



ENTRUST

CA Gateway 3.7

Docker Deployment Guide

Document issue: 1.1
Issue date: May 15, 2026

© 2026, Entrust. All rights reserved

Entrust and the hexagon design are trademarks, registered trademarks and/or service marks of Entrust Corporation in Canada and the United States and in other countries. All Entrust product names and logos are trademarks, registered trademarks and/or service marks of Entrust Corporation. All other company and product names and logos are trademarks, registered trademarks and/or service marks of their respective owners in certain countries.

This information is subject to change as Entrust reserves the right to, without notice, make changes to its products as progress in engineering or manufacturing methods or circumstances may warrant. The material provided in this document is for information purposes only. It is not intended to be advice. You should not act or abstain from acting based upon such information without first consulting a professional. ENTRUST DOES NOT WARRANT THE QUALITY, ACCURACY OR COMPLETENESS OF THE INFORMATION CONTAINED IN THIS ARTICLE. SUCH INFORMATION IS PROVIDED "AS IS" WITHOUT ANY REPRESENTATIONS AND/OR WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, BY USAGE OF TRADE, OR OTHERWISE, AND ENTRUST SPECIFICALLY DISCLAIMS ANY AND ALL REPRESENTATIONS, AND/OR WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT, OR FITNESS FOR A SPECIFIC PURPOSE.

Contents

1	About this guide	7
	Revision information	7
	Related documentation	7
	Documentation feedback	7
2	Overview	8
	Client	8
	Integrator	9
	Tenant	9
	Managed CA	9
3	Release notes.....	10
	Release notes for 3.2.0	10
	Release notes for 3.2.1	12
	Release notes for 3.3.0	13
	Release notes for 3.4.0	15
	Release notes for 3.7.0	15
4	Requirements	19
	Docker requirements.....	19
	HSM requirements.....	19
5	Preparing the deployment.....	20
	Getting the CA Gateway license	20
	Downloading the installation files.....	20
	Verifying the downloaded files	21
	Loading the CA Gateway Image	21
	Tagging the CA Gateway Image	21
	Listing the images.....	21
	Generating the systemd service for Podman	22
	Creating the configuration and credentials folders.....	23
	Running the Profile Creation Utility.....	24
	Verifying the installation	30

Securing settings with jTinyUAL	31
Obtaining the server certificate	32
6 Integrating Certificate Authorities.....	36
Integrating a DigiCert CA	36
Integrating a GlobalSign CA	36
Integrating a Microsoft CA.....	36
Integrating an AWS CA.....	59
Integrating an EJBCA.....	64
Integrating an Entrust CA	69
Integrating a Sectigo CA	78
7 Configuring	81
cagw	83
logging	144
management	145
server.....	149
8 Starting up and deploying	153
Configuring the Docker installation	153
Configuring clock synchronization	153
Running the CA Gateway Docker container	154
Stopping the execution	155
9 Enabling CRL revocation check	157
10 Configuring clients	158
11 Issuing public trust certificates	159
CA Authorization.....	159
Certificate Transparency.....	160
12 Administrating the deployment	161
Checking the CA Gateway health.....	161
Checking the health of a CA	161
Managing logs.....	161
Updating the configuration	162

13 API endpoints	164
CA properties endpoint	167
CA status endpoint	168
Certificate actions endpoint	169
Certificate events endpoint	174
Disk space endpoint	175
Documentation endpoint	176
Domain actions endpoint.....	176
Domains endpoint.....	179
Enrollments endpoint	179
Health endpoint.....	188
Metrics endpoint.....	189
Ping endpoint	190
Swagger endpoint	190
Version endpoint	191
14 CA Capabilities reference	193
Digicert CA capabilities	193
EJBCA capabilities	195
Entrust CA capabilities	199
GlobalSign capabilities	202
Microsoft CA capabilities	205
Sectigo CA capabilities	208
15 Troubleshooting.....	212
EJBCA authentication failures	212
EJBCA SSL/TLS Connection Failures	212
EJBCA certificate issuance failures.....	212
16 Integration report.....	213
Supported Certificate Authorities.....	213
Open-source plugins compatible with CA Gateway	214
Supported Platforms.....	214

Supported HSM.....	214
Supported TLS configurations.....	215

1 About this guide

This guide describes how to install and configure CA Gateway 3.5 in Docker.

- [Revision information](#)
- [Related documentation](#)
- [Documentation feedback](#)

Revision information

See the following table for the issued versions of this document.

Issue	Date	Section	Description
1.1	May 2026	Integration report	Add the supported TLS configurations
1.0	May 2026	All sections	The first release of this document

Related documentation

See the following table for the documentation related to this guide.

Document	Contents
CA Gateway 3.x - Entrust nShield Integration Guide	Integration of an nShield Hardware Security Module (HSM) with Entrust CA Gateway
CA Gateway 3.x - Thales Luna Integration Guide	Integration of a Thales Luna Hardware Security Module (HSM) with Entrust CA Gateway

Documentation feedback

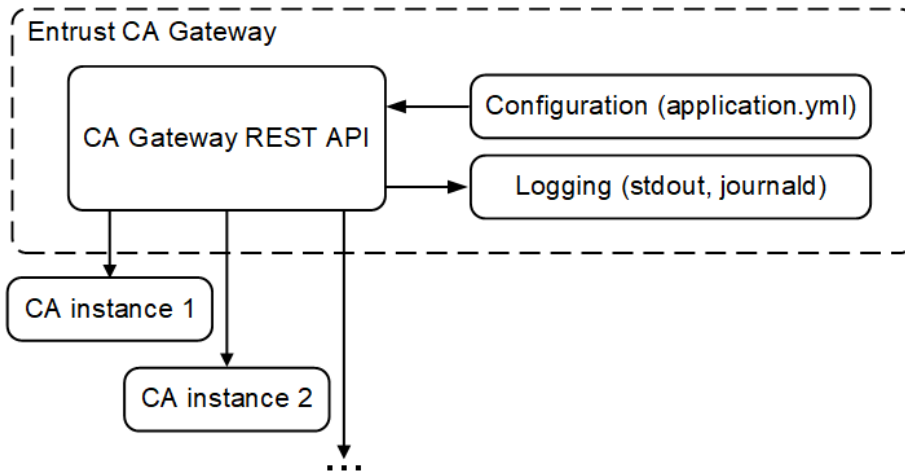
You can rate and provide feedback about product documentation by completing the online feedback form:

<https://go.entrust.com/documentation-feedback>

Any information you provide goes directly to the documentation team and is used to improve and correct the information in our guides.

2 Overview

CA Gateway is a lightweight, container-based module implementing a CA-agnostic Certificate Lifecycle and Policy Management API. Using CA Gateway, your applications can provide certificate issuance, renewal, and revocation actions across different Certification Authorities (CAs). CA Gateway provides policy retrieval capabilities so applications can customize API and user-facing dialogs to ensure that certificate actions conform to organizational policies.



See below for a description of each component.

- [Client](#)
- [Integrator](#)
- [Tenant](#)
- [Managed CA](#)

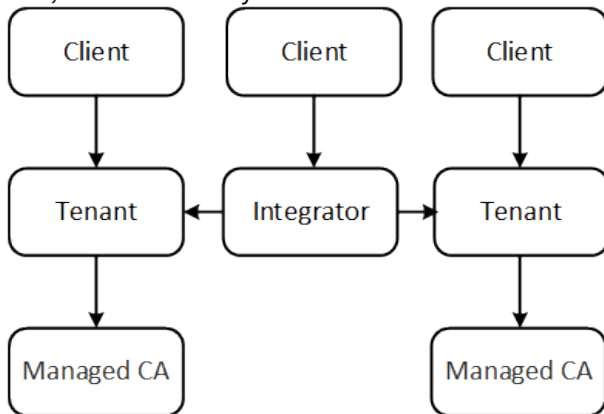
i CA-specific plugins communicate with the underlying CAs through mutually authenticated TLS.

Client

Each client is an authorized end entity of the CA Gateway API and is mapped either to a tenant or an integrator.

- Clients mapped to an integrator can access many Managed CAs.
- Clients mapped to a tenant can access only that tenant's Managed CA.

Thus, each CA Gateway client can access one or several CAs.



The CA Gateway API is regularly updated to add functionalities. Therefore, client applications:

- Should tolerate and ignore new fields.
- Should be recompiled against the new data model of each CA Gateway release.

Integrator

Each integrator is an access controller for one or more tenants.

Tenant

Each tenant is an access controller for a Managed CA. Thus, each tenant:

- Has only one integrator.
- Controls access to a different CA.

Managed CA

Each "managed CA" is a set of information that CA Gateway uses to connect to a CA. For example:

- Microsoft Active Directory Certificate Services.
- AWS Certificate Manager Private Certificate Authority.
- Entrust Certificate Authority.

3 Release notes

See below the release notes for CA Gateway 3.2.0 to 3.5

- [Release notes for 3.2.0](#)
- [Release notes for 3.2.1](#)
- [Release notes for 3.3.0](#)
- [Release notes for 3.4.0](#)
- [Release notes for 3.7.0](#)

Release notes for 3.2.0

See below the CA Gateway 3.2.0 release notes.

- [New features for 3.2.0](#)
- [Bugs fixed for 3.2.0](#)
- [Known issues in 3.2.0](#)

New features for 3.2.0

CA Gateway 3.2.0 adds the following features to CA Gateway.

- [Cache management \(ATEAM-17140\)](#)
- [Sectigo CA support \(ATEAM-18339\)](#)
- [Profile selection on request \(ATEAM-18358\)](#)
- [Multiple KeyUsage on request \(ATEAM-18374\)](#)

Cache management (ATEAM-17140)

This release adds support for enabling and disabling CA Gateway API caches.

Sectigo CA support (ATEAM-18339)

This release adds support for integrating Sectigo certificate authorities with CA Gateway.

Profile selection on request (ATEAM-18358)

This release adds support for selecting the user role of certificates issued by an Entrust Security Manager certificate authority.

Multiple KeyUsage on request (ATEAM-18374)

This release adds support for a plaintext comma-separated list of KeyUsage values on enrollment requests.

Bugs fixed for 3.2.0

CA Gateway 3.2.0 fixes the following CA Gateway bug.

- [Error when revoking certificates without Subject \(ATEAM-11625\)](#)
- [Missing HTTP Strict Transport Security header \(ATEAM-16176\)](#)
- [Missing Content Security Policy header \(ATEAM-16215\)](#)
- [Unspecified validation error \(ATEAM-17472\)](#)
- [Sectigo CA not supported \(ATEAM-18790\)](#)

Error when revoking certificates without Subject (ATEAM-11625)

CA Gateway returns an error when trying to revoke a certificate with an empty Subject field.

Missing HTTP Strict Transport Security header (ATEAM-16176)

CA Gateway does not add an HTTP Strict Transport Security (HSTS) header to responses.

Missing Content Security Policy header (ATEAM-16215)

CA Gateway does not add a Content Security Policy (CSP) header to responses.

Unspecified validation error (ATEAM-17472)

When entering a malformed value in the **Choose a key name** field, the Management Console displays a generic "Please match the requested format" breadcrumb message without indicating the source of the error.

Bug fixing: explicitly indicate that the CA key value must include only lowercase letters, numbers, and dashes ("-").

Sectigo CA not supported (ATEAM-18790)

Certificate Hub 4.2.0 does not support requesting certificates from a CA Gateway 3.2.0 instance integrated with a Sectigo CA.

 To fix this bug, you must install Certificate Hub 4.2.1 and CA Gateway 3.2.1.

Known issues in 3.2.0

CA Gateway 3.2.0 has the following known issues.

- [Mandatory parameters are mutually exclusive \(ATEAM-16246\)](#)
- [Slashes not supported in passwords \(ATEAM-18325\)](#)

Mandatory parameters are mutually exclusive (ATEAM-16246)

When configuring a CA Gateway client, the following mandatory parameters are mutually exclusive (that is, you must select one but not both).

- Tenant ID
- Integrator ID

However, the Management Console raises an error during validation if any of these values is unselected.

Detected in: CA Gateway 3.0.0 to 3.0.5.

Workaround:

1. Delete the client settings.
2. Recreate the client settings using either the **Tenant ID** or **Integrator ID** parameter.

Slashes not supported in passwords (ATEAM-18325)

CA Gateway deployments on appliances do not support slashes ("/") in passwords (such as keystore passwords, truststore passwords, Entrust Profile File passwords, etc.).

Detected in: CA Gateway installations on EDM (Entrust Deployment Manager) and PKI Hub.

Release notes for 3.2.1

See below the CA Gateway 3.2.1 release notes.

- [Fixed bugs for 3.2.1](#)
- [Known issues for 3.2.1](#)

Fixed bugs for 3.2.1

CA Gateway 3.2.1 fixes the following bug.

- [Missing HTTP Strict Transport Security header \(ATEAM-16176\)](#)
- [Missing Content Security Policy header \(ATEAM-16215\)](#)
- [Sectigo CA not supported \(ATEAM-18790\)](#)

Missing HTTP Strict Transport Security header (ATEAM-16176)

CA Gateway does not add an HTTP Strict Transport Security (HSTS) header to responses.

Missing Content Security Policy header (ATEAM-16215)

CA Gateway does not add a Content Security Policy (CSP) header to responses.

Sectigo CA not supported (ATEAM-18790)

Certificate Hub 4.2.0 does not support requesting certificates from a CA Gateway 3.2.0 instance integrated with a Sectigo CA.

 To fix this bug, you must install Certificate Hub 4.2.1 and CA Gateway 3.2.1.

Known issues for 3.2.1

CA Gateway 3.2.1 has the following known issues.

- [Mandatory parameters are mutually exclusive \(ATEAM-16246\)](#)
- [Slashes not supported in passwords \(ATEAM-18325\)](#)

Mandatory parameters are mutually exclusive (ATEAM-16246)

When configuring a CA Gateway client, the following mandatory parameters are mutually exclusive (that is, you must select one but not both).

- Tenant ID
- Integrator ID

However, the Management Console raises an error during validation if any of these values is unselected.

Detected in: CA Gateway 3.0.0 to 3.0.5.

Workaround:

1. Delete the client settings.

2. Recreate the client settings using either the **Tenant ID** or **Integrator ID** parameter.

Slashes not supported in passwords (ATEAM-18325)

CA Gateway deployments on appliances do not support slashes (/) in passwords (such as keystore passwords, truststore passwords, Entrust Profile File passwords, etc.).

Detected in: CA Gateway installations on EDM (Entrust Deployment Manager) and PKI Hub.

Release notes for 3.3.0

See below the CA Gateway 3.3.0 release notes.

- [Bugs fixed for 3.3.0](#)
- [Known issues for 3.3.0](#)

Bugs fixed for 3.3.0

CA Gateway 3.3.0 fixes the following bugs.

- [CA lookup fails when the DN includes tildes \(ATEAM-18768\)](#)
- [LDAP and LDAPS port incompatibility not documented \(ATEAM-18779\)](#)
- [Incorrect option name in the plugin selector \(ATEAM-18782\)](#)

CA lookup fails when the DN includes tildes (ATEAM-18768)

The lookup for a CA (Certificate Authority) fails when the DN (Distinguished Name) of the CA contains tildes, like in "MODERNIZAÇÃO".

LDAP and LDAPS port incompatibility not documented (ATEAM-18779)

The user guide does not properly outline that the **LDAP Port** and **LDAPS Port** settings are mutually exclusive.

Incorrect option name in the plugin selector (ATEAM-18782)

The **Connector Name** selector of the Management Console lists the Sectigo CA plugin as **com.Sectigo** instead of **com.SectigoCA**.

Workaround:

1. Export the CA Gateway configuration using the `clusterctl solution config export` command
2. Replace `com.Sectigo` with `com.SectigoCA` in the following files.
 - `config-schema.json`
 - `sectigo-schema.json`
 - `application.yml` (only if CA Gateway has been deployed with the Sectigo CA plugin).
3. Apply the new configuration with the `clusterctl solution config import` command.
4. Deploy or redeploy the solution with the `clusterctl solution deploy` command.

Known issues for 3.3.0

CA Gateway 3.3.0 for Docker has the following known issues.

- Mandatory parameters are mutually exclusive (ATEAM-16246)
- subject.certificates field omitted (ATEAM-16264)
- Slashes not supported in passwords (ATEAM-18325)
- Incorrect option name in the plugin selector (ATEAM-18782)
- Sectigo CA profile synchronization requires redeploying CA Gateway (ATEAM-18819)

Mandatory parameters are mutually exclusive (ATEAM-16246)

When configuring a CA Gateway client, the following mandatory parameters are mutually exclusive (that is, you must select one but not both).

- Tenant ID
- Integrator ID

However, the Management Console raises an error during validation if any of these values is unselected.

Detected in: CA Gateway 3.0.0 to 3.0.5.


Workaround:

1. Delete the client settings.
2. Recreate the client settings using either the **Tenant ID** or **Integrator ID** parameter.

subject.certificates field omitted (ATEAM-16264)

For performance reasons, the PKIaaS CA Plugin will not honor the `subject.certificates` field in the following endpoint.

```
api/v1/certificate-authorities/{caId}/subjects/dn
```

 Future releases may restore this functionality.

Detected in: CA Gateway 3.0.1 to 3.0.5.

Slashes not supported in passwords (ATEAM-18325)

CA Gateway deployments on appliances do not support slashes ('/') in passwords (such as keystore passwords, truststore passwords, Entrust Profile File passwords, etc.).

Detected in: CA Gateway installations on EDM (Entrust Deployment Manager) and PKI Hub.

Incorrect option name in the plugin selector (ATEAM-18782)

The **Connector Name** selector of the Management Console lists the Sectigo CA plugin as **com.Sectigo** instead of **com.SectigoCA**.

Workaround:

1. Export the CA Gateway configuration using the `clusterctl solution config export` command
2. Replace `com.Sectigo` with `com.SectigoCA` in the following files.
 - `config-schema.json`
 - `sectigo-schema.json`
 - `application.yml` (only if CA Gateway has been deployed with the Sectigo CA plugin).

3. Apply the new configuration with the `clusterctl solution config import` command.
4. Deploy or redeploy the solution with the `clusterctl solution deploy` command.

Sectigo CA profile synchronization requires redeploying CA Gateway (ATEAM-18819)

Changes to Sectigo CA custom fields are not automatically effective in the `requestedProperties` field.

Workaround: redeploy CA Gateway using the Management Console or the `clusterctl solution deploy` command.

Release notes for 3.4.0

See below the CA Gateway 3.4.0 release notes.

- [New features for 3.4.0](#)
- [Known issues for 3.4.0](#)

New features for 3.4.0

CA Gateway 3.4.0 adds the following new feature.

DigiCert CA Support (ATEAM-18568)

CA Gateway 3.4.0 introduces support for issuing DV (Domain Validation) certificates with DigiCert certificate authorities.

Known issues for 3.4.0

CA Gateway 3.4.0 has the following known issue.

- [Mandatory parameters are mutually exclusive \(ATEAM-16246\)](#)

Mandatory parameters are mutually exclusive (ATEAM-16246)

When configuring a CA Gateway client, the following mandatory parameters are mutually exclusive (that is, you must select one but not both).

- Tenant ID
- Integrator ID

However, the Management Console raises an error during validation if any of these values is unselected.

Detected in: CA Gateway 3.0.0 to 3.0.5.

Workaround:

1. Delete the client settings.
2. Recreate the client settings using either the **Tenant ID** or **Integrator ID** parameter.

Release notes for 3.7.0

See below the CA Gateway 3.7.0 release notes.

- [New features for 3.7.0](#)

- [Bugs fixed for 3.7.0](#)
- [Known issues for 3.7.0](#)


New features for 3.7.0

CA Gateway 3.7.0 adds the following new features.

- [Support for renewing Sectigo-issued certificates \(ATEAM-18908\)](#)
- [EJBCA support \(ATEAM-18931\)](#)
- [Support for post-quantum cryptography algorithms \(PKI-42161\)](#)

Support for renewing Sectigo-issued certificates (ATEAM-18908)

This release adds support for issuing certificates issued by a Sectigo CA.

 In the current CA Gateway release, GlobalSign integration is supported for testing purposes only.

EJBCA support (ATEAM-18931)

This release adds support for [Integrating an EJBCA CA](#).

Support for post-quantum cryptography algorithms (PKI-42161)

This release adds support for post-quantum cryptography (PQC) algorithms. Specifically, when [Integrating an Entrust CA](#), CA Gateway supports the following algorithms.

- [Module-Lattice-based Digital Signature Algorithm \(ML-DSA\)](#)
- [Module-Lattice-based Key Encapsulation Mechanism \(ML-KEM\)](#)

Bugs fixed for 3.7.0

CA Gateway 3.7.0 fixes the following bugs.

