

# HNAS

---

## Generation of custom certificates with an HNAS system

While HNAS provides utilities to create both self-signed and certificates signed by a trusted Certificate Authority (CA), there are times when these are not sufficient, and they must be customized using OpenSSL tools. This guide describes this process.

**MK-92HNAS081-01**

**June 2023**

© 2021 Hitachi Vantara LLC. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara Corporation (collectively, "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara Corporation at [https://support.HitachiVantara.com/en\\_us/contact-us.html](https://support.HitachiVantara.com/en_us/contact-us.html).

**Notice:** Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara Corporation.

By using this software, you agree that you are responsible for:

- 1) Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals to access relevant data; and
- 2) Verifying that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

**Notice on Export Controls.** The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

**EXPORT CONTROLS** - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API, or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, Flash Copy, IBM, Lotus, MVS, OS/390, PowerPC, RS6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z/VM, BCPI™ and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or web site are properties of their respective owners.

# Table of Contents

<b>Table of Contents</b> .....	<b>3</b>
<b>Preface</b> .....	<b>4</b>
About this document.....	4
Document conventions.....	4
Intended audience .....	4
Accessing product downloads.....	4
Comments .....	4
Getting Help.....	5
<b>Chapter 1: When are advanced options required?</b> .....	<b>6</b>
HNAS Overview .....	6
HNAS Certificate Tools.....	6
SMU Certificate Tools.....	6
Limitations of the Standard Process.....	7
<b>Chapter 2: Obtaining and importing certificates with advanced options</b> .....	<b>8</b>
SMU certificate .....	8
HNAS Cluster Certificate.....	9
<b>Chapter 3: Example</b> .....	<b>12</b>
Creating SMU Certificate .....	12
Creating HNAS Cluster Certificate .....	17

# Preface

## About this document

This document describes the process of generating certificates using advanced options using OpenSSL tools for an HNAS system.

## Document conventions

This document uses the following typographic convention:

Convention	Description
<b>Bold</b>	<ul style="list-style-type: none"><li>Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: <b>Click OK</b>.</li><li>Indicates emphasized words in list items.</li></ul>
<i>Italic</i>	Indicates a document title or emphasized words in text.
Monospace	Indicates text that is displayed on screen or entered by the user. Example: <code>pairdisplay -g oradb</code>

## Intended audience

This document is intended for customers and Hitachi Vantara partners who license and use HNAS Platform.

## Accessing product downloads

Product software, drivers, and firmware downloads are available on Hitachi Vantara Support Connect: <https://support.hitachivantara.com/>.

Log in and select Product Downloads to access the most current downloads, including updates that may have been made after the product's release.

## Comments

Please send us your comments on this document to [doc.comments@hitachivantara.com](mailto:doc.comments@hitachivantara.com). Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara.

# Getting Help

[Hitachi Vantara Support Connect](https://support.hitachivantara.com/en_us/contact-us.html) is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: [https://support.hitachivantara.com/en\\_us/contact-us.html](https://support.hitachivantara.com/en_us/contact-us.html).

[Hitachi Vantara Community](https://community.hitachivantara.com) is a global online community for customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to [community.hitachivantara.com](https://community.hitachivantara.com), register, and complete your profile.

# Chapter 1: When are advanced options required?

## HNAS Overview

HNAS provides the following tools to manage certificates:

### HNAS Certificate Tools

- **tls-certificate-create-custom.** Generates a self-signed certificate. Note however that despite the name, it only supports a limited set of parameters, specifically:
  - -o <Organization>
  - -u <Organizational Unit>
  - -l <Location> -
  - s <State>
  - -c <Country>
- **tls-certificate-show.** Show contents of current certificate installed.
- **tls-certificate-generate-csr.** Generates a certificate signing request (CSR) to be submitted to a CA
- **tls-certificate-import-trust-chain.** Imports the CA trust chain.
- **tls-certificate-import-signed.** Imports the CA signed certificate. Note that as of 13.9.6619.00 it also has the ability to include a private key, which can be useful for server migrations.

