 29 0e f4 22 28 5c   ..3..h....).."(\
    0010 - 93 22 b7 16 39 51 f9 a0-ca 8f 88 3d 33 05 28 6f   ."..9Q.....=3.(o
    0020 - 44 13 9a a3 74 84 8e 41-74 f5 aa da 66 30 27 e4   D...t..At...f0'.
    0030 - 54 86 37 b6 d1 98 94 ae-c4 fb 6c 46 a1 39 fb f9   T.7.......lF.9..
Policy OID: unspecified
Nonce: 0xBC6B597D588FA6DC
Certificate required: yes
Extensions:
```

4. Issue the time stamp request.

```
C:\Users\Administrator\Downloads>curl -H "Content-Type: application/timestamp-query" -H "Accept:
application/timestamp-reply" --data-binary "@hello-world.tsq" http://10.194.148.51/tsa/nshield-integration
--output hello-world.tsr
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3279  100  3177  100   102  73797   2369 --:--:-- --:--:-- --:--:-- 78071
```

5. Parse the timestamp response to validate the format.

```
C:\Users\Administrator\Downloads>openssl ts -reply -in hello-world.tsr -text
Using configuration from C:\Program Files\nCipher\nfast\\openssl\openssl.cnf
Status info:
Status: Granted.
Status description: Operation Okay
Failure info: unspecified
```

```
TST info:
Version: 1
Policy OID: Time Stamping
Hash Algorithm: sha512
Message data:
    0000 - 87 10 33 9d cb 68 14 d0-d9 d2 29 0e f4 22 28 5c   ..3..h....).."(\
    0010 - 93 22 b7 16 39 51 f9 a0-ca 8f 88 3d 33 05 28 6f   ."..9Q.....=3.(o
    0020 - 44 13 9a a3 74 84 8e 41-74 f5 aa da 66 30 27 e4   D...t..At...f0'.
    0030 - 54 86 37 b6 d1 98 94 ae-c4 fb 6c 46 a1 39 fb f9   T.7.......lF.9..
Serial number: 0xFC247B10D7AD0453
Time stamp: May 28 21:18:22 2025 GMT
Accuracy: unspecified
Ordering: no
Nonce: 0xBC6B597D588FA6DC
TSA: DirName:/CN=TSA
Extensions:
```

6. Verify the timestamp response against the original data. Notice the `chain.pem` file created in Export your root CA certificate.

```
C:\Users\Administrator\Downloads>openssl ts -verify -in hello-world.tsr -CAfile chain.pem -data hello-world.txt -ignore_critical -purpose any
Using configuration from C:\Program Files\nCipher\nfast\\openssl\openssl.cnf
Verification: OK
```

7. Verify the response against the timestamp request.

```
C:\Users\Administrator\Downloads>openssl ts -verify -in hello-world.tsr -CAfile chain.pem -queryfile hello-world.tsq -ignore_critical -purpose any
Using configuration from C:\Program Files\nCipher\nfast\\openssl\openssl.cnf
Verification: OK
```

Chapter 7. Troubleshoot

7.1. Troubleshoot the installation

1. SSH to the EDM virtual machine.

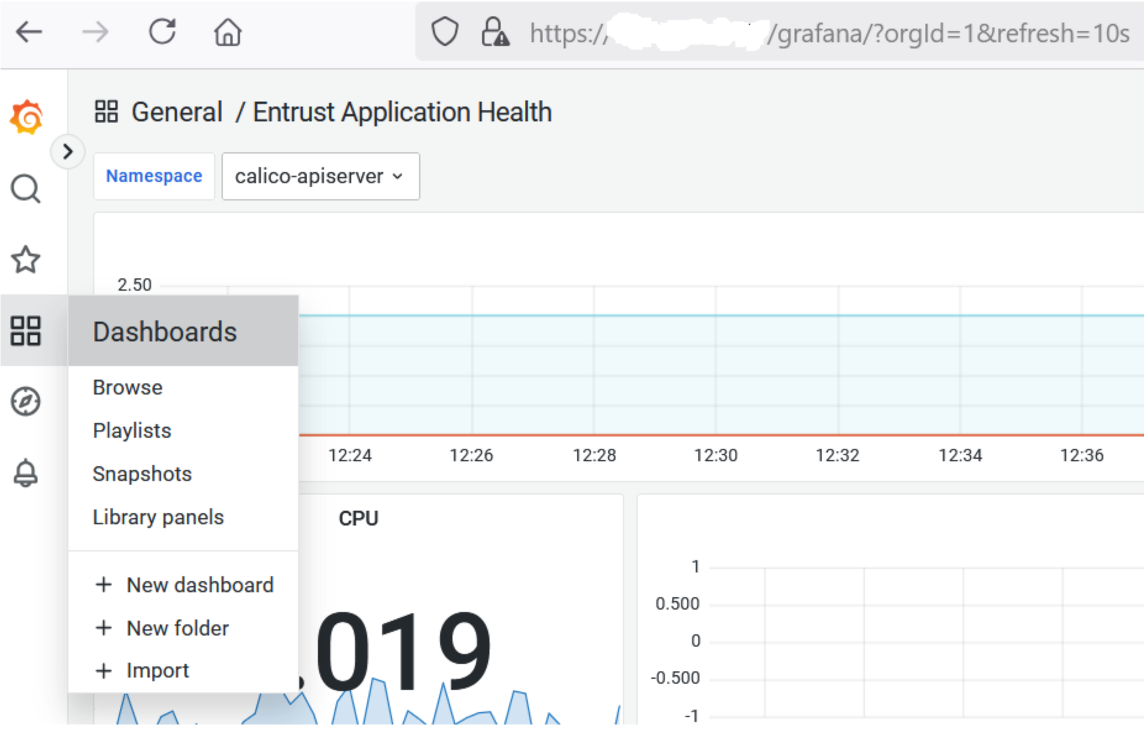
```
> ssh sysadmin@xxx.xxx.xxx.xxx
Authorized uses only. All activity may be monitored and reported.
sysadmin@10.194.148.51's password:
X11 forwarding request failed on channel 0
Last login: Thu May 22 23:03:33 2025 from 172.28.4.43
[sysadmin@timestamping-auth-edm ~]$
```

2. The logs are available at:

```
[sysadmin@timestamping-auth-edm entrust]$ ls -al /var/log/entrust/
total 4
drwxr-xr-x.  4 root root   28 May  9 13:39 .
drwxr-xr-x. 15 root root 4096 May 22 11:26 ..
drwxr-xr-x.  2 root root  245 May  7 17:32 edm
drwxrwxrwx.  2 root root   24 May 14 13:58 tsa
```

7.2. Troubleshoot the configuration

1. On a web browser navigate to `https://<machine>/grafana`, where <machine> is the EDM hostname or IP address. Log in as a user with Timestamping Authority management permissions, for example **admin**.
2. Select the **Dashboard**.

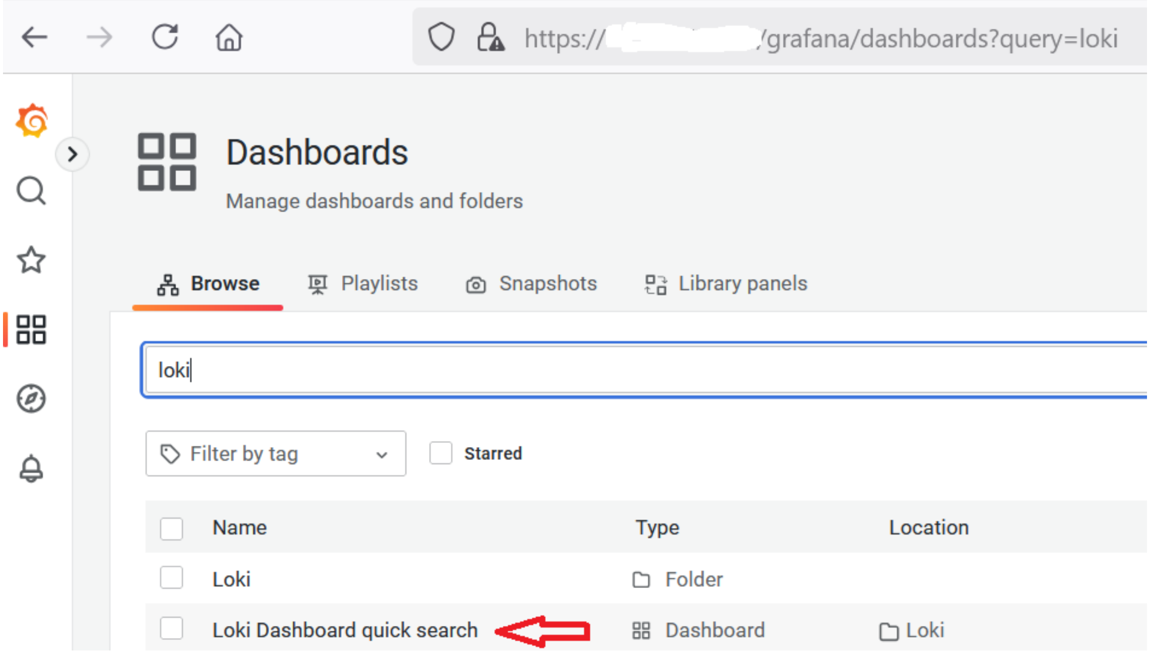


https://[redacted]/grafana/?orgId=1&refresh=10s

General / Entrust Application Health

Namespace: calico-apiserver

3. Type **loki** in the search bar and select **Loki Dashboard quick search**.



https://[redacted]/grafana/dashboards?query=loki

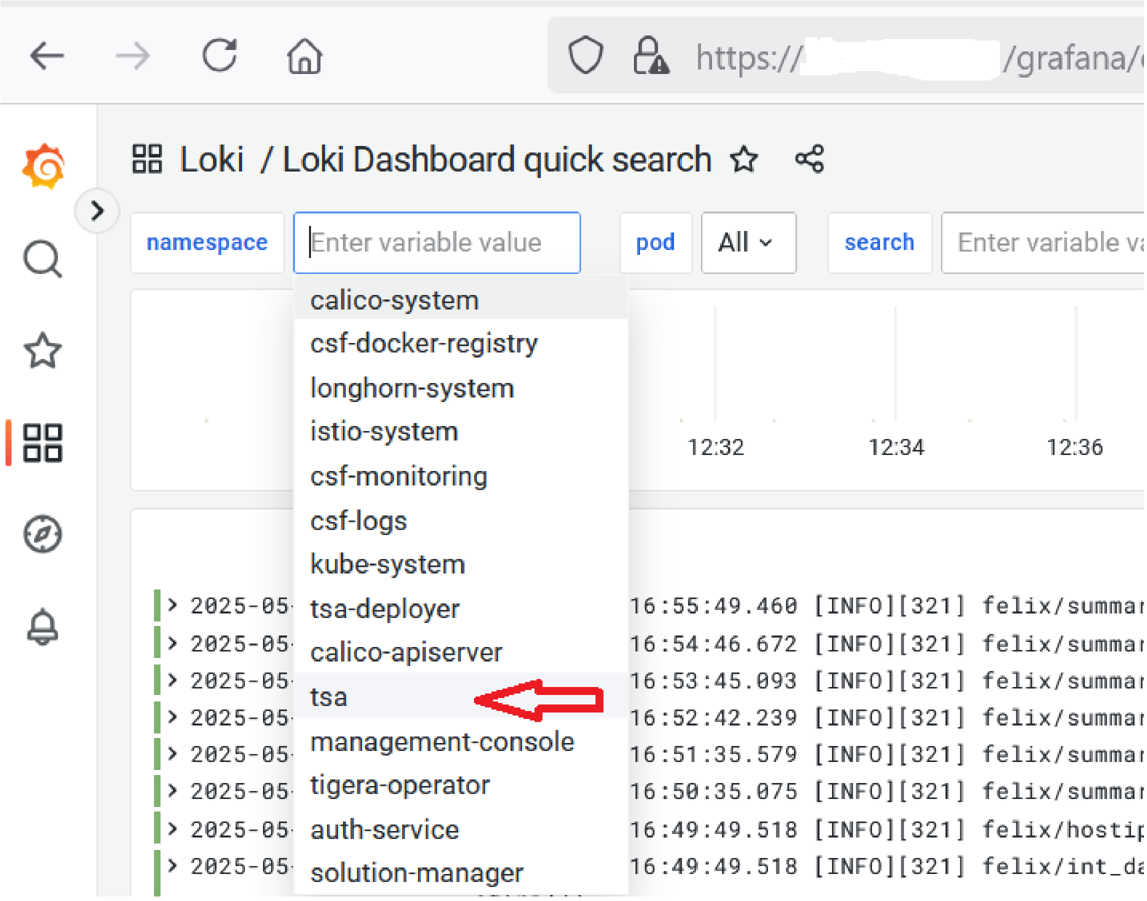
Dashboards

Manage dashboards and folders

loki

	Name	Type	Location
□	Loki	Folder	
□	Loki Dashboard quick search	Dashboard	Loki

4. In the **namespace** select **tsa**.



```
> 2025-05  16:55:49.460 [INFO][321] felix/summar
> 2025-05  16:54:46.672 [INFO][321] felix/summar
> 2025-05  16:53:45.093 [INFO][321] felix/summar
> 2025-05  16:52:42.239 [INFO][321] felix/summar
> 2025-05  16:51:35.579 [INFO][321] felix/summar
> 2025-05  16:50:35.075 [INFO][321] felix/summar
> 2025-05  16:49:49.518 [INFO][321] felix/hostip
> 2025-05  16:49:49.518 [INFO][321] felix/int_da
```

5. The log is now visible in the **Logs Panel**. Use the scroll bars and the **Refresh dashboard** icon on the top right to navigate the log.

Chapter 8. Additional resources and related products

8.1. nShield Connect

8.2. nShield as a Service

8.3. Entrust products

8.4. nShield product documentation