- [Unexpected warning logs \(ATEAM-18776\)](#)
- [URL not hardcoded on Sectigo configuration \(ATEAM-18856\)](#)
- [Error when renewing a Sectigo-issued SSL certificate \(ATEAM-18858\)](#)
- [Sectigo plugin requires profile synchronization enabled by default \(ATEAM-18863\)](#)
- [Failed deployments reported as successful \(ATEAM-18893\)](#)
- [DigiCert CA with Certificate Enrollment Gateway not supported \(ATEAM-18920\)](#)
- [Documentation does not describe the risks of enabling SAN attributes on request \(ATEAM-19009\)](#)
- [EJBCA plugin ignores the default-query-page-size setting \(ATEAM-19038\)](#)
- [Error when enrolling certificates with a DigiCert CA \(ATEAM-19060\)](#)

Unexpected warning logs (ATEAM-18776)

CA Gateway records unexpected warning logs triggered by the Spring framework. For example:

Check the corresponding BeanPostProcessor declaration and its dependencies/advisors. If this bean does not have to be post-processed, declare it with `ROLE_INFRASTRUCTURE`.

URL not hardcoded on Sectigo configuration (ATEAM-18856)

When integrating a Sectigo CA, the configuration settings require entering a URL setting. However, this URL value should be hardcoded, as is always the following.

```
https://cert-manager.com
```

Error when renewing a Sectigo-issued SSL certificate (ATEAM-18858)

CA Gateway returns an error when trying to renew an SSL certificate using a Sectigo CA.

Sectigo plugin requires profile synchronization enabled by default (ATEAM-18863)

The `enable-ca-profile-sync` setting defaults to `false` for all profiles. However, the Sectigo configuration requires this setting to default to `true`.

Failed deployments reported as successful (ATEAM-18893)

When a CA Gateway deployment fails, the management console incorrectly displays the deployment as successful, even though the process has failed.

Digicert CA with Certificate Enrollment Gateway not supported (ATEAM-18920)

Certificate issuance fails in the following situation:

- The Certificate Authority is Digicert CA
- The client application is Certificate Enrollment Gateway
- The enrollment protocol is ACME

Documentation does not describe the risks of enabling SAN attributes on request (ATEAM-19009)

The CA Gateway documentation does not clearly explain the risks associated with allowing Subject Alternative Name (SAN) attributes in enrollment requests.

Bug resolution: A new section in [Enabling SAN attributes in the enrollment request](#) describes the risks.

EJBCA plugin ignores the default-query-page-size setting (ATEAM-19038)

The plugin to integrate EJB Certificate Authorities ignores the value of the following configuration setting.

```
config:cagw.cert-event-tracking.default-query-page-size
```

Error when enrolling certificates with a DigiCert CA (ATEAM-19060)

CA Gateway returns a "cagw-5000" error when using a DigiCert CA.

Known issues for 3.7.0

CA Gateway 3.7.0 has the following known issues.

- [Mandatory parameters are mutually exclusive \(ATEAM-16246\)](#)
- [subject.certificates field omitted \(ATEAM-16264\)](#)
- [Slashes not supported in passwords \(ATEAM-18325\)](#)

Mandatory parameters are mutually exclusive (ATEAM-16246)

When configuring a CA Gateway client, the following mandatory parameters are mutually exclusive (that is, you must select one but not both).

- Tenant ID
- Integrator ID

However, the Management Console raises an error during validation if any of these values is unselected.

Detected in: CA Gateway 3.0.0 to 3.0.5.


Workaround:

1. Delete the client settings.
2. Recreate the client settings using either the **Tenant ID** or **Integrator ID** parameter.

subject.certificates field omitted (ATEAM-16264)

For performance reasons, the PKIaaS CA Plugin will not honor the `subject.certificates` field in the following endpoint.

```
api/v1/certificate-authorities/{caId}/subjects/dn
```

 Future releases may restore this functionality.

Detected in: CA Gateway 3.0.1 to 3.0.5.

Slashes not supported in passwords (ATEAM-18325)

CA Gateway deployments on appliances do not support slashes ('/') in passwords (such as keystore passwords, truststore passwords, Entrust Profile File passwords, etc.).

Detected in: CA Gateway installations on EDM (Entrust Deployment Manager) and PKI Hub.

4 Requirements

See below for the CA Gateway hardware and software requirements.

- [Docker requirements](#)
- [HSM requirements](#)

Docker requirements


CA Gateway is distributed and operates as a Docker container. Our objective in leveraging Docker is to allow customers to utilize the broad array of Docker capabilities, features, and plug-in drivers. Entrust will make reasonable efforts to support our CA Gateway product on the customer's chosen Docker deployment.

Platform	Version	Note
Docker	20.x	Operation as a Docker container is supported on all OS platforms supporting Docker.

HSM requirements

CA Gateway on Docker supports the following hardware security modules (HSMs).

Hardware	Client driver	Firmware	Guide
Entrust nShield Connect XC	13.9.0 (FIPS 140-2 Level 3 mode supported)	12.60.15 & 12.60.2	CA Gateway 3.x - Entrust nShield Integration Guide
Entrust nShield 5c	13.9.0	13.2.4	
Entrust nShield 5c 10G	Not supported	Not supported	
Thales Luna HSM 7	10.8.0	7.7.1-20	CA Gateway 3.x - Thales Luna Integration Guide
Thales TCT	10.8.0	7.7.1-20	

 Log in trustedcare.entrust.com to download the integration guides.

5 Preparing the deployment

Prepare the CA Gateway deployment as explained in the following sections.

- [Getting the CA Gateway license](#)
- [Downloading the installation files](#)
- [Verifying the downloaded files](#)
- [Loading the CA Gateway Image](#)
- [Tagging the CA Gateway Image](#)
- [Listing the images](#)
- [Generating the systemd service for Podman](#)
- [Creating the configuration and credentials folders](#)
- [Running the Profile Creation Utility](#)
- [Verifying the installation](#)
- [Securing settings with jTinyUAL](#)
- [Obtaining the server certificate](#)

Getting the CA Gateway license

To deploy CA Gateway you need a license. After making the order, you will receive an email from licensing@pki.entrust.com containing:

- The order number.
- A signed license file. Place this file in the configuration folder and grant the required permissions. See [Creating the configuration and credentials folders](#) for details.

i To switch from password-protected licenses to signed licensed files, replace the `zip-path` and `zip-password` settings under `license` with `signed-path`.

Downloading the installation files

You need to download the following installation files from TrustedCare.

File	Description
CA Gateway for Docker Installation	The image file for deploying CA Gateway on Docker.
CA Gateway Config Utility	The tool described in Running the Profile Creation Utility for generating and managing the CA Gateway configuration.

To download the CA Gateway installation files

1. Log in to <https://trustedcare.entrust.com>
2. Go to **PRODUCTS / PKI / Authority**.
3. Click on the CA Gateway version you want to download.
4. Select the **SOFTWARE DOWNLOADS** tab to download the installation files.
5. Select the **DOCUMENTS** tab to download the product documentation.

Verifying the downloaded files


Generate a digest to verify the integrity of each downloaded installation and documentation file. On a Windows machine, you can run the following command line to generate the digest of the `<file>` file.

```
certutil -hashfile <file> SHA256
```

For example:

```
>certutil -hashfile c:\Users\john\Downloads\edm-2.0.2.iso SHA256
SHA256 hash of c:\Users\john\Downloads\edm-2.0.2.iso:
d841d57c7e1433622d219a7dea405935ff593a6831c1c94ba1c9dbde763b5baa
CertUtil: -hashfile command completed successfully.
```

On the **SOFTWARE DOWNLOADS** and **DOCUMENTATION** tabs, click the **Digest** column for each downloaded file and verify the displayed SHA-256 digest matches the generated one.

 Although TrustedCare also displays the MD5 and SHA-1 digests, we recommend using only the SHA-256 algorithm, which is more secure. Further versions of TrustedCare will remove the MD5 and SHA-1 algorithms from the digest list.

Loading the CA Gateway Image

Load the CA Gateway image into Docker or Podman.

```
[docker|podman] load --input cagw-api-<VERSION>-docker.tar.gz
```

Where `<VERSION>` is the CA Gateway version – for example:

```
$ docker load --input cagw-api-3.3-2559.docker.tar.gz
```

Tagging the CA Gateway Image

Once loaded into Docker or Podman, tag the image as the latest.

```
[docker|podman] tag cagw/api:<VERSION> cagw/api:latest
```

We recommend tagging your images with the version information – for example:

```
$ docker tag localhost/cagw/api:3.3 cagw/api:latest
```

Listing the images

To verify that you have the correct image, list the images deployed into Docker or Podman.

```
[docker|podman] image ls
```

Generating the systemd service for Podman

When using Podman instead of Docker, generate a systemd service as explained below.

- [Generating the systems service file](#)
- [Starting the systemd service](#)
- [Checking the systemd service status](#)

Generating the systems service file

Run the following commands to generate the user's systemd service file.

```
mkdir -p $HOME/.config/systemd/user
cd $HOME/.config/systemd/user
podman generate systemd --new --files --name cagw
```

See below for a `container-cagw.service` sample file generated with these commands.

```
# container-cagw.service
# autogenerated by Podman 4.4.1
# Wed Sep 20 03:47:02 EDT 2023

[Unit]
Description=Podman container-cagw.service
Documentation=man:podman-generate-systemd(1)
Wants=network-online.target
After=network-online.target
RequiresMountsFor=%t/containers

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=always
TimeoutStopSec=70
ExecStart=/usr/bin/podman run \
  --cidfile=%t/%n.ctr-id \
  --cgroups=no-common \
  --rm \
  --sdnotify=common \
  --replace \
  -d \
  --name cagw \
  -p 8444:8080 \
  -p 9444:9090 \
  -e LOADER_PATH=/etc/cagw/config/plugins \
  -v /home/myuser/cagw/config:/etc/cagw/config:Z localhost/cagw/api:latest
ExecStop=/usr/bin/podman stop \
  --ignore -t 10 \
```

```

--cidfile=%t/%n.ctr-id
ExecStopPost=/usr/bin/podman rm \
-f \
--ignore -t 10 \
--cidfile=%t/%n.ctr-id
Type=simple
NotifyAccess=all

[Install]
WantedBy=default.target

```

Starting the systemd service

Run the following commands to start the systemd service managing the `cagw` container.

```

systemctl --user daemon-reload
systemctl --user enable container-cagw
systemctl --user start container-cagw.service

```

Checking the systemd service status

Run the following command to check the service status (it must be running).

```

systemctl --user status container-cagw.service

```

Creating the configuration and credentials folders

You need the following folders for the configuration and credential files.

Folder	Contents	Sample path
<HOST_CONFIG>	Host configuration files	/home/myuser/cagw/config
<HOST_CONFIG>/<CREDENTIALS>	Credential files	/home/myuser/cagw/config/credentials

To create the configuration and credentials folders

1. Create the empty folders – for example, running a single command:

```

mkdir -p /home/myuser/cagw/config/credentials

```

2. Copy all configuration and credential files to their respective folders.

 The tool described in [Running the Profile Creation Utility](#) creates sample configuration and credential files for these folders.

3. Run the following commands to make the folder files accessible to the predefined user and group with the 1339 identifier.

```
chown -R :1339 <HOST_CONFIG>
chmod -R g+rx <HOST_CONFIG>
chmod -R g+s <HOST_CONFIG>
```

For example:

```
chown -R :1339 /home/myuser/cagw/config
chmod -R g+rx /home/myuser/cagw/config
chmod -R g+s /home/myuser/cagw/config
```

Running the Profile Creation Utility

The CA Gateway distribution includes the Profile Creation Utility to:

- Create a basic configuration for testing the CA Gateway deployment.
- Migrate or validate an existing configuration.

See below for how to install and run this tool.

- [Installing the Profile Creation Utility](#)
- [Setting the Java path](#)
- [Normalizing exceptional Camel Case keys](#)
- [Normalizing a legacy configuration with the Profile Creation Utility](#)
- [Creating a basic configuration with the Profile Creation Utility](#)
- [Bootstrapping CA Gateway with the generated configuration](#)
- [Validating the configuration with the Profile Creation Utility](#)

Installing the Profile Creation Utility

See below for installing the Profile Creation Utility.

To install the Profile Creation Utility

1. Log in to <https://trustedcare.entrust.com>
2. Navigate to **Products > PKI > authority**
3. Select your CA Gateway release.
4. Click **Download** for **CA Gateway Config Utility**.
5. Extract the contents of the downloaded Zip file.

Setting the Java path

If the Java path is not in your global settings, configure the `JAVA_HOME` variable using the following scripts.

Script	OS	Example
bin/setenv.sh	Linux	export JAVA_HOME=/usr/java/jdk-17.0.6

Script	OS	Example
bin/setenv.bat	Windows	set JAVA_HOME=C:\Program Files\Java\jdk-17.0.6

Normalizing exceptional Camel Case keys

When [Normalizing a legacy configuration with the Profile Creation Utility](#), the tool can be instructed how to normalize exceptional Camel Case keys by making use of the strings listed in the following file.

```
conf/UpperCaseWords.txt
```

By default, this file contains the following lines.

```
DN
DNS
MSCA
AWS
```

Therefore, when normalizing to kebab-case, the utility formats `subjectDNCache` as `subject-dn-cache` instead of `subject-d-n-cache`. Edit this file to add additional exceptions.

Normalizing a legacy configuration with the Profile Creation Utility

Beginning with CA Gateway 2.8 all the keys in the `application.yml` configuration file must meet kebab-case (words separated by a hyphen). If you are reusing a legacy configuration, open a command-line interpreter in the `bin` folder of the Profile Creation Utility and run the following command.


```
cagw-util normalize-yaml <INPUT_YAML> <OUTPUT_YAML>
```

Where:

- `<INPUT_YAML>` is the path of the file containing the YAML configuration to normalize.
- `<OUTPUT_YAML>` is the path for the output file containing the normalized YAML configuration.

For example:

```
cagw-util normalize-yaml c:/test/input.yml c:/test/output.yml
```

 In the output YAML, the normalizer excludes the comments and the keys with null values (`null`, `!!null`, `~`, spaces, and empty lines).

Creating a basic configuration with the Profile Creation Utility

To generate a basic CA Gateway configuration, open a command-line interpreter in the `bin` folder of the Profile Creation Utility and run the following command.

```
cagw-util create-skeleton-config [-f] -pwd -env=<ENVIRONMENT> -host=<HOST> [-l=<LICENSE_FILE>] [-o=<OUTPUT_FOLDER>] -p=<PORT> [-e | -d -hc=<CREDENTIALS_DIR>]
```

For example:

```
cagw-util create-skeleton-config -f -pwd -env cagwtest -host cagw.test.org -l license.json -o c:/test/config -p 8080
```

```
cagw-util create-skeleton-config -f -pwd -env=cagwtest -host=cagw.test.org -l=license.json -o=c:/test/config -p=8080
```

When prompted, enter the password and press ENTER. For example:

```
Enter the Keystore password:
c:\test\config\server.p12 successfully created
c:\test\config\cagw-client-1.p12 successfully created
c:\test\config\ca.p12 successfully created
c:\test\config\application.yml successfully created

Client DN:      cn=cagwtest client 1, o=cagwtest
CAGW Server DN: cn=cagw.test.org, o=cagwtest
CA DN:         cn=cagwtest CA, o=cagwtest

CAGW URL:      https://cagw.test.org:8080/cagw/v1
CAGW API Docs: https://cagw.test.org:8080/cagw/api-docs
Swagger URL:   https://cagw.test.org:8080/cagw/swagger-ui
```

See below for a description of each parameter.

- `-d, --docker`
- `-env, --environment=<ENVIRONMENT>`
- `-f, --force-overwrite`
- `-hc, --host-config-dir=<CREDENTIALS_DIR>`
- `-host, --hostname=<HOST>`
- `-l, --license-file-name=<LICENSE_FILE_NAME>`
- `-m, --msca-proxy`
- `-o, --output-dir=<OUTPUT_DIR>`
- `-p, --port-number=<PORT>`
- `-pwd`

`-d, --docker`

Customize the generated `application.yml` configuration file for Docker environments.

i When using this convenience flag, each file path in the generated `application.yml` configuration file has a base path suited for Docker environments. Alternatively, you can omit this flag and select a customized base path with the `--host-config-dir` command.

`-env, --environment=<ENVIRONMENT>`

Set `<ENVIRONMENT>` as environment name. CA Gateway will use this value when setting unique subject names for the certificates.

Mandatory: Yes.

`-f, --force-overwrite`

Overwrite any existing configuration file.

`-hc, --host-config-dir=<CREDENTIALS_DIR>`

Use `<CREDENTIALS_DIR>` as the base path to reference credential files in the generated `application.yml` configuration file. For example:

```
trust-store: <CREDENTIALS_DIR>/truststore.p12
```

x This flag is mutually exclusive with `--docker`.

Mandatory: No. When omitting both this option and the `--docker` flag, the `application.yml` file assumes an empty value – for example:

```
trust-store: truststore.p12
```

`-host, --hostname=<HOST>`

Set `<HOST>` as the IP address or hostname for CA Gateway URLs.

i The utility populates `<HOST>` into the Subject Alternative Name of the generated server certificate.

Mandatory: Yes.


`-l, --license-file-name=<LICENSE_FILE_NAME>`

Use the `<LICENSE_FILE_NAME>` license, where `<LICENSE_FILE_NAME>` is the name (not the path) of the license file. The `application.yml` configuration file will reference this file using the following path.

```
<CREDENTIALS_DIR>/<LICENSE_FILE_NAME>
```

Where `<CREDENTIALS_DIR>` is the folder selected with one of the following commands.

- `--docker`
- `--host-config-dir`

 This command does not support legacy password-protected license files; it only supports signed license files.

Mandatory: No.


`-m, --msca-proxy`

Customize the generated configuration for supporting Entrust Microsoft CA Proxy.

Mandatory: No.

`-o, --output-dir=<OUTPUT_DIR>`

Save the generated files in the `<OUTPUT_DIR>` folder of the host where the utility is executed.

 Move the generated files to the `<CREDENTIALS_DIR>` folder selected with either the `--docker`, or `--host-config-dir` command.

Mandatory: No. This optional value defaults to the current folder.

`-p, --port-number=<PORT>`

Configure CA Gateway for listening in the `<PORT>` port.

Mandatory: Yes.

`-pwd`

Prompt the user for the keystores password.


Mandatory: Yes.

Bootstrapping CA Gateway with the generated configuration

After [Creating a basic configuration with the Profile Creation Utility](#), you can bootstrap a testing deployment of CA Gateway.

To bootstrap CA Gateway

1. Copy the following files, generated by the `cagw-util` tool, to the host configuration folder.
 - `application.yml`
 - `server.p12`
 - `cagw-client.p12`
 - `ca.p12`

 See [Creating the configuration and credentials folders](#) for details on the host configuration folder path and the required permissions.

2. Run CA Gateway as explained in [Starting up and deploying](#).

application.yml

The YAML file containing the main CA Gateway settings. The tool generates this file with a basic configuration that does not include managed CAs.

server.p12

The keystore and truststore of the CA Gateway server.

cagw-client.p12

The keystore and truststore of the CA Gateway client. On a Windows host, you can: double-click this file, follow the dialogue to add the key and certificate into a Windows certificate store, and use the browser for connecting to one of the CA Gateway URLs.

ca.p12

The keystore and truststore of an ephemeral CA for starting CA Gateway.

Validating the configuration with the Profile Creation Utility

To validate the `application.yml` configuration file, open a command-line interpreter in the `bin` folder of the Profile Creation Utility and run the following command.

```
cagw-util validate-yaml <APPLICATION_YML>
```

Where `<APPLICATION_YML>` is the path of the `application.yml` configuration file – for example:

```
$ cagw-util validate-yaml application.yml
Validating YML File...
YAML File is Valid but only in syntax, content has not been evaluated.
```

When detecting any error, the tool returns detailed information – for example:

```
$ cagw-util validate-yaml application.yml
Validating YML File...
YAML File failed to validate: mapping values are not allowed here
  in 'reader', line 51, column 21:
    properties:
```

Verifying the installation

Verify that you have successfully installed CA Gateway as a Docker container. Run this quick and unsecured test before configuring the SSL and CA connections.

- [Running the latest CA Gateway image](#)
- [Browsing to an endpoint](#)

Running the latest CA Gateway image

In a new terminal, run the latest CA Gateway image, mapping the internal CA Gateway port 8080 to the localhost port 8444.

```
[docker|podman] run -d --rm -p 8444:8080 -p 9444:9090 -v <HOST_CONFIG>:/etc/cagw/config cagw/api:latest
```

Where `<HOST_CONFIG>` is the folder described in [Creating the configuration and credentials folders](#). For example:

```
docker run -d --rm -p 8444:8080 -p 9444:9090 -v /home/myuser/cagw/config:/etc/cagw/config cagw/api:latest
```

See the table below for a description of each option.

Option	Description
-d, --detach	Launches the container in the background. Skip this option to see the CA Gateway log output while running, although it might terminate CA Gateway when closing the terminal.
-p, --expose	Binds a port.
--rm	Cleans up the container and removes the file system when the container exits.
-v, --volume	Maps the <code><HOST_CONFIG></code> folder (described in Creating the configuration and credentials folders) with a folder in the Docker container.


Refer to the Docker documentation for more details on the supported options.