The process of obtaining and importing a CA-signed certificate is described in the NAS Administration Guides > Server and Cluster Administration > Obtaining and importing a CA-signed certificate. The link for the latest version as of this writing is in : [https://knowledge.hitachivantara.com/Documents/Storage/NAS\\_Platform/14.6/NAS\\_Administration\\_Guides/Server\\_and\\_Cluster\\_Administration\\_Guide/56\\_Obtaining\\_and\\_importing\\_a\\_CA-signed\\_certificate.](https://knowledge.hitachivantara.com/Documents/Storage/NAS_Platform/14.6/NAS_Administration_Guides/Server_and_Cluster_Administration_Guide/56_Obtaining_and_importing_a_CA-signed_certificate.)

### SMU Certificate Tools

- **cert-showall.sh.** Shows the currently installed certificate.
- **cert-gencustom.ssh.** Creates a private key and self-signed certificate.
- **cert-gencsr.sh.** Generates CSR

- **cert-showcontent.sh.** Shows the content of the certificate file that you're about to import.
- **cert-importturstchain.sh.** Imports the CA certificate trust chain.
- **cert-import.sh.** Imports the CA-issued certificate.
- **cert-importprivatekey.sh.** Imports the private key.

The procedure to generate and install an SMU certificate is described in the NAS Administration Guides > Server and Cluster Administration > Generating a certificate signing request (CSR). For the latest version as of this writing go to:

[https://knowledge.hitachivantara.com/Documents/Storage/NAS\\_Platform/14.6/NAS\\_Administration\\_Guides/Server\\_and\\_Cluster\\_Administration\\_Guide/52\\_Generating\\_a\\_certificate\\_signing\\_request\\_\(CSR\)](https://knowledge.hitachivantara.com/Documents/Storage/NAS_Platform/14.6/NAS_Administration_Guides/Server_and_Cluster_Administration_Guide/52_Generating_a_certificate_signing_request_(CSR))

## Limitations of the Standard Process

When generating the CSR, the above methods allow you to customize the “Subject” part of the certificate which include: Common Name, Organization, Organizational Unit, Country, State or Provide, City. However, they do not allow you to customize Subject Alternative Names. Instead, these will be created by default:

SMU:

- Wildcard for the DNS domain, e.g. \*.<smu domainname>
- SMU hostname, without the domain, e.g. <smu hostname>
- SMU IP address

HNAS:

- Wildcard for the DNS domain, e.g. \*.<hnas domainname>
- Hostname of the node where the CSR was created
- <Admin EVS IP address>
- Each of the node IP addresses

There are cases where further customizing is required. One common use case is where additional options in Subject Alternative Name (SAN) must be customized. The commands included in HNAS OS and SMU described in the HNAS Administration Guides are not sufficient. Instead, we have to resort to the use of the OpenSSL tools to generate the CSR

and key. For convenience and consistency, we recommend using the ones already installed in the SMU.

## Chapter 2: Obtaining and importing certificates with advanced options

### SMU certificate

1. Using SSH, connect to the external SMU as [manager](#).
2. Navigate to the Linux bash shell by pressing **q** in the SMU managed servers menu
3. Once on Linux using a text editor, create a custom CSR configuration file **smu.cnf**. Alternatively, you can create the file **smu.cnf** in your local system, then upload it to the SMU `/home/manager` directory using the following structure

```
[ req ]
default_bits=2048
prompt=no
default_md=sha256
distinguished_name=dn
req_extensions=req_ext
[ dn ]
C=<country name>
ST=<state or region>
L=<town or city name>
O=<organization>
OU=<organizational unit>
CN=<SMU FQDN>
emailAddress=<admin email address>

[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1=<SMU FQDN>
DNS.2=<SMU IP address>
IP.2=<SMU-IP-Address>
```

You can add additional DNS alternative names, wildcards or IP addresses as needed with entries DNS.3, DNS.4, etc

4. Create the private key and CSR.

```
openssl req -new -sha256 -newkey rsa:2048 -nodes \  
-out smu.csr -keyout smu.key \  
-config smu.cnf
```

5. This will create two files: **smu.csr** and **smu.key**. You can confirm the contents of the CSR by running the command:

```
openssl req -in smu.csr -noout -text
```

6. Using an SCP client, connect to the external SMU as *manager*.
7. Download the file **smu.csr** to the local client.
8. Use the contents of **smu.csr** to request a certificate from the CA.
9. Once the certificate is ready to download, ensure that it is in PEM format. Save it with a file name **smu.cer**.
10. Download the CA trust chain certificate as well, saving it as **smu-chain.cer**.
11. From the SCP client transfer both the SMU and trust chain certificates to the 'manager' home directory in the external SMU.
12. From an SSH client, log in to the SMU as *manager*, and exit to the Bash Linux shell. This will place you in the */home/manager* directory
13. Concatenate the signed certificate, trust chain certificate and private key into a single combined PEM file:

```
cat smu.cer smu-chain.cer smu.key > smu_combined.pem
```

14. Become root by using the **su** command.
15. Import the combined PEM key by using the command:

```
cert-importprivatekey.sh -p smu_combined.pem
```

16. Restart the web server when prompted.
17. Once confirmed that the certificates are properly installed, you may delete all the files generated above: **smu.key**, **smu.csr**, **smu.cer**, **smu-chain.cer** and **smu-combined** from the */home/manager* directory

## HNAS Cluster Certificate

The HNAS cluster certificate is important for SOAP and REST API's to work correctly. Note that the certificate is created for the entire cluster. Therefore, each of the names and IP addresses for each node and the Admin EVS must be included in the SAN.

1. Using SSH, connect to the SMU as `manager`.
2. Navigate to the Linux bash shell by pressing `q` in the SMU managed servers menu
3. Once on Linux using a text editor, create a custom CSR configuration file **hnas.cnf**

```
[ req ]
default_bits=2048
prompt=no
default_md=sha256
distinguished_name=dn
req_extensions=req_ext

[ dn ]
C=<country name>
ST=<state or region>
L=<town or city name>
O=<organization>
OU=<organizational unit>
CN=<SMU FQDN>
emailAddress=<admin email address>

[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1=<Admin EVS FQDN>
DNS.2=<Node 1 FQDN>
DNS.3=<Node 2 FQDN>
DNS.4=<Admin EVS IP address>
DNS.5=<Node 1 IP address>
DNS.6=<Node 2 IP address>
IP.1=<Admin EVS IP address>
IP.2=<Node 1 IP address>
IP.3=<Node 2 IP address>
```

You can add additional DNS alternative names, wildcards or IP addresses as needed with entries DNS.7, DNS.8, etc.

If there are more than two nodes, the entries for the FQDN and IP addresses should be included as well in the SAN using the format above.

4. Generate a new server key and CSR by running the following command:

```
openssl req -new -sha256 -newkey rsa:2048 -nodes -out hnas.csr  
-keyout hnas.key -config hnas.cnf
```

This will create both the HNAS cluster server private key **hnas.key** and the CSR **hnas.csr**.