Browsing to an endpoint

In the CA Gateway server, open a Web browser and navigate to:

```
http://localhost/cagw/v1
```


You can replace `localhost` with the server's hostname or IP address. This URL omits the port number because the CA Gateway port is mapped to port 8444.

 See [Checking the CA Gateway health](#) for how to check the CA Gateway health endpoints.

Securing settings with jTinyUAL

The CA Gateway deployment provides the jTinyUAL utility for securing sensitive information in plain text files.

- [Checking entropy](#)
- [Creating jTinyUAL files to protect settings](#)
- [Creating a jTinyUAL properties file](#)

 If you secure a setting using jTinyUAL and specify a plaintext version of the same setting in another CA Gateway configuration file (such as `application.yml`), the jTinyUAL-protected setting takes precedence.


Checking entropy

The entropy required by the TinyUAL library of the jTinyUAL utility may exceed the available entropy on the CA Gateway system. If TinyUAL does not have sufficient entropy to encrypt settings, CA Gateway may experience significant delays during startup.

To check if your host system has enough entropy, run the following command.

```
head -c 8192 /dev/random | hexdump
```

If the command completes almost immediately, the server has enough entropy. However, if it takes several minutes, the server does not have enough entropy, and you must install a daemon.

 Run the below commands in the host machine because the Docker containers use the entropy provided by this machine.

If your system has a random number generator, install the `rngd` daemon. For example, in CentOS:

```
sudo yum -y install rng-tools
sudo systemctl start rngd
sudo systemctl enable rngd
```

If your system does not have a random number generator, install the `haveged` daemon.

```
sudo yum -y install haveged
sudo systemctl start haveged
sudo systemctl enable haveged
```


Creating jTinyUAL files to protect settings

For each setting you want to protect, create a separate jTinyUAL file with the following contents.

```
decrypted=<VALUE>
```

Where `<VALUE>` is the plaintext value of the setting you want to encrypt. When encrypting the file, jTinyUAL will change `decrypted` to `encrypted`, and replace the plaintext value with an encrypted value. For example:

```
#Sat Jun 29 16:34:47 EDT 2019
encrypted=VEl0WVVBTAf0F/
1mY3XEXgyRLnP3q05cjToYU1WG7Nc9n+617INxg2XkPtVYE5ZNRJkQxVzqrVFcbmV10rd4snp/HNU91jnIa/
SthwG1gomakWgE+x0zLJK8+cn5ggSIF6IrnSRqhxCxppENLSZGar0tUARNxLLA9+okmwTUF+lWRRWab+06nWj
mRov9ax+CTSTzuFhGjajCuBoJ5qPmIr48hu+5+z5QCGeGScYphcaw1reTuWS034BZLdiJq7qq0Zmcw05xIdE
alEnSXTdRSDYQKY4wYeLxAhgZP1hi10oTcFYtZk+xDZACpu0j42WzSEh9YlTJtgUYb6nlrBN1XcPQo18CrKFL
RCBcPjieU3IicI+LUfX0K8RSHqz/lv1c46Vo8v/
wYFjJyZQp075wyUJvzXte8pKXuMgptoFnpr+ty13Gf4M0rW7zkqX5FFUaV/
LAszy2j fux8Az1dMnJatM+ZJ47N80hh3B6a+q8Lst3cYisBuLqNEPFN2j2YwG6jF0Eg1qXENrxchneLk5o9KK
P/0qZI80sZIS9UAiep1fCuyz0w5+AmAanUqVgD0sScWLka1o311Ay79MqI4h1gAD9mZycYpUSKl7mL0hL/
uKHGbTy3egg34l8R/
Lz70+XAXtf07pt0Z4E7csPZfGEB7PVKNUk0XRGIGxXjdqXg2jVfCpoBUutHpc876npDPsnPBLBA2QY6yPO
```

 Do not share jTinyUAL files between instances of CA Gateway. These files only work for one CA Gateway instance.

Creating a jTinyUAL properties file

The jTinyUAL properties file maps each CA Gateway setting you want to protect with the jTinyUAL file that contains the setting value. In this file, add `<SETTING>=<FILE>` pairs where:

- `<SETTING>` is the CA Gateway setting in Java properties format.
- `<FILE>` is the name of the file that contains the setting value.

For example:

```
cagw.authorities.managed_cas.example_ca_id.properties.admin_epf_password=example_ca_admin_epf_password.tua1
```

After creating the jTinyUAL properties file:

1. Save the file (for example, as `tua1.properties`) in the same folder containing the jTinyUAL files.
2. Add the file path to the `tua1.properties` setting.

Obtaining the server certificate

CA Gateway requires a digital certificate for securing communications between the CA Gateway and authorized clients. See below to generate this certificate for a production environment.

- [Generating the server key pair](#)
- [Obtaining the key pair CSR](#)
- [Obtaining the server certificate](#)
- [Importing the server certificate into the keystore](#)

- [Importing CA certificates into a truststore](#)

⚠ The certificate must contain the server's fully qualified domain name (FQDN) as a DNS type Subject Alternative Name (subjectAltName) extension.

Generating the server key pair

To generate the key pair of the server, run the following command.

```
keytool -genkeypair -alias <ALIAS> -dname <DN> -keyalg <KEYALG> -keysize <KEYSIZE>
-sialg sha256WithRSA -ext san=dns:<DNS> -keystore <KEYSTORE> [-keypass <KEYPASS>] [-
storepass <STOREPASS>]
```

See the following table for a description of each flag.

Flag	Value
-alias	An alias for the key pair.
-dname	The DN for the key pair (and later, the certificate). Use the DN format expected by the CA that will issue the certificate.
-keyalg	The algorithm for the key pair (for example, RSA).
-keysize	The Key size. Select a secure key size (for example, 2048).
-ext	The DNS-type value of the Subject Alternative Name (subjectAltName) extension.
-keystore	The full path of the keystore file. If the keystore does not exist, the keytool utility will create it.
-keypass	The password of the private key. When you omit this option, the tool prompts for a password.
-storepass	The password for the keystore. When you omit this option, the tool prompts for a password.

Obtaining the key pair CSR

Create a Certificate Signing Request (CSR) by entering the following command:

```
keytool -certreq -alias <ALIAS> -file <FILE> -storetype pkcs12 -keystore <KEYSTORE>
[-storepass <STOREPASS>]
```

For example:

```
> keytool -genkeypair -alias example_alias -dname "cn=CA Gateway,ou=CA
Entry,o=Example,c=US" -keyalg RSA -keysize 2048 -sigalg sha256WithRSA -ext
san=dns:domain.example.com -keystore /CAGW/config/keystore.ks
> keytool -certreq -alias example_alias -file /tmp/cagw/cagw_csr.txt -keystore /CAGW/
config/keystore.ks
```

See the following table for a description of each option.

Option	Value
-alias	The alias previously specified when Generating the server's key pair.
-file	The full path of the CSR file.
-keystore	The full path of the keystore file.
-storepass	The password of the keystore. When you omit this option, the tool prompts for a password.

Obtaining the server certificate

Issue the certificate with either:

- Your Entrust Certificate Authority.
- A trusted certificate provider such as the Entrust Certificate Services at store.entrust.com.

Importing the server certificate into the keystore

Import the certificate into the keystore:

```
keytool -importcert -alias <ALIAS> -file <FILE> -keystore <KEYSTORE>
```

For example:

```
keytool -importcert -alias example_alias -file /tmp/cagw/cagw_cert.p7b-keystore /
home/myuser/cagw/config/keystore.ks
```

See the following table for a description of each option.

Option	Value
-alias	The alias previously specified when Generating the server's key pair.
-file	The full path of the PKCS #7 file containing the certificate and the certificate chain.

Option	Value
-keystore	The full path of the keystore file.

Importing CA certificates into a truststore

For each managed Certificate Authority, CA Gateway requires the following certificates.

CA type	Required certificates
Root	The self-signed root CA certificate.
Subordinate	The complete CA certificate chain, from the subordinate CA certificate up to the root CA certificate.

Import these certificates in either:

- The Truststore used when Importing the server certificate into the keystore.
- A new Truststore.

To import a CA certificate into a truststore using the Java `keytool` utility, run the following command.

```
keytool -importcert -trustcacerts -alias <ALIAS> -file <FILE> -keystore <KEystore> [-storepass <STOREPASS>]
```

For example:

```
keytool -import -trustcacerts -alias managed_ca1 -file /tmp/cagw/managed_ca1.cer -keystore /home/myuser/cagw/config/keystore.ks
```


See the following table for a description of each parameter.

Option	Value
-alias	The alias of the CA certificate.
-file	The full path of the CA certificate file.
-keystore	The full path of the Java keystore file. If not present, the keystore is created.
-storepass	The password of the Java keystore. When you omit this option, the tool prompts for a password.

6 Integrating Certificate Authorities

The below sections explain how to integrate CA Gateway with Certificate Authorities of different providers.

- [Integrating a DigiCert CA](#)
- [Integrating a GlobalSign CA](#)
- [Integrating a Microsoft CA](#)
- [Integrating an AWS CA](#)
- [Integrating an EJBCA](#)
- [Integrating an Entrust CA](#)
- [Integrating a Sectigo CA](#)

 See *CA Gateway - Configuration Reference* for the supported fields and Docker-specific considerations on referencing files.

Integrating a DigiCert CA

DigiCert CAs do not require preliminary integration steps. See [Configuring](#) for instructions on:

- Configuring a DigiCert CA.
- Add a DigiCert CA certificate profile.


See [DigiCert CA capabilities](#) for a complete description of the operations supported by these CAs.

Integrating a GlobalSign CA

GlobalSign Certificate Center (GCC) authorities do not require preliminary integration steps. See [Configuring](#) for instructions on:

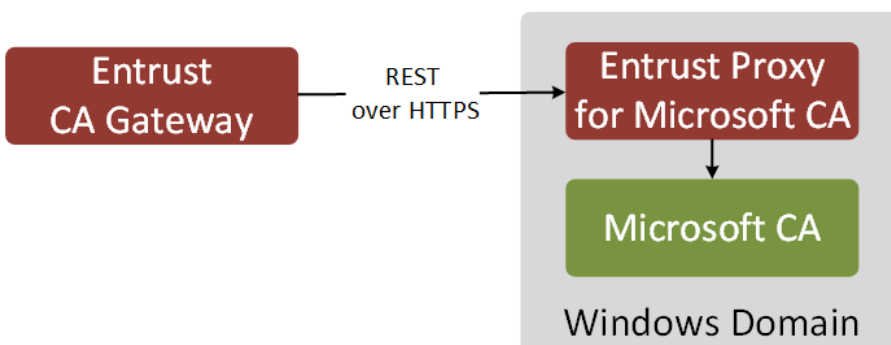
- Configuring a GlobalSign Certificate Center (GCC).
- Add a GlobalSign Certificate Center (GCC) certificate profile.

See [GlobalSign capabilities](#) for a complete description of the operations supported by these CAs.

 In the current CA Gateway release, GlobalSign integration is supported for testing purposes only.

Integrating a Microsoft CA


As illustrated by the following figure, CA Gateway manages each Microsoft CA instance through an Entrust Proxy for Microsoft CA.



In this architecture, CA Gateway is a client of Microsoft CA. See in the following sections how to configure the Entrust Proxy for Microsoft CA and CA Gateway to manage Microsoft CAs.

- [Adding Microsoft Management Console snap-ins](#)
- [Enabling supply in the request](#)
- [Configuring Request Handling in the Microsoft CA](#)
- [Enabling SAN attributes in the enrollment request](#)
- [Installing the Entrust Proxy for Microsoft CA](#)
- [Creating RA agents](#)
- [Issuing the SSL certificates](#)
- [Generating a client keystore for Entrust Proxy for Microsoft CA](#)
- [Generating a truststore for CA Gateway](#)
- [Generating the server keystore of the Entrust Proxy for Microsoft CA](#)
- [Running the Entrust Proxy for Microsoft CA](#)

See [Microsoft CA capabilities](#) for a complete description of the operations supported by these CAs.

 Only Microsoft Enterprise CA is supported; standalone CA is not supported.

Adding Microsoft Management Console snap-ins

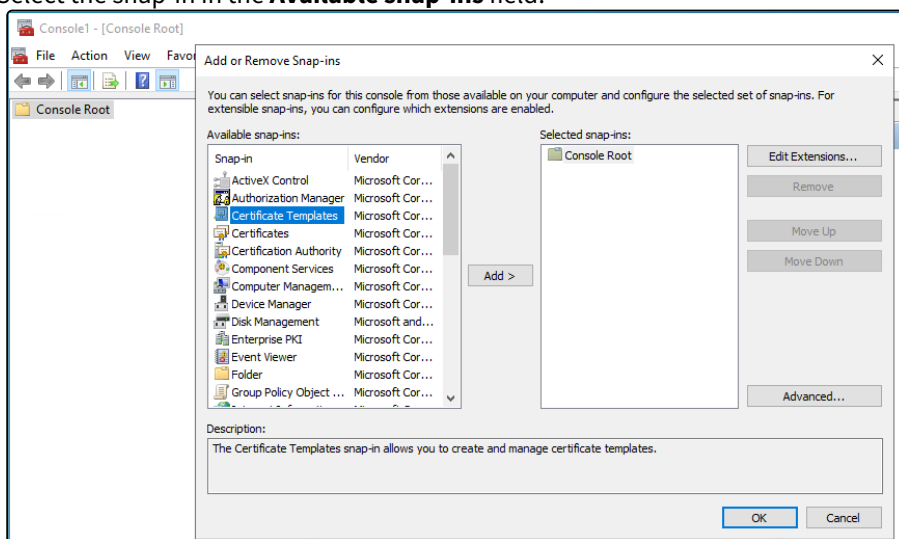
Run the Microsoft Management Console and add the following snap-ins.

- Certificate Templates
- Certificates
- Certificate Authorities

See below for how to add a span-in to the Microsoft Management Console.

To add a Microsoft Management Console snap-in

1. Log into the Microsoft CA server machine.
2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open Microsoft Management Console.
4. Select **File > Add Remove Snap-In**.
5. Select the snap-in in the **Available snap-ins** field.



6. Click **Add** to include the snap-in in the **Selected snap-ins** field.

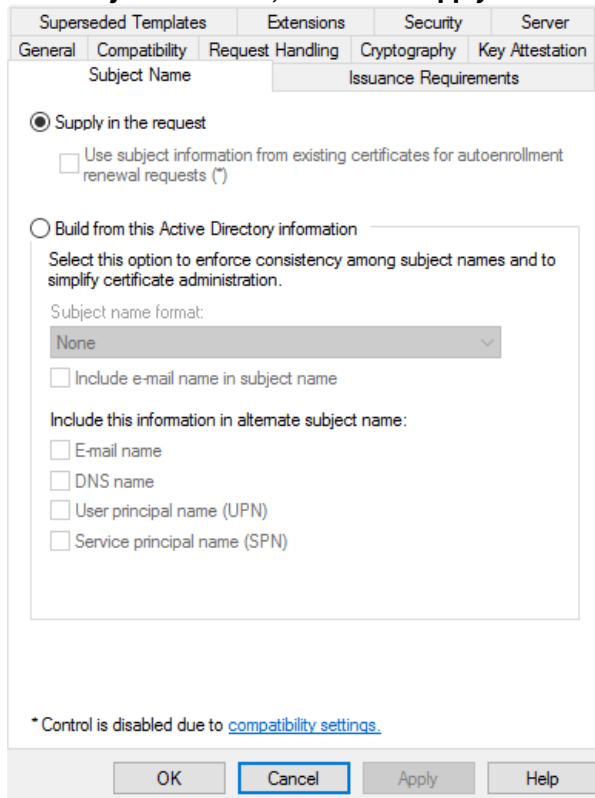
7. Click **OK**.

Enabling supply in the request

In all the managed Microsoft CA templates for issuing entity certificates, make sure that the Subject Name is supplied by the certificate request.

To enable supply in the request in a template

1. Log into the Microsoft CA server machine.
2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open Microsoft Management Console.
4. Go to **Certificate Authority**.
5. **Right-click Certificate Templates and select Manage.**
6. Right-click the template and select **Properties**.
7. In the **Subject Name** tab, enable the **Supply in the request** radio button.



Superseded Templates Extensions Security Server

General Compatibility Request Handling Cryptography Key Attestation

Subject Name Issuance Requirements

Supply in the request

Use subject information from existing certificates for autoenrollment renewal requests (*)

Build from this Active Directory information

Select this option to enforce consistency among subject names and to simplify certificate administration.

Subject name format:

None

Include e-mail name in subject name

Include this information in alternate subject name:

E-mail name

DNS name

User principal name (UPN)

Service principal name (SPN)

* Control is disabled due to [compatibility settings](#).

OK Cancel Apply Help

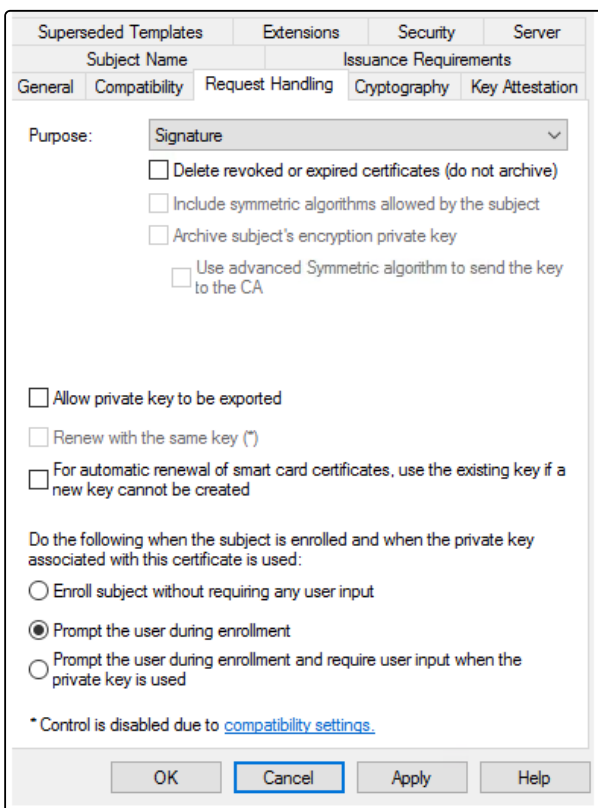
Configuring Request Handling in the Microsoft CA

If the Microsoft CA settings configure **Request Handling** as follows.

Parameter	Value
Purpose	Signature
Delete revoked or expired certificates	

Parameter	Value
Allow private key to be exported	
For automatic renewal of smart card certificates, use the existing key if a new key cannot be created	
Do the following when the public subject is enrolled and when the private key associated with this certificate is used	Prompt the user during enrollment

As we see, the **Archive subject's encryption private key** option is disabled when selecting the **Signature** template.



The screenshot shows the 'Security' tab of the 'Issuance Requirements' dialog box, specifically the 'Cryptography' section. The 'Purpose' is set to 'Signature'. The following options are visible:

- Delete revoked or expired certificates (do not archive)
- Include symmetric algorithms allowed by the subject
- Archive subject's encryption private key (disabled)
- Use advanced Symmetric algorithm to send the key to the CA
- Allow private key to be exported
- Renew with the same key (*)
- For automatic renewal of smart card certificates, use the existing key if a new key cannot be created

Below these options, there are radio button options for enrollment behavior:

- Enroll subject without requiring any user input
- Prompt the user during enrollment
- Prompt the user during enrollment and require user input when the private key is used

A note at the bottom states: * Control is disabled due to [compatibility settings](#).

Enabling SAN attributes in the enrollment request

Microsoft CA can be configured to allow Subject Alternative Name (SAN) values in enrollment requests and include them in the issued certificates.

- [Security risks of allowing SAN fields in enrollment requests](#)
- [Enabling the Config_CA_Accept_Request_Attributes_SAN flag](#)

⚠ If you don't enable this feature, you will lose the capability to extend the SAN attributes of the issued certificates with SAN from the certificate request.

Security risks of allowing SAN fields in enrollment requests

Allowing Subject Alternative Name (SAN) fields in enrollment requests creates a serious security risk. Once enabled, any certificate template may accept alternate names regardless of how the template defines the subject. An attacker could exploit this to obtain a certificate with an alternate name and impersonate another user.

For this reason, the CA must limit enrollment permissions to only trusted accounts. Specifically, the security model described in [Installing the Entrust Proxy for Microsoft CA](#) focuses on allowing only the Proxy Admin to have enrollment permission.

Enabling the Config_CA_Accept_Request_Attributes_SAN flag

If you want Microsoft CA to accept Subject Alternative Name (SAN) fields in enrollment requests, enable the following flag.

```
Config_CA_Accept_Request_Attributes_SAN
```

To enable Config_CA_Accept_Request_Attributes_SAN flag

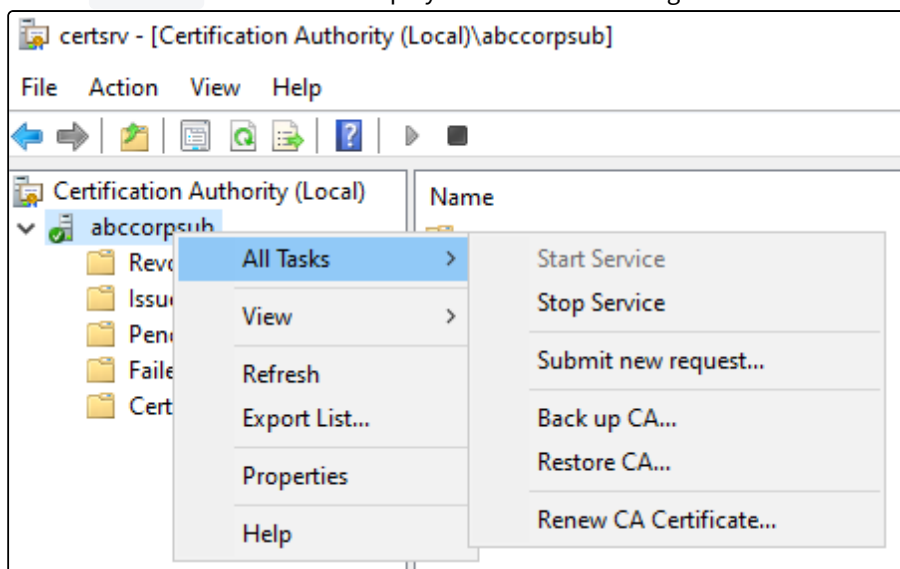
1. Log in to the Windows machine hosting the Microsoft CA server.
2. Run the `regedit` command to open the Registry Editor.
3. Select the following registry key (`<CA_CN>` is the Common Name of the Microsoft CA).

```
HKLM\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\<CA_CN>/  
PolicyModules/CertificateAuthority_MicrosoftDefault.Policy/EditFlags
```

4. Calculate an OR of the current key value and `0x000040000` . For example, if the current value is `11014e` , calculate:

```
0x00011014e OR 0x000040000 = 0x0015014e
```

5. Set the OR result as the new key value.
6. Run the `certsrv` command to display the CA service settings.



7. In the navigation tree, right-click the CA name.
8. Select **All Tasks > Stop service** to stop the Microsoft CA server.
9. Select **All Tasks > Start service** to restart the Microsoft CA server.

Installing the Entrust Proxy for Microsoft CA

Install the Entrust Proxy for Microsoft CA, as explained in the following sections.

- [System requirements for the Entrust Proxy for Microsoft CA](#)
- [Creating a Proxy Admin account](#)
- [Configuring the Windows domain account](#)
- [Adding the Windows domain account to the logon as a service policy](#)
- [Installing Java](#)
- [Downloading the Entrust Proxy for Microsoft CA installer](#)
- [Configuring logs](#)
- [Running the Entrust Proxy for Microsoft CA installer](#)
- [Uninstalling the Entrust Proxy for Microsoft CA](#)

System requirements for the Entrust Proxy for Microsoft CA

To install the Entrust Proxy for Microsoft CA, you need a machine with Windows Server 2016 (x64) or above.

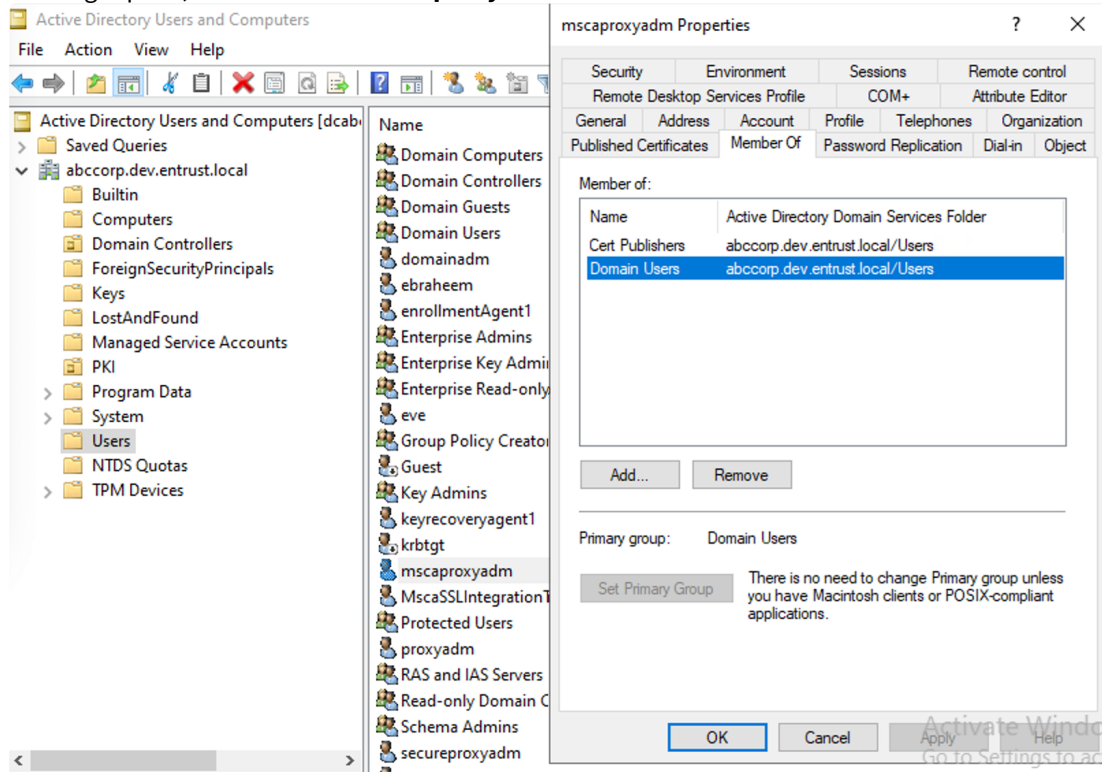
Creating a Proxy Admin account

Create a **Proxy Admin** user belonging to the **Cert Publishers** and **Domain User** groups.

To create a proxy admin user

1. Press **Win + R** to open the **Run** dialog.
2. Type `dsa.msc` and press Enter.
3. Select **Users** under the Windows domain

4. In the right pane, double-click on **mscaproxyadm**.



5. Create a **Proxy Admin** user

6. Add the users only to the **Cert Publishers** and **Domain User** groups.

Configuring the Windows domain account

Configure the Windows login account of the Entrust Proxy for Microsoft CA. See below for the supported combinations when the Entrust Proxy for Microsoft CA and the Domain Controller share the same server or run on different servers.

User	Service startup type	Same server	Different servers
A local service account	Automatic or Automatic (Delayed Start)	✓	✓
A domain user	Automatic (Delayed Start)	✓	✗

In either case, enable only the following user permissions.

- Issue and Manage Certificates
- Request Certificates
- Read

Adding the Windows domain account to the logon as a service policy

Add the new Windows account to the **Logon as a service** policy.

To add the Windows domain account to the logon as a service policy

1. Press **Windows + R** to open the **Run** window.
2. Type `secpol.msc` and press Enter.
3. In the **Local Security Policy** window, navigate to **Local Policies > User Rights Assignment**.
4. In the right pane, double-click on **Log on as a service**.
5. Click **Add User or Group**,
6. Select the account created in the previous section, [Creating a Proxy Admin account](#)
7. .
8. Click **OK**,
9. Click **Apply**.

Installing Java

Install Java on the machine that will host the Entrust Proxy for Microsoft.

To install Java

1. Log in using the account created in the previous section, [Creating a Proxy Admin account](#).
2. Install one of the following LTS (Long Term Support) Java distributions.
 - Oracle Java x86_64 version 17
 - OpenJDK 17
 - AdoptOpenJDK 17
3. Set the `JAVA_HOME` environment to the Java installation path.
4. Extend the value of the `PATH` environment variable to:

```
%JAVA_HOME%\bin
```

5. Run the following command to check the Java version and architecture details.


```
java -XshowSettings:properties -version
```

Downloading the Entrust Proxy for Microsoft CA installer

Download and extract the Entrust Proxy for Microsoft CA installer files.

To download the Entrust Proxy for Microsoft CA installer

1. Log in trustedcare.entrust.com
2. Go to **PRODUCTS > Cryptographic Security Platform**
3. Select the latest version.
4. Click the download link of the Entrust Proxy for Microsoft CA.
5. Unzip the compressed file contents to your selected installation directory on the Windows machine – for example, in `c:\mscaproxy`

 Installing into `c:\Program Files` may not be functional due to Windows privilege enforcement.

Configuring logs

Optionally, edit the configuration files to modify the default log recording settings.

Configuration file	Parameter	Value
MSCAProxy.xml	logpath	The folder where to save logs.
config\application.yml	com.entrust.mscaproxy	The supported log levels. Supported values in increasing severity are <code>TRACE</code> , <code>DEBUG</code> , <code>INFO</code> , <code>WARN</code> , <code>ERROR</code> , <code>FATAL</code> and <code>OFF</code> .

i If you edit these pages after starting Entrust Proxy for Microsoft, run the `MSCAProxy.exe restart` command to restart it.

For example, adding the following code to the `config\application.yml` file sets the log level to `INFO`.

```
logging:  
  level:  
    root: INFO  
    com.entrust.mscaproxy: INFO
```

Running the Entrust Proxy for Microsoft CA installer

See below for running the Entrust Proxy for Microsoft CA installer and registering the Entrust Proxy for Microsoft CA as a Windows service.

To run the Entrust Proxy for Microsoft CA installer

1. Log in to a Windows machine as the Proxy Admin user (created in section [Creating a Proxy Admin account](#)).
2. Run the following command.

```
MSCAProxy.exe install /p
```

3. When prompted, type the `<domainName>` domain name and the `<proxyAdminUserName>` Proxy Admin username. Supported formats are:
 - User Principal Name (UPN) – for example: `<domainName>@<proxyAdminUserName>`
 - The Down-Level Logon Name – for example: `<domainName>\<proxyAdminUserName>`
4. Type the password of the Proxy Admin user.
5. Type "y" to allow the log-on as a service.

i The installer does not wait for you to press the **Enter** key.

Uninstalling the Entrust Proxy for Microsoft CA

Run the following command as an administrator in case you want to uninstall the Entrust Proxy for Microsoft CA.

```
MSCAProxy.exe uninstall
```

Creating RA agents


Create RA recovery and enrollment agents as explained below.

- [Creating the CA enrollment agents](#)
- [Creating RA recovery agents](#)
- [Creating RA enrollment agents](#)

Creating the CA enrollment agents

You must create a CA Enrollment Agent (EA) before creating the RA recovery agents and the RA enrollment agents.

- [Publishing the enrollment template](#)
- [Creating an enrollment certificate for the CA Administrator](#)

 A CA enrollment agent is self-enrolled and internal to the CA, while a RA enrollment agent is co-located with CA Gateway.

Publishing the enrollment template


If not already published, publish the enrollment agent template as explained in this section.

To publish the enrollment agent template

1. Log into the Microsoft CA server machine.
2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open Microsoft Management Console.
4. Under the certificate authority name, right-click **Certificate Templates**.
5. Select **New > Certificate Template to issue**.
6. Select **Enrollment Agent**.

Creating an enrollment certificate for the CA Administrator

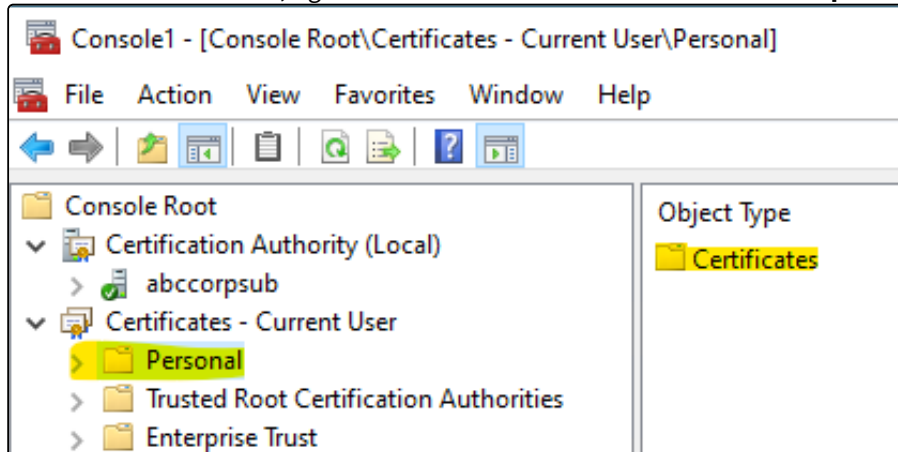
Create an enrollment certificate for the CA administrator user of the Microsoft CA server.

 Do not export the CA administrator's enrollment key.

To create an enrollment certificate for the administrator

1. In the Microsoft CA server machine, run MMC.

2. Under the **Personal** node, right-click **Certificates** and select **All Tasks > Request New Certificate**.



3. Follow the wizard instructions. When prompted, select the **Enrollment Agent** template.

Creating RA recovery agents

If you want to store and recover keys generated by the Microsoft CA, create one or more recovery agents as explained below.

To create a recovery agent

1. Log in to the Microsoft CA server machine.
2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open Microsoft Management Console.
4. Under the Certificate Authority node, right-click **Certificate Template**, and select **Manage**.
5. Right-click **Key Recovery Agent** and select **Duplicate Template**.
6. Configure the following settings in each tab of the **Properties of the New Template** dialog.
 - [General](#)
 - [Request Handling](#)
 - [Issuance Requirements](#)
 - [Security](#)
7. Under the Certificate Authority node, right-click **Certificate Template** and select **New > Certificate Template to issue**.
8. Select the newly created template.
9. Create a user in Active Directory.
10. Under the **Personal** node, right-click **Certificates** and select **All Tasks > Advanced Operations > Enroll On Behalf Of**
11. Follow the wizard instructions. When prompted, select the newly created user.
12. Right-click the issued certificate and select **Export**.
13. Follow the wizard instructions. In the **Export Private Key** dialog, select **Yes, export the private key**.

General

Click this tab and write a name for the new template in the **Template display name** field.

Request Handling

Click this tab and check the **Allow private key to be exported** box.

Issuance Requirements

Click this tab and set the following values.

Parameter	Value
CA certificate manager approval	Disable this option
This number of authorized signatures	1
Policy type required in signature	Application policy
Application Policy	Certificate Request Agent

Security

Click this tab and assign the following permissions to users and user groups.

Permission	Authenticated user	Proxy Admin
Full Control		
Read	✓	✓
Write		
Enroll		✓
Autoenroll		

Creating RA enrollment agents

To publish the issued certificates in Active Directory, you need one or more RA (Registration Authority) enrollment agents. See below for the supported credential generation modes.

- [Creating RA enrollment agent credentials in a keystore file](#)
- [Creating RA enrollment agent credentials in a PKCS#11 HSM](#)

Creating RA enrollment agent credentials in a keystore file

You can create the RA enrollment agent credentials in the following file formats.

- PKCS#12 (Personal Information Exchange Syntax Standard).
- JKS (Java KeyStore).
- JCEKS (Java Cryptography Extension KeyStore).
- PFX (Personal Information Exchange).

See the example below for how to create them in PKCS#12.

To create RA enrollment agent credentials in a keystone file

1. Log into the Microsoft CA server machine.

2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open Microsoft Management Console.
4. Under the Certificate Authority node, right-click **Certificate Template**, and select **Manage**.
5. Right-click **Enrollment Agent** and select **Duplicate Template**.
6. Configure the following settings in each tab of the **Properties of the New Template** dialog.
 - [General](#)
 - [Request Handling](#)
 - [Issuance Requirements](#)
 - [Security](#)
7. Under the Certificate Authority node, right-click **Certificate Template** and select **New >Certificate Template to issue**.
8. Select the newly created template.
9. Create a user in Active Directory.
10. Under the **Personal** node, right-click **Certificates** and select **Tasks > Advanced Operations > Enroll On Behalf Of**.
11. Follow the wizard instructions. When prompted, select the newly created user.
12. Right-click the issued certificate and select **Export**.
13. Follow the wizard instructions. In the **Export Private Key** dialog, select **Yes, export the private key**.

General

Click this tab and write a name for the new template in the **Template display name** field.

Request Handling

Click this tab and check the **Allow private key to be exported** box.

Issuance Requirements

Click this tab and set the following values.

Parameter	Value
This number of authorized signatures	1
Policy type required in signature	Application policy
Application Policy	Certificate Request Agent

Security

Click this tab and assign the following permissions to users and user groups.

Permission	Authenticated user	Proxy Admin
Full Control		
Read	✓	✓

Permission	Authenticated user	Proxy Admin
Write		
Enroll		✓
Autoenroll		

Creating RA enrollment agent credentials in a PKCS#11 HSM

When creating enrollment agents for the Microsoft CA, you can generate keys in a PKCS#11 HSM along with a CSR. When processing this CSR, the Microsoft CA issues a certificate chain for the RA Enrollment Agent that you can import into the HSM to pair with the private key.

 See the integration guides of the supported HSM for the required operations.

Issuing the SSL certificates

CA Gateway and the Entrust Proxy for Microsoft CA communicate with HTTP over SSL using mutual authentication. Thus, two SSL certificates are required:

- A server SSL certificate for the Entrust Proxy for Microsoft CA.
- A client authentication certificate for CA Gateway.

You can obtain both SSL certificates from any CA. Those steps are outside the scope of this document.


Generating a client keystore for Entrust Proxy for Microsoft CA

Generate a keystore containing:

- The private key of Entrust Proxy for Microsoft CA for client authentication.
- The key's certificate.
- The certificate's chain.

See below the required steps.

- [Creating a client authentication template for Microsoft CA](#)
- [Certifying a client key pair for Entrust Proxy for Microsoft CA](#)
- [Importing a key pair and a certificate in the client keystore for Entrust Proxy for Microsoft CA](#)
- [Deleting temporary files of the client keystore Entrust Proxy for Microsoft CA](#)

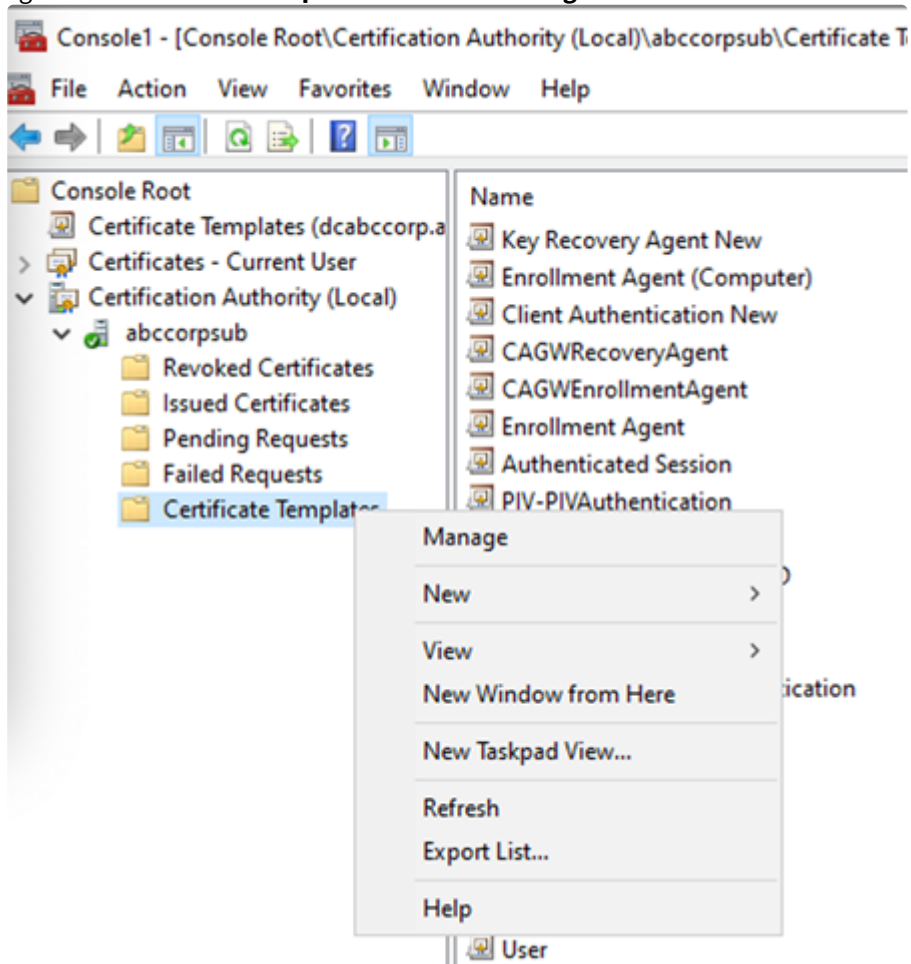
 When configuring CA Gateway, you will later assign the path of this file to the `client-cert-key-store` parameter described in [Microsoft CA properties](#).