To verify CSR contents:

```
openssl req -text -noout -verify -in hnas.csr
```

5. Use the contents of the CSR file to request the certificate to the CA.

6. Once the certificate is ready, ensure that the signed certificate is in PEM format. Save the certificate file as **hnas.cer**.
7. Download the CA trust chain certificate as well, saving it as **hnas-chain.cer**.
8. Using an SCP client, transfer both the root CA certificate and the new server certificate to the /home/manager directory on the SMU.
9. Using an SSH client, log in to the SMU as 'manager', and exit to the Bash Linux shell. This will place you in the /home/manager directory
10. Combine the private, trust chain and public keys into a single PEM file **hnas-combined.pem** by running the following command:

```
cat hnas.cer hnas-chain.cer hnas.key > hnas-combined.pem
```

11. Go into the HNAS managed cluster by typing **siconsole** and then selecting it.
12. Import the PEM file by typing:

```
tls-certificate-import-signed --path hnas-combined.pem  
--with-private-key --confirm
```

13. Verify by typing:

```
tls-certificate-show
```

14. (optional) verify from a browser by going to the server's REST port 8443:

<https://<server-fqdn>:8443>

15. Once confirmed, you may delete the files hnas.key, hnas.csr, hnas.cer, hnas-chain.cer and hnas-combined from the /home/manager directory, or store them in a more secure location.

## Chapter 3: Example

Let's assume that our policy does not allow the use of wildcards (e.g. \*.example.com) or IP addresses. In addition, we would like to configure some additional FQDN's that describe the function of our HNAS solution (hnas) and its location subdomain (sc.example.com). Hence our subject alternative names should be:

	Hostname	Subject Alternative Names
HNAS cluster	zanzibar	zanzibar.example.com zanzibar-1.example.com zanzibar-2.example.com zanzibar.sc.example.com hnas.sjc.example.com 172.20.252.29 172.20.252.28 172.20.252.27
SMU	zanzibar-smu	zanzibar-smu.example.com zanzibar-smu.sjc.example.com hnas.sjc.example.com

When we use the standard `tls-certificate-create-custom` tools, despite the name we do not have a choice to customize the Subject Alternative Name. Instead, it will create them by default as follows:

```
X509v3 Subject Alternative Name:  
DNS:*.example.com, DNS:zanzibar-1, DNS:172.20.252.29,  
DNS:172.20.252.28, DNS:172.20.252.27, IP Address:172.20.252.29,  
IP Address:172.20.252.28, IP Address:172.20.252.27
```

Likewise, our SMU 'zanzibar-smu' will create a certificate with the following subject alternate names:

```
SubjectAlternativeName [  
  DNSName: *.sustlab.example.com  
  DNSName: zanzibar-smu  
  DNSName: 172.20.252.20  
  IPAddress: 172.20.252.20  
]
```

## Creating SMU Certificate

We create the file `smu.cnf`, and upload it to the SMU `/home/manager` directory.

```
[manager@zanzibar-smu ~]$ cat smu.cnf  
[ req ]  
default_bits=2048  
prompt=no  
default_md=sha256
```

```
distinguished_name=dn
req_extensions=req_ext
[ dn ]
C=US
ST=CA
L=San Jose
O=Example Corporation
OU=Engineering
CN=zanzibar-smu.example.com
emailAddress=user@example.com
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1=zanzibar-smu.example.com
DNS.2=zanzibar-smu.sc.example.com
DNS.3=hnas.sc.example.com
DNS.4=172.20.252.20
IP.1=172.20.252.20
```

We generate the CSR file **smu.csr** and private key file **smu.key**

```
[manager@zanzibar-smu ~]$ openssl req -new -sha256 -newkey rsa:2048
-nodes \
> -out smu.csr -keyout smu.key \
> -config smu.cnf
Generating a RSA private key
.+++++
.....+++++
writing new private key to 'smu.key'
-----
[manager@zanzibar-smu ~]$ ls
smu.cnf  smu.csr  smu.key
```