Creating a client authentication template for Microsoft CA

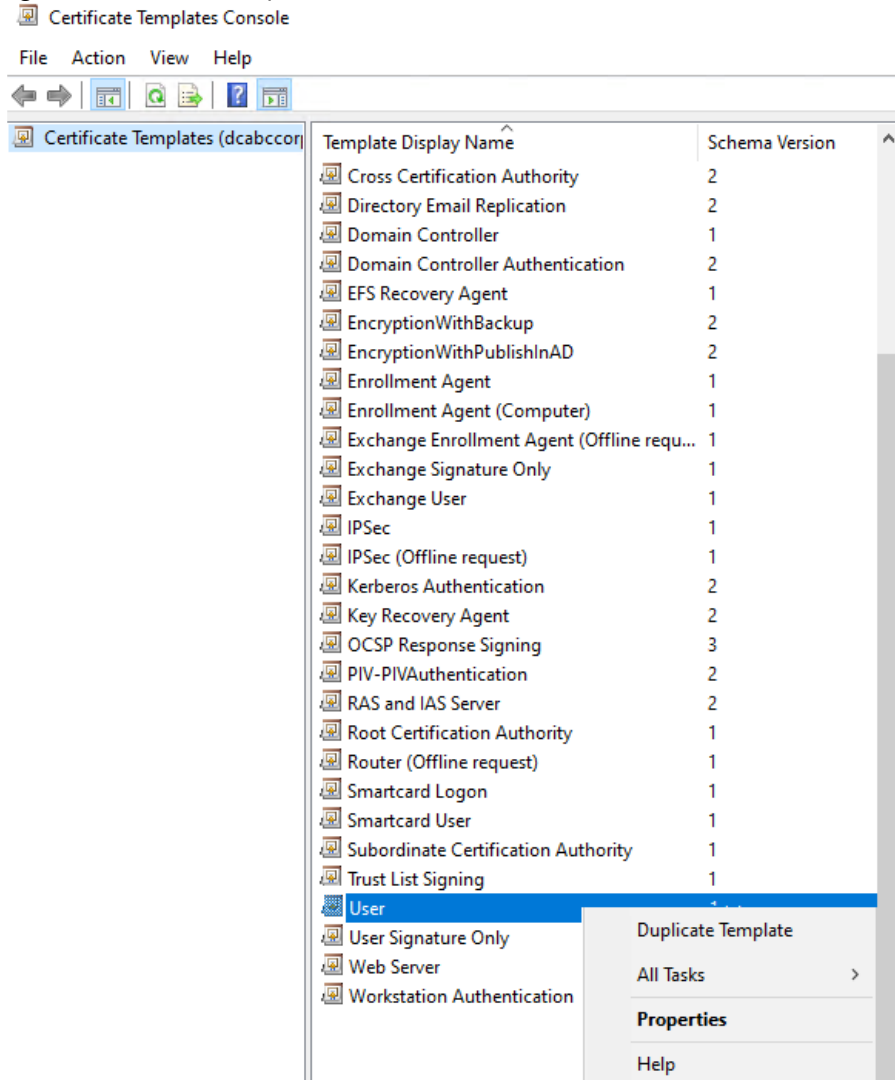
Create an authentication template for generating the client keystore. For example, you can copy an existing template and configure it as explained below.

To create a client authentication template for Microsoft CA

1. Log in to the Microsoft CA server machine.
2. Press **Win + R** to open the **Run** dialog.
3. Type "mmc" and press Enter to open the Microsoft Management Console.
4. Go to **Certificate Authority**.
5. Right-click **Certificate Templates** and select **Manage**.



6. Right-click the **User** template and select **Duplicate Template**.



7. In the template properties dialog, configure the settings described below.

- General
- Security
- Subject Name
- Extensions

8. Click **OK** to close the dialog.

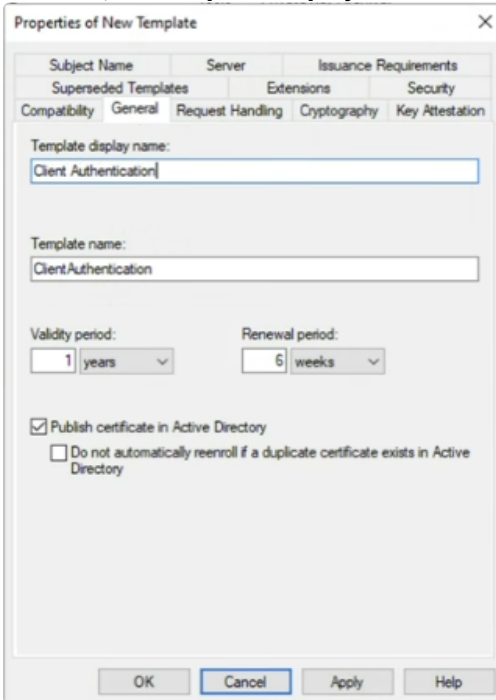
9. Go to **Certificate Authority**.

10. Right-click **Certificate Templates** and select **New >Certificate Template to Issue**.

11. Select **Client Authentication** from the list.

General

In this tab, set **Template display name** to **Client Authentication**.



The screenshot shows the 'Properties of New Template' dialog box with the 'General' tab selected. The 'Template display name' field is set to 'Client Authentication' and the 'Template name' field is set to 'ClientAuthentication'. The 'Validity period' is set to 1 year and the 'Renewal period' is set to 6 weeks. The checkbox 'Publish certificate in Active Directory' is checked, and the checkbox 'Do not automatically reenroll if a duplicate certificate exists in Active Directory' is unchecked. The 'Cancel' button is highlighted.

Security

In this tab:

- Grant necessary permissions to a user group – for example, click **Read, Write, Enroll** for the **Domain Admins** group.
- Deselect the **Write** and **Enroll** permissions from the **Authenticated Users** group.
- Remove unnecessary groups.

Subject Name

In this tab, enable the **Supply in request** radio button.



Click **OK** to close the warning pop-up message.

Extensions

In this tab, edit **Application Policies** and remove:

- Encrypting File System
- Secure Email

Certifying a client key pair for Entrust Proxy for Microsoft CA

In a temporary directory under the Microsoft Proxy Server, run the following commands to generate and certify a key pair.

Generate a key pair and keystore

```
keytool -genkey -noprompt -alias <alias> -dname <dname> -keyalg <keyalg> -keysize <keysize> -keystore <keystore> -storepass <storepass> -keypass <keypass>
```

Generate a Certificate Signing Request (CSR)

```
keytool -certreq -alias <alias> -file <file> -keystore <keystore> -storepass <storepass>
```

Submit CSR to Microsoft CA and get certificate chain

```
certreq.exe -f -attrib "<attrib>" -config "<host>\<CA>" <file> CertChainFileOut
<CertChainFileOut>
```

i Depending on the Microsoft CA setup, you may need to manually approve the request and retrieve the certificate.

See the following table for a description of the main parameters.

Option	Value
<alias>	A label for the new keystore
<attrib>	The name of the template described in Creating a client authentication template for Microsoft CA .
<ca>	The CA name assigned to the Microsoft CA in CA Gateway.
<dname>	The distinguished name of the key pair holder
<file>	The path of the generated certificate request
<host>	The Microsoft CA's hostname
<keyalg>	The algorithm for generating the key pair
<keypass>	The password for protecting the key pair withing the keystore
<keysize>	The size of the generating keys
<keystore>	The path of the generated keystore
<storepass >	The password for protecting the keystore

For example:

Generate a key pair and keystore

```
keytool -genkey -noprompt -alias mscaproxyclient -dname "cn=mscaproxy client" -keyalg
RSA -keysize 2048 -keystore mscaproxyclient.jks -storepass ***** -keypass *****
```

Generate a Certificate Signing Request (CSR)

```
keytool -certreq -alias mscaproxyclient -file mscaproxyclient.csr -keystore mscaproxyclient.jks -storepass *****
```

Submit CSR to Microsoft CA and get certificate chain

```
certreq.exe -f -attrib "CertificateTemplate:ClientAuthentication" -config "<HOST>\<CA>" mscaproxyclient.csr CertChainFileOut mscaproxyclient.p7b
```

Importing a key pair and a certificate in the client keystore for Entrust Proxy for Microsoft CA

Import into a keystore the certificate and the keys generated when [Certifying a client key pair for Entrust Proxy for Microsoft CA](#).

```
keytool -import -noprompt -alias <alias> -file <file> -keystore <keystore> -storepass <storepass>
```

Import the keys and the certificate into the keystore.

Option	Value
<alias>	The keystore <alias> selected when Certifying a client key pair for Entrust Proxy for Microsoft CA
<file>	The <CertChainFileOut> file path of the certificate chain obtained when Certifying a client key pair for Entrust Proxy for Microsoft CA
<keystore>	The <keystore> file path of the keystore generated when Certifying a client key pair for Entrust Proxy for Microsoft CA
<storepass>	The <storepass> keystore password selected when Certifying a client key pair for Entrust Proxy for Microsoft CA

```
keytool -import -noprompt -alias mscaproxyclient -file mscaproxyclient.p7b -keystore mscaproxyclient.jks -storepass *****
```

Deleting temporary files of the client keystore Entrust Proxy for Microsoft CA


Delete the temporary files created in the previous sections. For example:

```
del mscaproxyclient.csr
del mscaproxyclient.p7b
```

Generating a truststore for CA Gateway

- [Creating the SSL directory](#)
- [Importing the CA certificates](#)

You need a truststore containing the CA chain for validating the Entrust Proxy for Microsoft CA's connection during TLS handshake. This is a chain of CA certs which has issued the server certificate of Entrust Proxy for Microsoft CA.

 The following instructions create a Java KeyStore (JKS) with the Java `keytool` command line utility. Consider using a more secure PKCS#12 type instead.

Creating the SSL directory

Create an SSL directory under the Entrust Proxy for Microsoft CA installation. For example:

```
c:\mscaproxy\ssl
```

Importing the CA certificates

In the SSL directory, run the following command to include the certificate of the root CA and all the intermediate CAs. See below for a description of each parameter.

```
keytool -import -noprompt -alias <alias> -file <file> -keystore <keystore> -storepass <storepass>
```

See below for a description of each parameter.

Option	Value
<alias>	The alias of the CA in CA Gateway
<file>	The path of the CA certificate file
<keystore>	The path of the truststore file
<storepass>	The password of the trustore

For example:

```
keytool -import -noprompt -alias myca -file myca.crt -keystore truststore.jks
-storepass ****
```


Generating the server keystore of the Entrust Proxy for Microsoft CA

You need a keystore containing:

- The SSL authentication certificate of the Entrust Proxy for Microsoft CA.
- The private key of the certificate.
- The validation chain of the certificate.

See below the required steps.

- [Generating the keystore](#)
- [Setting the Subject Name](#)
- [Adding the keystore password to the configuration](#)
- [Adding the truststore password to the configuration](#)
- [Restarting Entrust Proxy for Microsoft CA](#)


 The following instructions create a Java KeyStore (JKS) with the Java `keytool` command line utility. Consider using a more secure PKCS#12 type instead.

Generating the keystore

Go to the SSL directory containing the `truststore.jks` file previously generated. For example:

```
c:\mscaproxy\ssl
```

Run the following commands to generate the key.

 The below commands use the default Web Server certificate template. If you need to customize any settings of the Web Server certificate template, use a copy of it.

```
keytool -genkey -noprompt -alias mscaproxy -dname "cn=MS CA proxy server FQDN"
-keyalg RSA -keysize 2048 -keystore mscaproxy.jks -storepass <STOREPASS> -keypass
<KEYPASS>
```

```
keytool -certreq -alias mscaproxy -ext SAN=dns:MS CA proxy server FQDN -file
mscaproxy.csr -keystore mscaproxy.jks -storepass <STOREPASS>
```

```
certreq.exe -f -attrib "CertificateTemplate:WebServer" -config "MS CA host name\CA
name" mscaproxy.csr CertChainFileOut mscaproxy.p7b
```

```
keytool -import -noprompt -alias mscaproxy -file mscaproxy.p7b -keystore  
mscaproxy.jks -storepass <STOREPASS>
```

```
del CertChainFileOut  
del CertChainFileOut.rsp  
del mscaproxy.csr  
del mscaproxy.p7b
```

Where:

- "MS CA proxy server FQDN" is the fully qualified domain name of your Entrust Proxy for Microsoft CA's server.
- <STOREPASS> is the password of the keystore.
- <KEYPASS> is the password of the private key.

Setting the Subject Name

Edit the `application.yml` file of the Entrust Proxy for Microsoft CA installation folder.

```
config\application.yml
```

Uncomment all lines (by removing #) and assign to `subject-dn` the distinguished name set with `-dname` when generating the client keystore. For example:

```
subject-dn: "cn=mscaproxy client"
```

Adding the keystore password to the configuration

Edit the following file.

```
MS CA Proxy Installation\config\key-store-password.scrt
```

Set the following parameter:

```
decrypted=<STOREPASS>
```

Where <STOREPASS> is the password of the keystore described in [Generating the keystore](#).

Adding the truststore password to the configuration

Edit the following file:

```
MS CA Proxy Installation\config\trust-store-password.scrt
```

Set the following parameter.

```
decrypted=<STOREPASS>
```

Where `<STOREPASS>` is the password of the keystore described in [Generating the keystore](#).

Restarting Entrust Proxy for Microsoft CA

If the Entrust Proxy for Microsoft CA is running, execute the following command as an administrator to restart it.

```
MSCAProxy.exe restart
```

Running the Entrust Proxy for Microsoft CA

Administrators can start, stop, and restart the Entrust Proxy for Microsoft CA with the following commands.

- `MSCAProxy.exe start`
- `MSCAProxy.exe stop`
- `MSCAProxy.exe restart`

Once started, you can check the correct execution of the Entrust Proxy for Microsoft CA using a Chrome browser.

To check the execution of the Entrust Proxy for Microsoft CA

1. Run the following command to generate a PKCS#12 from the `mscaproxyclient.jks` keystore.

```
keytool -importkeystore -srckeystore mscaproxyclient.jks -destkeystore  
mscaproxyclient.p12 -srcstoretype JKS -srcstorepass <SRCSTOREPASS>  
-deststoretype PKCS12 -deststorepass <DESTSTOREPASS>
```

2. Import the generated `mscaproxyclient.p12` file into Chrome.
3. Go to:

```
https://<proxyserver>:8443/MSCAProxy/rest/status/ping
```

4. Check the server response. The "MS CA proxy is running" message indicates a correct operation.

Integrating an AWS CA

This section explains how to configure CA Gateway for integrating CAs of the Amazon Web Services (AWS).

- [Installing and configuring the AWS CA plugin](#)
- [Handling certificate events with DynamoDb](#)

Installing and configuring the AWS CA plugin

Entrust distributes each `<VERSION>` version of AWS CA plugin in a file with the following name.

```
cagw-plugin-awsca-<VERSION>.zip
```

This ZIP file contains the following folders:

- lib
- config/edm/mc

See below for how to install and configure the AWS CA plugin.

To install the AWS CAs plugin

1. Extract the contents of the ZIP distribution file.
2. Create a `<HOST_CONFIG>/plugins` folder in the CA Gateway machine. Where `<HOST_CONFIG>` is the folder described in [Creating the configuration and credentials folders](#).
3. In this folder, copy the the JAR files of the `lib` folder.
4. Make the `plugins` directory accessible to the `cagw` user inside the CA Gateway container.

```
sudo chown -R :1339 <HOST_CONFIG>/plugins sudo chmod -R g+rx <HOST_CONFIG>/  
plugins
```

5. When starting the CA Gateway instance, set the `LOADER_PATH` environment variable to the location of the plugins directory within the container.

```
docker run -p 8444:8080 -e LOADER_PATH=/etc/cagw/config/plugins -v  
<HOST_CONFIG>:/etc/cagw/config cagw/api:latest
```

Handling certificate events with DynamoDb

You can handle certificate events with the DynamoDb table hosted in your AWS environment. This method has additional costs but also improved performance: AWS will charge for the DynamoDb service and all of the traffic to and from CA Gateway. With this cost, however, comes increased speed and scalability.

Our testing found that, in the worst case, DynamoDb performs evenly with the AWS Audit Report method. Still, as the number of events in the CA grows, DynamoDb performs significantly faster than the AWS Audit Report method. The improvement is particularly evident when the number of requested events is small compared to the total number of events in the CA. DynamoDb also uses no additional memory in CA Gateway.

When a certificate is issued or revoked by an ACM Private CA:

1. Amazon CloudTrail logs an `IssueCertificate` or `RevokeCertificate` event.
2. An Amazon EventBridge rule triggers an AWS Lambda function.
3. The lambda function receives the event data and extracts the relevant details to create a record.
4. The lambda function saves the record in the DynamoDb table where CA Gateway queries certificate events.

See the following sections for how to configure this method in AWS and CA Gateway.

- [Creating the AWS DynamoDb table](#)
- [Creating the AWS Lambda function](#)
- [Configuring the AWS Lambda function](#)
- [Limiting AWS DynamoDB permissions \(optional\)](#)
- [Limiting AWS Private Certificate Manager permissions \(optional\)](#)
- [Creating the AWS EventBridge rule](#)

- [Configuring the AWS DynamoDB table in CA Gateway](#)

Creating the AWS DynamoDb table

Create an AWS DynamoDb table for storing the certificate events.

To create an AWS DynamoDb table

1. Search for `DynamoDB` in the search box at the top of the AWS Management Console.
2. Navigate to the main page of the DynamoDB service.



3. Select **Create table**.
4. Provide a table name. You will later add the selected name in Creating the AWS Lambda function and Configuring the AWS DynamoDB table in CA Gateway. The Lambda script provided by Entrust uses the `certificate-events` default table name.
5. Under **Partition key**, enter `certificate_authority_arn` for the primary key and select **String** for the type (default value).
6. Check **Add sort key**.
7. In the resulting text box, enter `time` and select **String** for the type (default value).
8. In the **Table settings** section, do not uncheck **Use default settings** (checked by default) unless you know what you're doing and would like to change these settings for your own needs.
9. Review the configured settings:
 - The partition key must be `certificate_authority_arn`.
 - The sort key must be `time`.
 - The table name is up to you.
10. Click **Create** to create the table.

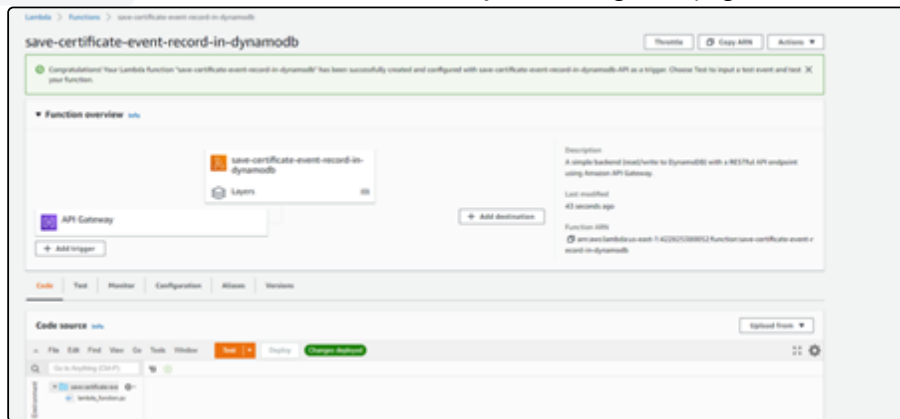
Creating the AWS Lambda function

Create a Lambda function for storing certificate event records in the AWS DynamoDb table.

To create the Lambda function

1. Search for `Lambda` in the search box at the top of the AWS Management Console.
2. Navigate to the main page of the Lambda service.
3. Select **Create function**.
4. Select **Use a blueprint**.
5. Search for `microservice-http-endpoint-python` in the **Blueprints** search box.
6. Select the `microservice-http-endpoint-python` blueprint.

7. Click **Configure**.
8. In the resulting page, provide a function name.
9. In **Execution role**, select **Create a new role from AWS policy templates**.
10. Provide a name for the role.
11. Under **Policy templates**, select **Simple microservice permissions: DynamoDB**.
12. In the **API Gateway trigger** section under **API**, select **Create an API**.
13. For **API type**, select **REST API**.
14. For security, select **IAM**.
15. Click **Create function**.
16. Review the performed steps. For example, after selecting the `save-certificate-event-record-in-dynamodb` name for the Lambda function, you should go to a page like the following.



Configuring the AWS Lambda function

Configure the Lambda function previously created in [Creating the AWS Lambda function](#).

To configure the Lambda function

1. Copy the contents of the `dynamo-db-lambda.py` file provided with the AWS CA Gateway plugin artifact.
2. Paste the contents under the **Code source** tab to replace the `dynamo-db-lambda.py` contents.
3. Assign to the `TABLE_NAME` variable the name of the DynamoDB table created earlier. For example:

```
TABLE_NAME = "certificate-events"
```

⊗ Be careful if you choose to edit the Lambda code in any other way. The Lambda code is directly responsible for storing certificate event records in the DynamoDB table, so the correctness of CA Gateway certificate events API depends on the correctness of this Lambda code.