We verify contents of the CSR file

```
[manager@zanzibar-smu ~]$ openssl req -in smu.csr -noout -text
Certificate Request:
  Data:
    Version: 1 (0x0)
    Subject: C = US, ST = CA, L = San Jose, O = Example
Corporation, OU = Engineering, CN = zanzibar-smu.example.com,
emailAddress = user@example.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public-Key: (2048 bit)
      Modulus:
        00:b7:1f:5f:71:d7:41:fd:0e:58:2b:e8:6d:ec:c2:
        e1:9d:99:46:6b:a5:fd:ee:de:e2:a9:51:0b:db:e4:
        b2:c7:d4:9a:ab:46:51:bb:a1:f1:3d:de:de:a5:e1:
        36:e3:17:65:09:1c:64:c2:81:e0:01:1e:9f:41:69:
        26:16:ab:b1:0e:d4:1f:a1:b9:5f:32:5b:08:a2:71:
        64:42:4b:20:eb:cc:3f:5e:2f:15:2d:c1:9a:ac:5b:
        d1:8e:cd:bf:a2:20:b4:1b:06:80:f6:3c:fa:f2:52:
        c6:84:b6:97:52:c8:0a:2a:93:38:84:c4:bd:47:9e:
```

```

79:89:f9:8b:b5:bb:f8:45:dd:9e:0c:eb:87:33:63:
56:eb:8c:2b:25:b5:56:ea:8f:1e:56:d9:4b:7f:69:
1f:d4:61:e5:87:de:f2:45:dc:00:92:a7:c2:1c:fe:
a2:47:34:9a:cf:00:0f:09:ad:1e:88:72:28:89:a5:
ad:70:f3:d4:c0:40:83:58:55:3a:ef:ce:b5:df:34:
4b:4b:b3:6f:7f:ea:32:4d:32:18:1f:3a:24:92:50:
61:0b:63:4c:2a:93:61:fb:f3:f4:ca:d6:32:f0:63:
e7:3e:27:3a:90:fc:dc:1a:fa:e8:34:fb:1b:ce:da:
41:1a:42:b3:6b:28:0d:76:19:85:30:30:72:9f:d1:
6e:17
Exponent: 65537 (0x10001)
Attributes:
Requested Extensions:
  X509v3 Subject Alternative Name:
    DNS:zanzibar-smu.example.com, DNS:zanzibar-
smu.sjc.example.com, DNS:hnas.sjc.example.com, DNS:172.20.252.20, IP
Address:172.20.252.20
Signature Algorithm: sha256WithRSAEncryption
38:be:d2:52:87:59:7f:0c:77:e4:be:38:67:15:19:d0:37:45:
d7:c3:66:e7:f4:76:eb:ff:0f:33:88:55:84:a4:8d:c2:42:17:
8e:a1:5b:24:7e:03:da:66:35:c6:bb:39:cf:14:50:99:fb:0f:
da:dd:e9:40:e8:49:15:15:44:48:26:8c:b0:49:0a:27:bd:c3:
c5:f9:54:c5:87:d7:f3:d2:f0:16:0c:68:1a:64:5b:c3:91:cb:
e4:cb:ce:2a:77:1d:84:8e:26:b0:b7:07:c8:9d:08:a4:7e:ba:
42:c4:77:bd:f8:33:4d:6b:82:0e:6a:07:a5:95:a1:be:33:85:
8f:9f:fa:6c:76:f8:01:50:cf:e6:90:77:08:e7:3f:8b:de:15:
5b:b1:4a:7f:3b:78:90:28:68:35:bc:3f:00:0b:2b:b3:13:e6:
73:d1:39:9c:de:eb:b2:f4:9b:a5:b5:00:ef:2b:f3:70:98:b6:
4d:e5:ae:f0:d7:70:21:9a:12:7a:56:10:26:3a:d1:9b:28:b6:
ad:f3:96:e2:7a:13:61:cc:99:c4:36:c1:6f:d0:0d:91:68:18:
e9:f0:8d:3a:19:e9:5e:44:be:80:88:96:04:00:4c:23:5d:b7:
8f:90:b8:29:81:64:b9:91:69:b4:3b:b6:6a:6f:42:5e:15:37:
7c:14:c9:34

```

We download the CSR file `smu.csr`. and use it to request a certificate.