4. Click **Deploy** above the code file.
5. Click the **Configuration** tab.
6. On the left pane, select **Permissions**.
7. In the **Execution role** section, click the role associated with this Lambda function. You should see a summary of the Lambda function role in a new tab.
8. In the new tab, click **Attach policies** under **Permissions**.
9. On the resulting page, search for `AWSCertificateManagerPrivateCAReadOnly`.
10. Select the `AWSCertificateManagerPrivateCAReadOnly` policy.

11. Click **Attach policy**. Attaching this policy to the role allows the Lambda function to retrieve certificate content from an AWS private CA.
12. Verify that your role has all of the necessary permissions. The following permissions will suffice.
 - Certificate Manager Private Certificate Authority: List, Read
 - DynamoDB: Read, Write
 - CloudWatch Logs: Write

Limiting AWS DynamoDB permissions (optional)

You may wish to limit the DynamoDB permissions to saving certificate events in the AWS DynamoDb table.

To limit the DynamoDB permissions

1. In the **AWSLambdaMicroserviceExecutionRole** section containing the DynamoDB permissions, click **Edit policy**.
2. In the JSON tab, replace the value in the Resource field with the ARN of the DynamoDb table previously created in [Creating the AWS DynamoDb table](#). To get this ARN in the DynamoDB service console, go to the **Table details** section and click the **Overview** tab.

Limiting AWS Private Certificate Manager permissions (optional)

You may wish to limit Certificate Manager Private CA permissions to the CA(s) the Lambda code will access. As the policy already attached to the Lambda role is a pre-packaged AWS-managed policy, you won't be able to edit it. Instead, you'll have to create a policy.

To limit the Private Certificate Manager permissions

1. Remove the **AWSCertificateManagerPrivateCAReadOnly** policy by clicking the **X** to the right of it. Confirm this action when prompted.
2. Click **Attach policies**.
3. Click **Create policy**.
4. Create the policy either via the **Visual editor** or the **JSON** editor.

Creating the AWS EventBridge rule

Create an AWS EventBridge rule for invoking the Lambda function.

To create an EventBridge rule

1. Search for **Amazon EventBridge** in the search box at the top of the AWS Management Console.
2. Navigate to the main page of the Amazon EventBridge service.
3. Select **Create rule**.
4. Provide a name and description for the rule.
5. In the **Define pattern** section, select **Event pattern**.
6. In **Event matching pattern**, select **Pre-defined pattern by service**.
7. In **Service provider** select **AWS**.
8. In **Service name**, select **Certificate Manager Private CA**.
9. In **Event type** select **AWS API Call via CloudTrail**.
10. Select **Specific operation(s)** instead of the default **Any operation**.
11. Add the **IssueCertificate** and **RevokeCertificate** operations.
12. Review the configuration. The **Event pattern** should look like this.

```
{
  "source": ["aws.acm-pca"],
```

```

"detail-type": ["AWS API Call via CloudTrail"],
"detail": {
  "eventSource": ["acm-pca.amazonaws.com"],
  "eventName": ["IssueCertificate", "RevokeCertificate"]
}

```

13. In the **Select event bus** section, select **AWS default event bus** (selected by default).
14. In the **Select targets** section under the first **Target**, select **Lambda function** (selected by default).
15. Under **Function** select the Lambda function previously created in Creating the AWS Lambda function.
16. Click **Create**.

Configuring the AWS DynamoDB table in CA Gateway

Edit the CA Gateway `application.yml` file and configure the following [AWS CA properties](#).

Setting	Value
certificate-events-storage-method	"DynamoDb"
dynamodb-table-name	The table name selected in Creating the AWS DynamoDb table .

For example:

```

AWSWITHDYNAMODB:
  name: "AWS Private Root CA"
  issuer-dn: "L=Dallas, CN=WY Root CA 1, ST=Texas, OU=Terraforming, O=Weylan-
Yutani Corporation, C=US"
  connector-name: com.entrust.awsca
  properties:
    aws-region: "us-east-1"
    aws-ca-arn: "arn:aws:acm-pca:us-east-1:422825380052:certificate-authority/
2b33862c-e9f0-490d-9a3a-ad74476f7bca"
    aws-ca-s3-crl-arn: ""
    aws-ca-audit-report-s3-bucket-name: "cagw-audit-report"
    aws-api-url: "https://acm-pca.us-east-1.amazonaws.com:443/"
    aws-user-login-url: "https://edc-admin-services-dev.signin.aws.amazon.com/
console"
    aws-user-arn: "arn:aws:iam::422825380052:user/cagw-ca-user"
    aws-user-access-key-id: "ABCDEFG1234567890123"
    aws-user-secret-access-key: "ABCDEFG123456789012345678901234567890123"
    certificate-events-storage-method: "DynamoDb"
    dynamodb-table-name: "certificate-events"

```

Integrating an EJBCA

This section explains how to configure CA Gateway for integrating Enterprise Java Beans Certificate Authorities (EJBAs).

- [Configuring and issuing the EJBCA client certificate](#)

10. If prompted, authorize the operation. The operation may require more than one authorization. See the Entrust Certificate Authority documentation for details.

Creating a role for the administrator profile

To connect and perform operations with an Entrust Certificate Authority, CA Gateway requires an administrator profile issued by the Entrust Certificate Authority. This profile must have a role with the following permissions.

Permission category	Permissions
Certificates	Administer at least one certificate category. Currently, CA Gateway supports only Enterprise certificate types.
Certificate Types	Administer at least one certificate type.
Groups	<ul style="list-style-type: none">• View• Administer at least one group
License Information	View
Roles	<ul style="list-style-type: none">• View• Administer at least one role.
Searchbases	<ul style="list-style-type: none">• View• Administer at least one searchbase.
Security Policy	<ul style="list-style-type: none">• View Security Policy• Export Certificate Specification• Export User Templates• Force CRLs• View User Policy
User Templates	Administer at least one template

Permission category	Permissions
User - General	<ul style="list-style-type: none">• View• Add• Reactivate• Deactivate/Remove• Change DN• Modify properties• Revoke certificates• Update key pairs• Set for key recovery• Cancel key recovery• Modify key update options• View activation code• Reissue activation code
User - Advanced	<ul style="list-style-type: none">• Change user's role• Perform PKIX requests• Create user profile



Refer to the Entrust Certificate Authority Administration documentation for more details on role configuration.

To create a new role for the administrator profile

1. Log in to Entrust Certificate Authority Administration for the Entrust Certificate Authority.
2. In the tree view, expand **Security Policy > Roles**.
3. Select **Policies > Roles > New** to create a new role. Alternatively, you can copy the **Administrator** role because this role includes most of the permissions required for the new role.
 - a. Select **Administrator**.
 - b. Select **Policies > Roles > Selected Role > Copy**. A copy of the role appears at the bottom of the list of roles in the tree view, and the new role's properties appear in the right pane.
4. Click the **Role** tab.
 - a. Into the **Unique name** field, enter `CAGW Admin Role`.
 - b. In the **Authorizations** field, enter 1.
 - c. In the **User Policy** drop-down list, select **CAGW Admin Policy**. This is the client policy you created earlier.
 - d. Unselect the **End User** check box. This check box should already be deselected.
5. Click the **Permissions** tab.
6. Configure the permissions documented in the above table and click **Apply**.
7. If prompted, authorize the operation. As explained in the Entrust Certificate Authority Administration documentation, the operation may require more than one authorization.
8. A **Permission Dependencies** pop-up dialog may list additional permissions required for the role to function properly. Add these missing permissions to the role.

Creating a user entry for the administrator profile

Create a user entry in Entrust Certificate Authority for the administrator profile.

To create a user entry for the administrator profile

1. Log in to the Entrust Certificate Authority administration portal.
2. Select **Users** > **New User** to display the **New User** dialog.
3. Select the following tabs to configure the corresponding fields.
 - [Naming](#)
 - [General](#)
 - [Certificate Info](#)
 - [Key Update Options](#)
4. Click **OK**.
5. If prompted, authorize the operation. The operation may require more than one authorization. See the Entrust Certificate Authority documentation for details.
6. Copy the reference number and authorization code required to create the administrator profile. You will require them later to create and activate the user's Entrust digital ID. See the Entrust Certificate Authority documentation for more details about how the Registration number and Authorization codes are used.

Naming

Configure the following fields under this tab.

Field	Value
Type	Select a user type.
User fields	Enter a value for all configuration fields of the selected user type.
Add to	Select a searchbase for the user – for example, select CA Domain Searchbase to add the user entry to the default searchbase.

General

Configure the following fields under this tab.

Field	Value
User role	Select the role described in Creating a role for the administrator profile .
User group(s)	Assign the user to one or more groups.

Certificate Info

Configure the following fields under this tab.

2. Click the **+** icon.
3. Complete the **Add New Person** fields. Specifically:
 - Provide a secret.
 - Select the same email address you will later use for the administrator.

Issuing the client certificate

Issue a client certificate to authenticate in the Sectigo API.

1. Open the newly created enrollment form in a new browser tab.
2. Issue a certificate using the same secret and email address configured for the newly created person.
3. Download the PKCS #12 file.

Creating an administrator

Create a user with administrator privileges in Sectigo.

1. Navigate to **Settings > Admins**.
2. Click the **+** icon.
3. In the **Add Admin Type** dialog, select **Standard**.
4. Complete the **Add New Admin** fields. Specifically:
5. Use the same email address configured for the newly created person.
6. In the **Authentication** tab, select the newly issued certificate.


```
  jsse:
    debug: false
    level: ssl
server:
  port: 8080
  servlet:
    context-path: /cagw
  ssl:
    enabled: true
    insecure-mode: false
    protocol: TLS
    key-alias: cagw
    key-store: /etc/cagw/config/keystore.p12
    key-store-password: Password
    key-store-type: pkcs12
    trust-store: /etc/cagw/config/truststore.p12
    trust-store-password: Password
    trust-store-type: pkcs12
    client-auth: need
management:
  server:
    port: 9090
    ssl:
      enabled: true
      protocol: TLS
      key-alias: cagw
      key-store: /etc/cagw/config/keystore.p12
      key-store-password: Password
      key-store-type: pkcs12
      trust-store: /etc/cagw/config/truststore.p12
      trust-store-password: Password
      trust-store-type: pkcs12
      client-auth: want
  endpoint:
    health:
      group:
        custom:
          include: diskSpace,ping
          show-components: always
          show-details: always
  endpoints:
    web:
      base-path: /cagw/management/actuator
      exposure:
        include: health,prometheus
cagw:
  connector-filters:
    filter-lists: {}
  cert-event-tracking: {}
  authorities:
    managed-cas: {}
  tenants: []
  integrators: []
```

```

clients: []
deploy:
  enable: {}
caches:
  subject-dn-cache: {}
  profile-cache: {}
  profiles-cache: {}
  subject-builder-cache: {}
  ca-capabilities-cache: {}
  ca-information-cache: {}
  requested-properties-cache: {}
license:
  signed-path: /etc/cagw/config/license.lic

```

 See [Running the Profile Creation Utility](#) for how to create a basic configuration.

cagw

Under this section, define the following application settings.

- [authorities](#)
- [caches](#)
- [ca-polling-initial-delay](#)
- [ca-polling-interval](#)
- [cert-event-tracking](#)
- [clients](#)
- [cmp](#)
- [connector-filters](#)
- [integrators](#)
- [license](#)
- [tenants](#)
- [tual.properties](#)

authorities

Under this section, define the following authority settings.

- [halt-for-error](#)
- [managed-cas.<ca>](#)
- [ssl](#)

halt-for-error

The performed action when finding an invalid CA configuration.

Value	Action
true	Halt CA Gateway.

min-rsa

The minimum key size for RSA keys, as a number of bytes.

reject

The performed action when the CA key does not meet the `min-ecc` or `min-rsa` size requirements.

Value	Action
off	Nothing
log	Log the key size
block	Reject the key

name

A friendly name for the CA.

Mandatory: Yes

profiles.<profile>

Under this field, add the following settings for each profile with the `<profile>` identifier.

- [copy-cn-to-san](#)
- [key-size \(profile\)](#)
- [name \(profile\)](#)
- [CA-specific profile settings](#)
- [san-requirements](#)
- [subject-builder-config](#)
- [subject-variable-requirements](#)

For example:

```

profiles:
  CA-1003-PROF-1001:
    name: default profile
    subject-variable-requirements:
      - required: true
        name: CN
        description: common name
    subject-builder-config:
      subject-builder-name: >-
        com.entrust.adminservices.cagw.common.subjects.TemplateSubjectBuilder
    properties:
      template: cn=<First Name> <Last Name>,ou=CA01,o=pki,dc=hooli,dc=com
    properties:
      cert-type: ent-twokeypair
      cert-definition: Verification
  
```


i See [Allowing server-generated verification keys](#) for how to enable PKCS #12 generation on Entrust Certificate Authority.

filter-list

The identifier of the filter list for issuing trusted certificates.

Mandatory: No.

properties

Under `profiles.<profile>`, add a `properties` section with the following Entrust Certificate Authority-specific settings.

- `ca-variable-<i>`
- `cert-definition`
- `cert-type`
- `create-ldap-entry`
- `directory-mode`
- `user-role`
- `user-role-override`
- `user-type`

`ca-variable-<i>`

When `create-ldap-entry` is `false`, use the following set of parameters to define each user variable CA Gateway supplies to Entrust Authority Entrust Certificate Authority.

- `ca-variable-<i>-type`
- `ca-variable-<i>-name`
- `ca-variable-<i>-value`
- `ca-variable-<i>-in-dn`

Where `<i>` is an integer value number starting at 0. For example:

```
ca-variable-0-type: UserType
ca-variable-0-name: cn
ca-variable-0-value: <firstname>
ca-variable-0-in-dn: true
ca-variable-1-type: UserType
ca-variable-1-name: sn
ca-variable-1-value: <lastname>
ca-variable-1-in-dn: false
ca-variable-2-type: Custom
ca-variable-2-name: email
ca-variable-2-value: <email>
ca-variable-2-in-dn: false
```

Mandatory: Only when `create-ldap-entry` is `false`.

`ca-variable-<i>-type`

The type of variable. Supported values are:

- CertType
- Custom
- UserType
- Variable

`ca-variable-<i>-name`

The name of the Managed CA variable. See the table below for examples.

name	type	Value
cn	UserType	Common Name attribute in the directory.
sn	UserType	Serial Number attribute in the directory.
email	Custom	The certificate subject's email. Entrust Authority Entrust Certificate Authority and clients like UMS understand the <code>email</code> variable, so no additional configuration is necessary.

`ca-variable-<i>-value`

The value of the Managed CA variable. This value must include one or more substrings surrounded by angle brackets. For example:

- `<firstname>` for the user's first name.
- `<lastname>` for the user's last name.

Client applications will provide the actual values during enrollment.

`ca-variable-<i>-in-dn`

`true` to include the variable value in the user's Distinguished Name (DN), `false` to exclude the variable value from the user DN.

`cert-definition`

The certificate definition for processing enrollment requests under the certificate profile. For example:

- Verification
- Dual usage
- Encryption

This certificate definition must have an assigned certificate definition policy. Otherwise, enrollments will fail.

Mandatory: Yes.

`cert-type`

The Entrust Certificate Authority certificate type to use when processing an enrollment request under the certificate profile. For example:

- `ent_twokeypair`
- `ent_default`

The administrator EPF for the Managed CA must have permission to administer this certificate type.

cert-type	Description
PV_INTRA_ECCP256	Intranet SSL (ECC P-256) certificate
PV_CLOUD	Cloud SSL certificate
PV_CLOUD_ECC2	Cloud SSL (ECC) certificate

Mandatory: Yes

base-option

The trust model and validation rules for the requested certificate. See the table below for the supported values for each certificate type.

base-option	Trust model	Supported certificate types
gip	GlobalSign Internet PKI for publicly trusted certificate profiles	PEV_SHA2, PV_SHA2, PV_CLOUD, PV_CLOUD_ECC2
private	GlobalSign Internet PKI for non-public or internal certificates	PV_INTRA_SKIP, PV_INTRA_SHA2, PV_INTRA_SKIP_SHA2, PV_INTRA_ECCP256
wildcard	A publicly trusted certificate that contains a wildcard SAN, such as <code>*.example.com</code>	PEV_SHA2, PV_SHA2, PV_CLOUD, PV_CLOUD_ECC2

Mandatory: No

key-size

The key size for the issued certificates. Supported values are:

- 2048
- 4096

Mandatory: only for the following certificate types.

- PV_INTRA_SKIP_SHA2
- PV_INTRA_ECCP256

domain-id

The identifier of the managed SSL domain record in the GlobalSign MSSL account.

Mandatory: Yes

validity-months

The default validity period for issued certificates, as a number of months.

Mandatory: Yes.


ra-enroll-key-store-provider

The security provider of the key store. When creating RA enrollment agent credentials in a Key Store file, supported values are the following.

Value	Security provider
SunJSSE	PKCS#12 and PFX
SUN	JKS
SunJCE	JCEKS

When creating RA enrollment agent credentials in PKCS#11 HSM, supported values are the following.

Value	Security provider
SunPKCS11	nCipher
LunaProvider	Luna

 CA Gateway tries loading the key store with any available security provider when this value is omitted or incorrect.

Mandatory: Yes.

ra-enroll-key-store-provider-config

The path of the SunPKCS11 configuration file described in the Thales Luna integration guide.

Mandatory: Yes.

ra-enroll-key-store-type

The type of key store. See the table below for the supported values.

Credential type	Supported key store types
RA enrollment agent credentials in a Key Store file	pkcs12
	pfx
	jks
	jceks


Credential type	Supported key store types
RA enrollment agent credentials in PKCS#11 HSM	pkcs11

Mandatory: Yes.

Sectigo CA profile settings

When integrating a Sectigo CA, configure the Sectigo-specific profile settings as follows.

- [Configuring the static Sectigo CA profile settings](#)
- [Enabling Sectigo CA dynamic profile generation](#)

 Sectigo integration only supports TLS profiles. S/MIME profiles are not supported.

Configuring the static Sectigo CA profile settings

Under `profiles.<profile>`, add a static configuration like the following.

```
profiles:
  sectigo-profile-1:
    name: "static-profileA-local"
    properties:
      cert-type: 60515
```

Where `cert-type` is the identifier of a profile configured in Sectigo.

Enabling Sectigo CA dynamic profile generation

After [Configuring the static Sectigo CA profile settings](#):

1. Set the `enable-ca-profile-sync` flag to `true`.
2. Make an API call to the `/v1/certificate-authorities/{caId}/profiles` endpoint.

CA Gateway will return the static configuration merged with profiles configured at Sectigo. For example:


```
{
  "message": {
    "message": "Profiles retrieved successfully.",
    "details": []
  },
  "profiles": [
    {
      "id": "sectigo-profile-1",
      "name": "static-profileA-local",
      "properties": {
        "cert_type": "60515"
      },
      "protocols": [],
      "requestedProperties": [],
      "subjectAltNameRequirements": []
    }
  ]
}
```

```
    "subjectVariableRequirements": []
  },
  {
    "id": "60515",
    "name": "profileA",
    "properties": {
      "cert_type": "60515",
      "description": "",
      "key_type": "{\"EC\": [\"P-256\", \"P-384\", \"P-521\"], \"RSA\": [\"2048\", \"3072\", \"4096\", \"8192\"]}",
      "terms": "[30]",
      "use-secondary-org-name": "false"
    },
    "protocols": [],
    "requestedProperties": [],
    "subjectAltNameRequirements": [],
    "subjectVariableRequirements": []
  },
  {
    "id": "63999",
    "name": "Sectigo Enterprise Pro - Multi-Domain (OV)",
    "properties": {
      "cert_type": "63999",
      "description": "",
      "key_type": "{\"EC\": [\"P-256\", \"P-384\"], \"RSA\": [\"2048\", \"4096\", \"8192\"]}",
      "terms": "[47, 200, 365, 397]",
      "use-secondary-org-name": "false"
    },
    "protocols": [],
    "requestedProperties": [],
    "subjectAltNameRequirements": [],
    "subjectVariableRequirements": []
  },
  {
    "id": "64442",
    "name": "test",
    "properties": {
      "cert_type": "64442",
      "description": "asdfsadf",
      "key_type": "{\"EC\": [\"P-256\", \"P-384\", \"P-521\"], \"RSA\": [\"2048\", \"3072\", \"4096\", \"8192\"]}",
      "terms": "[30]",
      "use-secondary-org-name": "false"
    },
    "protocols": [],
    "requestedProperties": [],
    "subjectAltNameRequirements": [],
    "subjectVariableRequirements": []
  }
],
"type": "ProfilesResponse"
```

```
}
```


See below for the main parameters in this configuration.

- [key-types](#)
- [terms](#)

 In this API response, some parameter names include underscores for backwards compatibility, and key type values are escaped because they contain embedded JSON code.

`key-types`

A list of supported key types for the issued certificate.

 This parameter is dynamically provisioned by Sectigo when enabling the [enable-ca-profile-sync](#) flag, so you do not need to set it manually.

For example

```
"key_types": "{\"EC\":[\"P-256\",\"P-384\"],\"RSA\":[\"2048\",\"4096\",\"8192\"]}"
```


As described in the table below, the requested validity (if any) must be included on this list.