Once generated, we rename the SMU certificate as **`smu.cer`** and the trust chain certificate as **`smu-chain.cer`**.

Using an SCP client logged in as `manager`, we upload these certificates back to the SMU's `/home/manager` directory.

```
[manager@zanzibar-smu ~]$ ls
smu.cer  smu-chain.cer  smu.cnf  smu.csr  smu.key
```

We concatenate the certificate, trust chain certificate, and private key certificate into a combined file **`smu_all_with_key.pem`**:

```
[manager@zanzibar-smu ~]$ cat smu.cer smu-chain.cer smu.key >
smu_all_with_key.pem
```

We become root by issuing the `su` command.

```
[manager@zanzibar-smu ~]$ su
```

Password:

```
[root@zanzibar-smu manager]# ls
smu_all_with_key.pem  smu.cer  smu-chain.cer  smu.cnf  smu.csr
smu.key
```

**Note:** we are still in the /home/manager directory because we typed 'su' instead of 'su -'.

We import the combined certificate:

```
[root@zanzibar-smu manager]# cert-importtrustchain.sh -p smu-
chain.cer -a MyTrustedCA
Backing up current keystore to /etc/opt/mercury-
papi/ssl/keystore.backup
```

The new certificate will now be imported into the keystore as alias=MyTrustedCA.  
Certificate was added to keystore

Warning:

The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore /etc/opt/mercury-papi/ssl/nas.keystore -destkeystore /etc/opt/mercury-papi/ssl/nas.keystore -deststoretype pkcs12".

Next we import the signed certificate:

```
[root@zanzibar-smu manager]# cert-importprivatekey.sh -p
smu_all_with_key.pem
```

Replaces the private key and certificate in this system's keystore. The private key should be in pkcs8 format (unencrypted), and the cert should be x509, both PEM encoded in one file. You may restore the default system certificate by running cert-gendefault.sh

Are you sure you want to continue [y/n]? **y**

```
Checking for certificate file at smu_all_with_key.pem
Backing up current keystore to /etc/opt/mercury-
papi/ssl/keystore.backup
```

The new certificate will now be imported into the keystore as alias=nas.  
Importing keystore /tmp/cert.pl2 to /etc/opt/mercury-papi/ssl/nas.keystore...  
Warning: Overwriting existing alias 1 in destination keystore [Storing /etc/opt/mercury-papi/ssl/nas.keystore]

Warning:

The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore /etc/opt/mercury-papi/ssl/nas.keystore -destkeystore /etc/opt/mercury-papi/ssl/nas.keystore -deststoretype pkcs12".

In order for the certificate to take effect, the web server must be restarted.

Do you want to restart the web server now [y/n]: **y**

Using CATALINA\_BASE: /opt/smu/tomcat

Using CATALINA\_HOME: /opt/smu/tomcat

Using CATALINA\_TMPDIR: /var/opt/smu/tmp

Using JRE\_HOME: /opt/smu/javaTools/java

Using CLASSPATH:

/opt/smu/tomcat/bin/bootstrap.jar:/opt/smu/tomcat/bin/tomcat-juli.jar

Using CATALINA\_OPTS:

umask not set in axalon.properties so setting the default umask of 027

UMASK=0027

Using CATALINA\_BASE: /opt/smu/tomcat

Using CATALINA\_HOME: /opt/smu/tomcat

Using CATALINA\_TMPDIR: /var/opt/smu/tmp

Using JRE\_HOME: /opt/smu/javaTools/java

Using CLASSPATH:

/opt/smu/tomcat/bin/bootstrap.jar:/opt/smu/tomcat/bin/tomcat-juli.jar

Using CATALINA\_OPTS:

Tomcat started.

We can confirm that the certificate was installed by using the command cert-showall.sh.

```
[root@zanzibar-smu manager]# cert-showall.sh
```

```
Keystore type: jks
```

```
Keystore provider: SUN
```

```
Your keystore contains 1 entry
```

```
Alias name: nas
```

```
Creation date: Jun 26, 2023
```

```
Entry type: PrivateKeyEntry
```

```
Certificate chain length: 2
```

```
Certificate[1]:
```

```
Owner: EMAILADDRESS=user@example.com, CN=zanzibar-smu.example.com,
```

```
OU=Engineering, O=Example Corporation, L=San Jose, ST=CA, C=US
```

```
Issuer: CN=MyRootCA, DC=example, DC=com
```

```
Serial number: 1a0000070273f09601d22d48c1000200000702
```

```
Valid from: Mon Jun 26 16:41:55 PDT 2023 until: Tue May 13 05:47:59
```

```
PDT 2025
```

```
Certificate fingerprints:
```

```
SHA1:
```

```
BF:BD:63:13:12:A2:08:51:C6:81:D5:2B:C5:31:6F:76:02:31:94:3B
```

```
SHA256:
8F:52:4A:8F:C7:A8:29:32:C3:71:D2:45:0D:A5:49:60:13:5E:E8:B6:F5:92:CC
:12:AF:95:B1:E4:CE:84:7D:31
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
```

Extensions:

...

At this point, we can delete the files in the /home/manager directory, since we no longer need them.

```
[manager@zanzibar-smu ~]$ ls
smu_all_with_key.pem  smu.cer  smu-chain.cer  smu.cnf  smu.csr
smu.key
[manager@zanzibar-smu ~]$ rm smu*
```

## Creating HNAS Cluster Certificate

Our HNAS server has the following IP's and nodes

```
zanzibar-1:$ evs list -e 0
```

Node	EVS ID	Type	Label	Enabled	Status	IP Address	Port
1		Cluster	zanzibar-1	Yes	Online	172.20.252.28	eth1
1	0	Admin	zanzibar-1	Yes	Online	172.20.252.29	eth1
2		Cluster	zanzibar-2	Yes	Online	172.20.252.27	eth1

We begin by creating a configuration file, hnas.cnf placed in the /home/manager directory of the SMU. The Subject Alternative Name includes all the node names, IP addresses, and cluster names.

```
[manager@zanzibar-smu ~]$ cat hnas.cnf
[ req ]
default_bits=2048
prompt=no
default_md=sha256
distinguished_name=dn
req_extensions=req_ext
[ dn ]
C=US
ST=CA
L=San Jose
O=Hitachi Vantara
OU=File Engineering
CN=zanzibar.example.com
emailAddress=user@example.com
[ req_ext ]
subjectAltName = @alt_names
```

```
[alt_names]
DNS.1=zanzibar.example.com
DNS.2=zanzibar-1.example.com
DNS.3=zanzibar-2.example.com
DNS.4=zanzibar.sjc.example.com
DNS.5=hnas.sjc.example.com
DNS.6=172.20.252.29
DNS.7=172.20.252.28
DNS.8=172.20.252.27
IP.1=172.20.252.29
IP.2=172.20.252.28
IP.3=172.20.252.27
```

We generate the CSR

```
[manager@zanzibar-smu ~]$ openssl req -new -sha256 -newkey rsa:2048
-nodes -out hnas.csr -keyout hnas.key -config hnas.cnf
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'hnas.key'
-----
```

We confirm that a CSR and a private key file have been created

```
[manager@zanzibar-smu ~]$ ls
hnas.cnf hnas.csr hnas.key
```

Using the contents of the hnas.csr file, we request a server key from our CA.

Once we receive the signed certificate, we save it as hnas.cer. We also save the trust chain file as hnas-chain.cer.