Request type	Requested key type	Key type for the issued certificate
PKCS #10	Included in the <code>key-types</code> list	As selected in the request
	Not included in the <code>key-types</code> list	Error
PKCS #12	—	The first in the <code>key-types</code> list, if key-type is not available, uses default.

Mandatory: No, this optional value defaults to a list containing a single RSA 2048 key type.

`terms`

A list of supported certificate validity periods, in days.

 This parameter is dynamically provisioned by Sectigo when enabling the [enable-ca-profile-sync](#) flag, so you do not need to set it manually.

For example

```
terms": "[47, 200, 365, 397]"
```

As described in the table below, the requested validity (if any) must be included on this list.

Request type	Requested validity	Validity of the issued certificate
PKCS #10	Included in the <code>terms</code> list	As selected in the request
	Not included in the <code>terms</code> list	Error
	None	The first in the <code>term</code> list, if term list not available, uses default.
PKCS #12	—	The first in the <code>term</code> list, if term list not available, uses default.

Mandatory: No, this optional value defaults to a list containing a single 30-day period.

san-requirements

Under this section, define the requirements of the Subject Alternative Name (SubjectAltName) expected during enrollment requests.

- `type`
- `required`

For example:

```
san-requirements:
  - type: rfc822Name
    required: false
  - type: otherName
    required: false
```

type

The type for the Subject Alternative Name, as defined by [RFC 5280](#). For example:

- `rfc822Name`
- `dNSName`

Mandatory: Yes.

required

`true` if the Subject Alternative Name is required, `false` if the Subject Alternative Name is optional.

Mandatory: Yes.

subject-builder-config

Under this section, define the Subject Builders for constructing the subject DN for the enrollment request.

- `subject-builder-name`
- `properties`

subject-builder-name

The class name of the subject builder.

- [com.entrust.adminservices.cagw.common.subjects.BasicSubjectBuilder](#)
- [com.entrust.adminservices.cagw.common.subjects.SubAltNameSubjectBuilder](#)
- [com.entrust.adminservices.cagw.common.subjects.TemplateSubjectBuilder](#)

⚠ CA Gateway ignores this setting, and the subject name in the CSR prevails when an enrollment request for a Sectigo CA specifies the PEM, PEM_WITHOUT_LINE_BREAKS, or X509 formats.

Mandatory: No. This optional parameter defaults to [com.entrust.adminservices.cagw.common.subjects.BasicSubjectBuilder](#).

`com.entrust.adminservices.cagw.common.subjects.BasicSubjectBuilder`

Select this subject builder to append all the supplied subject variables together in the order of arrival.

Sample BasicSubjectBuilder

```
- name: "Use BasicSubjectBuilder"
  unique-id: "CA-1003-PROF-1001"
  subject-builder-config:
    subject-builder-name:
      "com.entrust.adminservices.cagw.common.subjects.BasicSubjectBuilder"
```

Sample subject variables

```
"subjectVariables" : [
  {
    "type" : "cn",
    "value" : "test"
  },
  {
    "type" : "o",
    "value" : "pki"
  }
]
```

Subject DN generated by the sample builder when parsing the sample variables

```
cn=test,o=pki
```

`com.entrust.adminservices.cagw.common.subjects.SubAltNameSubjectBuilder`

Select this subject builder to construct the Subject DN from the Subject Alternative Name provided in the request or CSR. Specifically, this builder:

Example: building the Common Name from Subject Variables

To build the final DN, the following template expects an enrollment request with subject variables for "First Name" and "Last Name".

i When subject variables are sent, any fields in the template are considered required and must be supplied in the request as subject variables.

```
- name: "Use TemplateSubjectBuilder"
  unique-id: "CA-1003-PROF-1002"
  subject-variable-requirements:
    - name: First Name
      description: "First Name"
      required: true
    - name: Last Name
      description: "Last Name"
      required: true
  subject-builder-config:
    subject-builder-name:
      "com.entrust.adminservices.cagw.common.subjects.TemplateSubjectBuilder"
    properties:
      template: "cn=<First Name> <Last Name>, ou=CA01,o=pki,dc=test,dc=com"
```

For example, when receiving the following request values.

```
"subjectVariables" : [
  {
    "type" : "First Name",
    "value" : "PKI"
  },
  {
    "type" : "Last Name",
    "value" : "Test"
  }
]
```

The template builds the following Distinguished Name.

```
cn=PKI Test,ou=CA01,o=pki,dc=test,dc=com
```

Example: building the Common Name when no Subject Variables are provided

To build the final DN when no Subject Variables are provided, the following template parses the CSR for common name.

i When no subject variables are sent, this configuration will scrape the CN and UID from the CSR. No other fields are supported.

- CA Gateway only talks to the Entrust Proxy for Microsoft CA.
- The Entrust Proxy for Microsoft CA is on the CA's same domain and talks to the CA.


For example:

```
ca-host: msca.abccorp.dev.entrust.com
ca-name: abccorpsub
ldap-port: 389
ldap-host: dc.abccorp.dev.entrust.com
```

Mandatory: Yes.

ldap-port

The port number for LDAP connections with Microsoft Active Directory (for LDAPS connections, configure `ldaps-port` instead).


 The port is anonymously bound. The Microsoft CA proxy connects to Active Directory to get certificate template information.

This value is typically 389, the well-known port for LDAP.

Mandatory: When not configuring `ldaps-port`.

ldaps-port

The port number for LDAPS connections with Microsoft Active Directory (for LDAP connections, configure `ldap-port` instead).


 The port is anonymously bound. The Microsoft CA proxy connects to Active Directory to get certificate template information.

This value is typically 636, the well-known port for LDAPS.

Mandatory: When not configuring `ldap-port`.

proxy-host-name

The hostname of the proxy for accessing the CA server.

 The proxy configured using this parameter is part of your corporate infrastructure; it is not an Entrust product.

Mandatory: Only when traffic to the CA server passes through a proxy.

proxy-password

The password for authenticating in the server proxy.

Mandatory: Only when the proxy requires authentication.

proxy-port

The port for accessing the proxy.

Mandatory: Only when traffic to the CA server passes through a proxy.

proxy-ssl

Under this section, configure the following authentication settings.

Parameter	Description	Mandatory
client-cert-key-alias	The alias of the CA Gateway client key	✓
client-cert-key-store	The filename of the CA Gateway client JKS	✓
client-cert-key-store-password	The password of the CA Gateway client JKS	✓
client-cert-key-store-type	Set this parameter to <code>JKS</code>	✓
ssl-trust-store	The path of the CA Gateway trust store (See Configuring for how to reference file paths)	✓
ssl-trust-store-password	The password of the CA Gateway trust store.	✓
ssl-trust-store-type	The type of CA Gateway trust store. Supported values are <code>JKS</code> and <code>PKCS12</code> .	✓

proxy-username


The username for authenticating in the CA server proxy.

Mandatory: Only when the proxy requires authentication.

Sectigo CA Properties

Under [properties](#), configure the following Sectigo CA-specific settings.

- [Sectigo settings](#)
- [Authentication settings](#)
- [SSL settings](#)
- [Enrollment settings](#)
- [Proxy settings](#)

 See [Configuring](#) for how to reference file paths.

Sectigo settings

Configure the following mandatory Sectigo settings.

Setting	Value
customer-uri	The customer identifier provided by Sectigo
org-id	The organization identifier provided by Sectigo
url	The URL of the Sectigo API

Authentication settings

CA Gateway supports the following modes to authenticate in the Sectigo API.

- Password (recommended)
- Key store
- API key (future releases)

See below for the settings each mode requires.

Setting	Value	Password	Key store
login	A Sectigo login name for a user with the privileges described in Setting Sectigo permissions for API login	✓	✓
login-password	The password of the selected Sectigo login name	✓	
client-cert-key-store	The path of the client trust store described in Creating a Sectigo client key store		✓
client-cert-key-alias	The alias of the client key in the client trust store		✓
client-cert-key-store-password	The password of the client trust store		✓
client-cert-key-store-type	The type of client trust store. Supported values are <code>JKS</code> and <code>PKCS12</code> .		✓

SSL settings

Configure the following mandatory SSL settings to connect with the Sectigo API.

Setting	Value
ssl-trust-store	The path of the trust store described in Creating the Sectigo SSL credentials trust store
ssl-trust-store-password	The password of the trust store
ssl-truststore-type	The type of CA Gateway trust store. Supported values are <code>JKS</code> and <code>PKCS12</code> .

Enrollment settings

The following settings control the enrollment requests.

Key	Value	Default
enroll-back-off-timer	The starting back-off period for certificate retrieval	2 sec
enroll-max-back-off-timer	The maximum back-off period before the next certificate retrieval attempt	32 sec
enroll-max-attempts	The maximum number of certificate retrieval attempts	5

After submitting an enrollment, CA Gateway waits for the following period.

```
min(enroll-back-off-timer^attempt, enroll-max-back-off-timer)
```

Where the `attempt` value:

1. Starts at 1 on the first enrollment attempt.
2. Is increased by 1 after each retrieval attempt, until reaching the `enroll-max-attempts` value.

CA Gateway responds with the following HTTP codes to the client enrollment requests.

Code	Description
HTTP 200	The certificate has been retrieved on time
HTTP 202	The request has been processed, but CA Gateway has exceeded the <code>enroll-max-attempts</code> period
HTTP 404	Any other failure

When receiving a HTTP 202 response, you can:

1. Look up the certificate using the `{caId}` Certificate Authority identifier and the `{dn}` Distinguished Name.


```
/v1/certificate-authorities/{caId}/subjects/{dn}
```

2. Ascertain the serial number from the response.
3. Look up the certificate using the `{sn}` serial number.

```
/v1/certificate-authorities/{caId}/certificates/{sn}
```

Proxy settings

Configure the following settings if traffic to the CA server passes through a proxy.


 The proxy configured using these settings is part of your corporate infrastructure; it is not an Entrust product.

setting	Value
proxy-host-name	The hostname of the proxy for accessing the CA server.
proxy-port	The port for accessing the proxy
proxy-username	The username for authenticating in the proxy (if required)
proxy-password	The password for authenticating in the proxy (if required)

ssl

Configure the following settings when communicating with Entrust Certificate Services using a non-standard certificate. For example, for QA purposes.

- [trust-store](#)
- [trust-store-password](#)

 Do not use these settings in production.

trust-store

The full path of the truststore that contains the CA certificates.

 See [Configuring](#) for how to reference file paths.

Mandatory: Yes.

trust-store-password

The password of the truststore referenced by the [trust-store](#) parameter.

Mandatory: Yes.

caches

See the following table for the cache parameters each CA Gateway API supports and their default value.

API	Main YAML parameter	expire-after-access	expire-after-write	expire-after-unit	expire-after-value	initial-capacity	maximum-size
ca-capabilities-cache	ca-capabilities-cache	false	false			10	500
caInformation	ca-information-cache	false	false		60 (with internal eviction policy)	10	500
CMPv2 enrollment	cmp-transaction-cache	(Not supported)	true	SECONDS	10	2	2
profile	profile-cache	false	false			20	100
profiles	profiles-cache	false	false			20	100
requested Properties	requested-properties-cache	false	false		60 (with internal eviction policy)	20	100
subjectBuilder	subject-builder-cache	false	false			20	100
subjectDN Cache	subject-dn-cache	false	false			2000	10000

For example:

```

caches:
  ca-information-cache:
    enabled: true
    maximum-size: 500
    initial-capacity: 10

```

```

expire-after-write: true
expire-after-value: 60
expire-after-unit: SECONDS
profiles-cache:
  enabled: false
  maximum-size: 100
  initial-capacity: 20
  expire-after-access: true
  expire-after-value: 10
  expire-after-unit: SECONDS

```

See below for a description of each parameter.

- [Enabled parameter](#)
- [Expiry start parameters](#)
- [Cache size parameters](#)

 Setting the following parameters is an expert-level configuration.


Enabled parameter

All API caches support the `enabled` boolean parameter to enable or disable the cache.

Expiry start parameters

If you want to configure the expiry of an API cache, select the event that triggers the expiry countdown.

Parameter	Event
<code>expire-after-access</code>	The most recent event among: cache creation, most recent cache replacement, latest cache access.
<code>expire-after-write</code>	The most recent event among: cache creation, most recent cache replacement.

 These parameters are mutually incompatible. When both are set to `True`, the `expire-after-access` parameter takes precedence.

Expiry period parameters

After selecting the event that triggers the cache expiry, use the following parameters to set the timeout.

Parameter	Value
<code>expire-after-unit</code>	The time unit for the expiry period: SECONDS, MINUTES, HOURS
<code>expire-after-value</code>	The number of time units before removing an entry from the API cache.

CA Gateway will disable cache expiry for the selected API unless:

- Either the `expire-after-access` or the `expire-after-write` parameters are set to `True`.
- Both the `expire-after-unit` and `expire-after-value` parameters are configured.

Cache size parameters

Use the following parameters to set the size of a cache API.

Parameter	Value
<code>initial-capacity</code>	The initial number of entries in the API cache.
<code>maximum-size</code>	The maximum number of entries supported by the API cache.

`ca-polling-initial-delay`

The number of seconds before starting the CA Health Check service. This service polls the CAs configured under `managed-cas.<ca>` with a `ca-polling-interval` periodicity.

Mandatory: No. This optional parameter defaults to 30 seconds.

`ca-polling-interval`

The number of minutes between each execution of the CA Health Check service. This service polls the CAs configured under `managed-cas.<ca>`.

Mandatory: No. This optional parameter defaults to 5 minutes.

`cert-event-tracking`

Under this section, configure the following settings.

- `default-aws-query-backoff-seconds`
- `default-query-page-size`
- `overhead-epoch-adjustment`
- `sm-clock-drift-tolerance`

✘ Setting the following parameters is an expert-level configuration.

`default-aws-query-backoff-seconds`

The number of seconds to back off between queries to AWS certificate authorities.

✘ Setting this parameter is an expert-level configuration.

Mandatory: No. This optional value defaults to 5.

`default-query-page-size`

The default page size to use when making queries, as an integer value equal to or greater than 1.

- [permitted-digest-algorithms](#)
- [permitted-ec-public-key-algorithms](#)
- [permitted-mac-algorithms](#)
- [permitted-signature-algorithms](#)
- [permitted-signature-classes](#)

For example, the following code defines customization rules with the `enable-sha1`, `relaxed` and `password-based` identifiers.

```
cagw:
  cmp:
    specification:
      customizations:
        enable-sha1: # value used to identify this entry, i.e. specificationName
passed in the request
        validation:
          permitted-digest-algorithms:
            - SHA-1 # Adds this value to the list if not already present
        relaxed:
        validation:
          permitted-digest-algorithms:
            - SHA-1
            - SHA-384- # If the last character is '-' it means remove this value
from list if it is present
          excluded-tests:
            - TS_33_310.certRequestId # Ignore the actual value, the spec says this
value MUST be 0 but AS tool does not use that value
            - certConf.SignatureBased.extraCertsPresent # Ignore checking if
extaraCerts are present, spec says they SHALL be omitted but AS tool provides them
and we're just going to ignore them
        password-based:
        validation:
          permitted-digest-algorithms:
            - SHA-1
          excluded-tests:
            - TS_33_310.certRequestId
            - ir.Protection.rejectPBM # Enable PBM for just IR operations as KUR
must be signed by previously issued certificate for the device
            - certconf.Protection.rejectPBM # Since only IR is PBM enabled, and
certConf uses the same protection as the initial request, certConf for KUR won't be
affected
```

Mandatory: Define at least one specification.

excluded-tests

The list of specific tests to be excluded during validation of the message.

Mandatory: No.

implementation

The extended validation mechanism. Supported values are:

check-domains-from-csr

`true` for CA Gateway to make CAA checks for domains inside the CSR, `false` otherwise.

Mandatory: No. This optional parameter defaults to `true`.

dns-server.<i>.<setting>

The DNS settings, where "i" is an index starting at 0. You can omit this index when defining a single DNS.

<setting>	Value	Default
ip	The IP address of the local DNS server that CA Gateway will use to look up the DNS issuer resource record.	-
port	The port of the DNS server.	53
timeout-first-seconds	The timeout of the first DNS lookup attempt, in seconds.	3
timeout-second-seconds	Timeout of the second DNS lookup attempt, in seconds. Applicable if the first attempt results in an error.	7
timeout-dsquery-seconds	Timeout in seconds of the Delegation Signer (DS) query when querying DNSSEC support.	7

Mandatory: Yes.

issuer-string

The CAA issuer name, as expected in the DNS resource record. Real-world examples include:

- entrust.net
- pki.goog

The name is owned and defined by the issuer and registered in DNS for any CA to check.

Mandatory: Yes

log-server.<i>.<setting>

The settings of each log server CA Gateway must contact to request the signed CT response. Therefore, you must define at least one server, with <i> starting a 0.

<setting>	Value
name	A friendly name for the log server. For example: "Google Log Server".

i CA Gateway can cope with slow running or unresponsive log servers when the number of servers configured under `log-server.<i>.setting` exceeds the number of required responses.

The general form of this JSON value is:

```
{
  sct-policy:[
    [<months-threshold>,<threshold-equals>,<google-min-responses>,<non-google-min-responses>]
  ],
  insurance:<insurance>
}
```

See below for a description of each field.

- [months-threshold](#)
- [threshold-equals](#)
- [google-min-responses](#)
- [non-google-min-responses](#)
- [insurance](#)

Mandatory: No. This optional parameter defaults to:

```
{
  sct-policy:[
    [39,true,0,1]
  ],
  insurance:0
}
```

In the configuration, you can flatten this default value to:

```
{sct-policy:[[38,true,0,1]],insurance:0}
```

months-threshold

The applicability of the `sct-policy` policy according to the certificate lifetime, as a number of months.

- When defining multiple policies, this value determines which policy to apply for issuing a certificate.
- Specifying a high value ensures this policy applies to all certificates issued.

threshold-equals

The operator for comparing the months-threshold and the actual certificate lifetime.

Value	Operator
true	=

Value	Operator
false	<=

google-min-responses

The minimum number of Google-compatible log server responses to include in the issued certificate.

non-google-min-responses

The minimum number of non-Google-compatible log server responses to include in the issued certificate.

insurance

The number of log server responses to collect above the following minimum.

```
google-min-responses + non-google-min-responses
```

log-server.<i>.<setting>


The settings of each log server CA Gateway must contact to request the signed CT response. Therefore, you must define at least one server, with `<i>` starting a 0.

<setting>	Value
name	A friendly name for the log server. For example: "Google Log Server".
url	The URL of the log server
google	True if the SCTs produced by this log server are Google Chrome compatible.
public-key	The public key of the log server, as a Base64 DER-encoded public key. Log servers typically advertise their keys publicly.
tls-trust-anchor	The trust anchor for the CT Filter to perform the TLS handshake with the log server, as a Base64 DER-encoded certificate.

Mandatory: Yes.

proxy-host-name

The hostname of the proxy for accessing the CA server.

 The proxy configured using this parameter is part of your corporate infrastructure; it is not an Entrust product.

Mandatory: Only when traffic to the CA server passes through a proxy.

socket-timeout-millis

The TCP Socket timeout for the HTTP communication with the log server, in milliseconds.

Mandatory: No. This optional parameter defaults 5000 milliseconds.

integrators

Under this section, add the following properties for each integrator.

- [name](#)
- [tenant-ids](#)
- [unique-id](#)


name

A friendly name for the integrator.

Mandatory: Yes.

tenant-ids

One or more of the tenant identifiers defined when configuring tenants. This setting maps an integrator to one or more tenants.

 Do not map multiple integrators to the same tenant. Errors will occur if you map more than one integrator to the same tenant.

Mandatory: Yes.

unique-id

A unique identifier for the integrator. When creating certificates for clients, you can specify this integrator ID to map a client to an integrator.

Mandatory: Yes.

license

Under this section, define the license settings required by your installation mode.

- [signed-path](#)
- [zip-path](#)
- [zip-password](#)

signed-path

The path of the signed license file.

- If the file is in the host configuration folder, you only must provide the file name.
- If you place the file outside of the host configuration folder, you must provide the full path and grant to this file the same permissions granted to the host configuration folder.

tual.properties

The path of the file jTinyUAL properties file.

Mandatory: When [Securing settings with jTinyUAL](#).

logging

Under this section, define the following logging settings.

- [jtk](#)
- [level.root](#)
- [ssl.jsse](#)

jtk

Under this section, define the following JTK logging settings.

- [debug](#)
- [level](#)

debug

`true` to enable JSTK logging, `false` otherwise.

Required: No. This optional parameter defaults to `false`.

level

The level of detail for the JSTK logs.

- 0
- 1
- 2
- 3
- 4

Required: No. This optional parameter defaults to 0.

level.root

The level of detail for the root CA Gateway logger. In increasing severity.