We upload these files to the SMU's /home/manager directory using an SCP client.

```
[manager@zanzibar-smu ~]$ ls
hnas.cer hnas-chain.cer hnas.cnf hnas.csr hnas.key
```

We concatenate the signed certificate, trust chain certificate, and private key into a combined PEM file.

```
[manager@zanzibar-smu ~]$ cat hnas.cer hnas-chain.cer hnas.key >
hnas-combined.pem
```

We enter the HNAS OS shell.

```
[manager@zanzibar-smu ~]$ siconsole
```

```
Available servers:
=====
```

```
1) 172.20.252.29 zanzibar-avn.corp.bigo.local
```

Please choose a server, or type 'q' to exit to a bash shell.  
Server: **1**

Hitachi NAS OS Console  
MAC ID : 7A-B6-FA-2D-53-5C  
Cluster MAC ID : 7A-B6-FA-2D-53-5C

zanzibar-1:\$

We import the combined file:

```
zanzibar-1:$ tls-certificate-import-signed --path hnas-combined.pem  
--with-private-key --confirm
```

```
Wrote 5.355 KB (5484 B) in 51 ms at 105 KB/s (107529 B/s)  
Read 5484 bytes from the certificate file.  
Importing X509 certificates bundle with a key...  
certificate-management: Certificate in registry differs from  
certificate in keystore. Will update certificate in keystore.  
certificate-management: A new keystore has been successfully  
created.  
certificate-management: Certificate has been propagated from the  
registry to the keystore.  
certificate-management: Certificate in registry differs from  
certificate in keystore. Will update certificate in keystore.  
certificate-management: A new keystore has been successfully  
created.  
certificate-management: The trust chain CA0 has been propagated from  
the registry to the keystore.  
certificate-management: Certificate has been propagated from the  
registry to the keystore.  
Extracted and imported trust chain with alias: 'CA1'.  
Extracted and imported the X509 certificate(s) with the private key  
successfully.  
The provided certificate has been successfully imported.  
Restarting HTTPS server...  
HTTPS server restarted.  
Restarting RestAPI server...  
RestAPI server restarted.  
[tls-certificate-import-signed took 8 s.]  
zanzibar-1:$
```

We confirm that the certificate was installed:

```
zanzibar-1:$ tls-certificate-show  
Is certificate default: no
```

Certificate:

```
Data:  
Version: 3 (0x2)  
Serial Number:  
1a:00:00:07:03:ac:6c:ae:aa:80:7a:57:56:00:02:00:00:07:03  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: DC=com, DC=example, CN=MyRootCA  
Validity  
Not Before: Jun 27 01:01:15 2023 GMT
```

```
Not After : May 13 12:47:59 2025 GMT
Subject: C=US, ST=CA, L=San Jose, O=Example, OU=File
Engineering, CN=zanzibar.example.com/emailAddress=user@example.com
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public-Key: (2048 bit)
  Modulus:
...
  X509v3 extensions:
    X509v3 Subject Alternative Name:
      DNS:zanzibar.example.com, DNS:zanzibar-
1.example.com, DNS:zanzibar-2.example.com,
DNS:zanzibar.sjc.example.com, DNS:hnas.sjc.example.com,
DNS:172.20.252.29, DNS:172.20.252.28, DNS:172.20.252.27, IP
Address:172.20.252.29, IP Address:172.20.252.28, IP
Address:172.20.252.27
...
zanzibar-1:$
```

We also go to <https://172.20.252.29:8443/> to verify the contents of the certificate.

## Hitachi Vantara

Corporate Headquarters 2535 Augustine Drive  
Santa Clara, CA 95054 USA [www.HitachiVantara.com](http://www.HitachiVantara.com) [community.HitachiVantara.com](http://community.HitachiVantara.com)

### Regional Contact Information

Americas: +1 866 374 5822 or [info@hitachivantara.com](mailto:info@hitachivantara.com)

Europe, Middle East and Africa: +44 (0) 1753 618000 or [info.emea@hitachivantara.com](mailto:info.emea@hitachivantara.com)

Asia Pacific: +852 3189 7900 or [info.marketing.apac@hitachivantara.com](mailto:info.marketing.apac@hitachivantara.com)