Level	Description
TRACE	Captures the most detailed information
DEBUG	Provides diagnostic information, such as HTTP headers
INFO	Records general operational events that confirm the system is working as expected

Level	Description
WARN	Indicates a potential issue or unexpected situation that does not stop the application but may require attention
ERROR	Reports errors that prevent a specific operation from completing, but do not cause the entire application to stop
FATAL	Signals a severe failure that forces the application or a major subsystem to shut down
OFF	Disables all logging output

Each level sets the lowest message level to show. For example, the `WARN` level provides messages with the `WARN`, `ERROR`, and `FATAL` status.

Required: No. This optional parameter defaults to `INFO`.

ssl.jsse

Under this section, define the following JSSE (Java Secure Socket Extension) logging settings.

- `debug`
- `level`

debug

`true` to enable JSSE (Java Secure Socket Extension) logging, `false` otherwise.

Required: No. This optional parameter defaults to `false`.

level

The level of detail for the JSSE logs. Supported values are:

- `ssl`
- `ssl:handshake`
- `all`

Required: No. This optional parameter defaults to `ssl`.


management

Under this section, configure the Spring actuator-based liveness and health check endpoints for CA Gateway.

- `endpoint`
- `endpoints`
- `server`

Type	Description
jks	Java truststore
pkcs12	PKCS #12 truststore

Mandatory: Yes.

 The decision on synchronization depends on the deployed environment. System administrators must observe the nodes and detect if the lack of synchronization impacts the results.

Running the CA Gateway Docker container

Use the following Docker command to run the CA Gateway container in SSL/TLS secure mode.

```
docker run -d -p <CAGW_HOST_PORT>:<server.port> -p
<MONITOR_HOST_PORT>:<management.server.port> -v <HOST_CONFIG>:/etc/cagw/config -h
<HOST> cagw/api:latest --cap-drop ALL
```

If the host system uses SELinux, append a `Z` to the volume mount string. For example:

```
docker run -d -p 8444:8080 -p 9444:9090 -v /home/myuser/cagw/config:/etc/cagw/
config:Z -h myserver cagw/api:latest --cap-drop ALL
```

See the following sections for a description of each option.

- `--cap-drop ALL`
- `-d`
- `-h <HOST>`
- `-p <CAGW_HOST_PORT>:<server.port>`
- `-p <MONITOR_HOST_PORT>:<management.server.port>`
- `-v <HOST_CONFIG>:/etc/cagw/config`
- `-e JAVA_OPTS="-Dcagw.enable.crl dp.checking=true"`

`--cap-drop ALL`

Drop all Linux capabilities from the Docker container.

`-d`

Launch the container in the background. Remove this option to see the CA Gateway log output while running, although it might terminate CA Gateway when closing the terminal.

`-h <HOST>`

Use `<HOST>` when logging the active URL, where `<HOST>` is the hostname of the CA Gateway server. When omitting this option, the active URL recorded in the logs displays a random hostname. For example:

```
Active URL: http://f719b61263fa:8444/cagw/swagger-ui/index.html
```

`-p <CAGW_HOST_PORT>:<server.port>`

Map the following ports.

- The `<CAGW_HOST_PORT>` user-selected port to expose CA Gateway on the host machine.

- The `<server.port>` value of the server `port` configuration parameter in the `application.yml` file.

`-p <MONITOR_HOST_PORT>:<management.server.port>`

Map the following ports:

- The `<MONITOR_HOST_PORT>` user-selected port to expose the monitoring service on the host machine.
- The `<management.server.port>` value of the management server `port` configuration parameter in the `application.yml` file.

See [Checking the CA Gateway health](#) for how to check the health check and monitoring service.

`-v <HOST_CONFIG>:/etc/cagw/config`

Map the following folders.

- The `<HOST_CONFIG>` configuration folder described in [Creating the configuration and credentials folders](#).
- The `/etc/cagw/config` configuration folder in the Docker container.

Alternatively, you can add several `-v` flags to map different files and folders. For example:


```
-v /home/myuser/cagw/config/application.yml:/etc/cagw/config/application.yml:ro  
-v /home/myuser/cagw/config/tls/cagw-tls.p12:/etc/cagw/config/cagw-tls.p12:ro
```

The `ro` option sets the read-only mode for the mapped file.

`-e JAVA_OPTS="-Dcagw.enable.crlp.checking=true"`

Enable CRL checking. This command adds the following option to the `JAVA_OPTS` environment variable passed to the CA Gateway.

```
-Dcagw.enable.crlp.checking=true
```

 When the revocation checking is enabled, all client certificates must include a CDP extension pointing to an up-to-date CRL. Handshakes will not complete if the client certificate does not include a CDP extension or the URL in this extension is unavailable.

Stopping the execution

To look up the name of the Docker container:


```
docker container ls
```

To stop the Docker container with `<CONTAINER_ID>` identifier:

```
docker container stop <CONTAINER_ID>
```

9 Enabling CRL revocation check

When starting the CA Gateway server, you can use the `-e` option to enable CRL checking. See [Running the CA Gateway Docker container](#) for details.

 When the revocation checking is enabled, all client certificates must include a CDP extension pointing to an up-to-date CRL. Handshakes will not complete if the client certificate does not include a CDP extension or the URL in this extension is unavailable.

10 Configuring clients

To configure a client of the CA Gateway API, set the following parameters in the `application.yml` file.

- `managed-cas.<ca>`
- `clients`
- `tenants`
- `integrators`

11 Issuing public trust certificates

CA Gateway supports issuing certificates intended to be publicly trusted. See the following sections for how to enable this feature using filter lists.

- [CA Authorization](#)
- [Certificate Transparency](#)

i In the current release, only the Entrust CA described in [Integrating an Entrust CA](#) supports this feature.

CA Authorization

With the configured CAA filter, CA Gateway lookups CAA records for the domain and each parent domain. For example, for the following domain:

www.acme.com

CA Gateway performs the following lookups:

www.acme.com

acme.com

com

CA Gateway traverses up the tree in search of CAA records. This CAA check passes if:

- The issuer in a CAA record matches the issuer defined in the `issuer-string` setting of the [com.entrust.CAAuthorization](#) filter.
- No CAA record defines an issuer or specifies "Any CA". In this case, the domain owner is not asserting a particular issuing CA.
- No CAA record is found. In this case, the domain owner is not asserting a particular issuing CA.

The above applies to each domain requested in the CA Gateway enrollment request. For example, domains inside the CSR, subject to the following flag if applied.

`optionalCertificateRequestDetails/useSANFromCSR`

Domains are requested in the separate subjectAltNames, or in the following fields externally to the CSR.

`optionalCertificateRequestDetails/extensions`

CA Gateway will check CAA records for wildcard domains under [RFC8659](#).

Defining Multiple DNS Servers

When defining multiple DNS servers, the DNS lookups run in parallel. The check for a domain stops when reaching the number of positive responses defined in the `dns.response-threshold` configuration parameter. Thus, this parameter provides additional assurance by forcing consultation of multiple separate DNS responders while allowing some contingency if a DNS server fails to respond quickly.

For example, when using three DNS servers, setting `dns.response-threshold` to "2" ensures at least two positive DNS checks against two distinct responders while allowing for the unavailability of one of the three responders.


DNS Infrastructure Guidance

Before using the CAA check feature of CA Gateway, read [RFC8659](#) with particular attention to section 5 covering security considerations. This RFC provides rules and advice for CAA checking. Deploying the DNS infrastructure is the responsibility of the customer.

The DNS responders referenced in the CA Gateway configuration are under the CA and CA Gateway responsibility (not under the control of a third-party cache such as Google or CloudFlare). All records received by CA Gateway come from authoritative nameservers. Caching of these records at the responder is allowed.

DNSSEC

As stated by [RFC8659](#), DNSSEC allows CA Gateway to ensure that an empty resource record (potentially containing the domain owner's stated issuer) is legitimately empty or not empty after a record suppression. CA Gateway will validate DNSSEC if present but still proceed if no DNSSEC applies for the domain.

 CA Gateway does not archive the DNSSEC proof for future audits.

Certificate Transparency

CA Gateway can collect a set of signed CT log server responses and ask the underlying CA if the certificates for public trust include these responses in an SCT List extension. The certificate transparency filter:

1. Sends parallel requests to all of the configured log servers.
2. Waits for sufficient log server responses to arrive. In the filter configuration, a certificate transparency policy states the type and the minimum of required responses.
3. Requests the final certificate to the CA.

This approach allows defining a surplus of log servers to guard against slow or offline servers.

12 Administrating the deployment

Once deployed, you can administrate CA Gateway as explained below.

- [Checking the CA Gateway health](#)
- [Checking the health of a CA](#)
- [Managing logs](#)
- [Updating the configuration](#)

Checking the CA Gateway health

As explained in [API endpoints](#), CA Gateway provides the following endpoints to check the health of the CA Gateway server.

- [Health endpoint](#)
- [Disk space endpoint](#)
- [Metrics endpoint](#)

Checking the health of a CA

As explained in [API endpoints](#), CA Gateway provides the [CA status endpoint](#) endpoint to check the health of a Certificate Authority.

Managing logs

On-premises CA Gateway sends all logging to the standard output captured by the Docker or Podman infrastructure.

- [Getting the container identifier](#)
- [Viewing logs](#)
- [Following logs](#)
- [Dumping logs](#)
- [Checking error codes](#)

Getting the container identifier

Run the following command to get the `<CONTAINER_ID>` identifier of the container recording the logs

```
[docker|podman] ps
```

For example

```
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f5ed108d9981 cagw/api:1.6.0 "/entrypoint.sh" 25 hours ago Up 25 hours
0.0.0.0:443-8080/tcp unruffled_curran
```

Viewing logs

Run the following container to view the logs recorded by the container with the `<CONTAINER_ID>` identifier.

```
[docker|podman] logs <CONTAINER_ID>
```

Following logs

Run the following container to follow the logs recorded by the container with the `<CONTAINER_ID>` identifier.

```
[docker|podman] logs <CONTAINER_ID> -f
```

Dumping logs

Run the following container to dump into the `<FILE>` file the logs recorded by the container with the `<CONTAINER_ID>` identifier.

```
[docker|podman] logs <CONTAINER_ID> > <FILE>
```

Checking error codes

For a description of each error code recorded in the CA Gateway logs, see the CA Gateway API documentation at:

```
https://<HOST>:<CAGW_HOST_PORT>/<server.servlet.context-path>/docs
```

Where `<HOST>` and `<CAGW_HOST_PORT>` are the hostname and port of the CA Gateway service. CA Gateway logs can also include the following warning message.

```
Version <= 1.4 profile configuration detected for CA <ca>. This configuration syntax is deprecated. Please update.
```

CA Gateway records this warning message when the YAML configuration includes a deprecated profile syntax under:

```
cagw.authorities.managed-cas.<CA>
```

To avoid this message, configure the CA profiles as described in the configuration guide.

Updating the configuration

Occasionally, you may need to update the configuration files used by CA Gateway. For example, you may need to add or edit a managed CA.

To update the configuration files for CA Gateway

1. Stop CA Gateway, as explained in [Stopping the execution](#).
2. In a location outside the container, make changes to the configuration files.
3. Re-mount the modified files into the container.
4. Restart CA Gateway with the Docker `restart` command instead of `run`. Otherwise, the container will no longer decrypt the tual files. You will need to add the original unencrypted secrets again to the tual files and remove the encrypted entries.

13 API endpoints

To enable endpoints in Docker deployments, you must:

1. Configure the `health.group.<group>.include` parameter in the `application.yml` file.
2. Run CA Gateway with the `-p` command described in [Running the CA Gateway Docker container](#).

CA Gateway will expose the following endpoints.

Description	Endpoint
CA properties endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/ properties? fields={properti es}</pre>
CA status endpoint	<pre>https://{host}: {port}/ {server.servlet. context-path}/ v1/certificate- authorities/ {caId}/status</pre>
Certificate actions endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/ certificates/ {sn}/actions</pre>

Description	Endpoint
Certificate events endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/ certificate- events</pre>
Disk space endpoint	<pre>https://{host}: {port}/ <management.endp oints.web.base- path>/health/ {group}/ diskSpace</pre>
Documentation endpoint	<pre>https://{host}: {port}/ {server.servlet. context-path}/ doc</pre>
Domain actions endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/domains/ {domain}/actions</pre>

Description	Endpoint
Domains endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/domains</pre>
Enrollments endpoint	<pre>https://{host}: {port}/ <server.servlet. context-path>/ v1/certificate- authorities/ {caId}/ enrollments</pre>
Health endpoint	<pre>https://{host}: {port}/ {management.endp oints.web.base- path}/health</pre>
Metrics endpoint	<pre>https://{host}: {port}/ {management.endp oints.web.base- path}/prometheus</pre>
Ping endpoint	<pre>https://{host}: {port}/ {management.endp oints.web.base-</pre>

Description	Endpoint
	<code>path}/health/ {group}/ping</code>
Swagger endpoint	<code>https://{host}: {port}/ {server.servlet. context-path}/ swagger-ui</code>
Version endpoint	<code>https://{host}: {port}/ {server.servlet. context-path}/v1</code>

CA properties endpoint

The following endpoint returns property values on certificate authorities.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/  
{caId}/properties?fields={properties}
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter
{caId}	The certificate authority identifier

Field	Value
{properties}	A comma-separated list of the <code>defaultPolicyOIDs</code> , <code>encryptionPolicyOIDs</code> , and <code>verificationPolicyOIDs</code> property identifiers

For example, the following request checks all these properties on the Entrust Certificate Authority with the `CA3` identifier.

```
GET https://localhost:8444/cagw/v1/certificate-authorities/CA3/status?
fields=defaultPolicyOIDs,encryptionPolicyOIDs,verificationPolicyOIDs
```

The response looks like the following.

```
{
  "type": "CAPropertiesResponse",
  "CAPropertiesInformation": {
    "properties": {
      "defaultPolicyOIDs": [
        "1.1.1.1",
        "2.2.2.2"
      ],
      "encryptionPolicyOIDs": [
        "1.1.1.1"
      ],
      "verificationPolicyOIDs": [
        "2.2.2.2"
      ]
    }
  }
}
```

CA status endpoint

The following endpoint returns the `UP` or `DOWN` status of a certificate authority.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/
{caId}/status
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server

Field	Value
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter
{caId}	The certificate authority identifier

For example, the following request checks the status of a certificate authority with the `CA3` identifier.

```
GET https://localhost:8444/cagw/v1/certificate-authorities/CA3/status
```

The response looks like the following.

```
{
  "type" : "CAStatusResponse",
  "status" : "UP",
}
```

Certificate actions endpoint

The following endpoint returns the supported actions for a certificate.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/
{caId}/certificates/{sn}/actions
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{management.endpoints.web.base-path}	The value of the management.endpoints.web.base-path parameter
{group}	One of the groups listed by the Health endpoint

See below for CA-specific considerations on this endpoint.

- [Revoking GlobalSign certificates with the certificate actions endpoint](#)
- [Renewing GlobalSign certificates with the certificate actions endpoint](#)
- [Renewing Sectigo CA certificates with the certificate actions endpoint](#)

Revoking GlobalSign certificates with the certificate actions endpoint

When using the certificate actions endpoint to revoke a certificate issued by a GlobalSign Certificate Center (GCC) authority, the `action.reason` parameter is:

- Required by the CA gateway API.
- Ignored by the GlobalSign MSSSL (Managed SSL) API, which does not support specifying a revocation reason.

Therefore, this value must be included in the requests, although it will not be evaluated. For example:

```
{
  "action": {
    "comment": "revoke",
    "succeedIfAlreadyInRequestedState": true,
    "type": "RevokeAction",
    "reason": "unspecified"
  }
}
```

See below for a parameter reference.

- `action.comment`
- `action.succeedIfAlreadyInRequestedState`
- `action.reason`
- `action.type`

 Parameter names are case sensitive.

`action.comment`

A descriptive text on the requested action.

Mandatory: No.

`action.succeedIfAlreadyInRequestedState`

The request output when the requested action has already been performed.

Value	Request output
true	Success (the operation is idempotent)
false	Error

Mandatory: No. This optional value defaults to `false`.

`action.reason`

The certificate revocation reason. Supported values are the following.

- `unspecified`
- `keyCompromise`

- affiliationChanged
- superseded
- cessationOfOperation

Mandatory: Yes.

action.type

The type of action requested to the endpoint.

Endpoint	Value	Action
CA actions	RenewAction	Renew a certificate
CA actions	RevokeAction	Permanently invalidate a certificate
CA domain actions	ReverifyDomainAction	Reverify a previously validated domain because the prior validation expired, failed, or must be refreshed to meet CA/Browser Forum or GlobalSign requirements.

Mandatory: Yes.

Renewing GlobalSign certificates with the certificate actions endpoint

When using the certificate actions endpoint to renew a certificate issued by a GlobalSign Certificate Center (GCC) authority, the JSON request must include the [action.properties.validityPeriod](#) field. For example:

```
{
  "action": {
    "comment": "renew",
    "properties": {
      "csr": "...",
      "validityPeriod": "P1Y0M0D"
    },
    "type": "RenewAction"
  }
}
```

See below for a parameter reference.

- [action.comment](#)
- [action.properties.csr](#)
- [action.properties.validityPeriod](#)
- [action.type](#)

 Parameter names are case sensitive.

action.comment

A descriptive text on the requested action.

Mandatory: No.

`action.properties.csr`


The base64 encoding of a certificate request.

Mandatory: When requesting or renewing a certificate for a client-generated key pair and CSR.

`action.properties.validityPeriod`

The duration for which the certificate will be valid, as an ISO-8601 duration. See the table below for examples.

validityPeriod	Start date	Duration	End date
"2018-07-06T13:00Z/ 2019-07-06T13:00:00Z"	July 6, 2018 at 13:00 UTC	Automatically calculated	July 6, 2019 at 13:00 UTC
"2018-07-06T13:00Z/ P1Y3M0DT0H0M"	July 6, 2018 at 13:00 UTC	1 year and 3 months	Automatically calculated
"P0Y3M0DT0H0M/ 2018-12-31T00:00Z"	Automatically calculated	3 months	31, 2018 at 00:00 UTC
"P1Y3M10DT0H0M"	Current date	1 year, 3 months, 10 days	Automatically calculated

 The selected value must comply with the GlobalSign profile maximum validity rules.

`action.type`

The type of action requested to the endpoint.

Endpoint	Value	Action
CA actions	RenewAction	Renew a certificate
CA actions	RevokeAction	Permanently invalidate a certificate
CA domain actions	ReverifyDomainA ction	Reverify a previously validated domain because the prior validation expired, failed, or must be refreshed to meet CA/Browser Forum or GlobalSign requirements.

Mandatory: Yes.

Renewing Sectigo CA certificates with the certificate actions endpoint

When using the certificate actions endpoint to renew a certificate issued by a Sectigo CA, the JSON request must include the `action.properties.csr` field. For example:

```
{
  "action": {
    "comment": "Test",
    "properties": {
      "csr":
      "MIIC1TCCAAb0CAQAwIzEhMB8GA1UEAwYbWF3aWEudGVzdC5zZW50aWdvLnJlbnV3MIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt/
      hiyBuXWjhsnhIrr32Xnv9fUwrUKru4UjuNEtmqt0ftfNkUMnBmwKFZ1DxHeuatzeVYXPCXoWAFteU2g4JibAC
      MJAEDMARd338Xi0VDRZW2e/V0s49wZJN9q7f+IFVw5B3x4qCDSE7zjsohgP3/
      ISvMyHkjs9h4sYw+Bn9+VBk1yA+S3IrWigbMA5ILod2y+MKZ6F2kdcgKl6ytFzDFV9n3TgzpJKVaQvgH696b6
      +4cqZMlOn+2ScuQDJ+bLZixpqrsnT9ag4It2RPtputIX/
      DW5LvHfFGERPyD+GI10hi6B8iUxm6wfpcGwzkxxy8F5LZAD5FKUvKk94M4+yaAwIDAQABoG0wawYJKoZIhvc
      NAQK0MV4wXDAJBgNVHRMEAjAAMASGA1UdDwQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIw
      IwYDVR0RBWwGoIYbWF3aWEudGVzdC5zZW50aWdvLnJlbnV3MA0GCSqGSIb3DQEBCwUAA4IBAQAk7YbCBBrqN
      fhJsnQbdTa68bt8UTqbPL/2PsapsKPOkPugggN5dP0D5glzRG/tiBhsNNxjEPW+TKyX7g8j5ACHCt5r/
      cSy3zKkj9Z2TJCj0yt+wC1GYqDLse7vHqb+102920GwqY/dRYq/uswuP4ZTo/
      TpVIEbwhXZzWVfaAC40F809Wb8S8UCyuv2hV+NW9gE9bTEC1sNr1XU9+5TxikRbHda2SPqR+pK2/
      yfdMQ970e82jfhIsvBJWKHBnbvd3yrRVeCKJ0kQ1yuuR2/
      kzPdfB3DG5xkDIcNodvDznKEMHLv5bR8f+kzLz4poYggZa2Kw617pUgB5GGP2AiWiHn/"
    },
    "type": "RenewAction"
  }
}
```

See below for a parameter reference.

- [action.comment](#)
- [action.properties.csr](#)
- [action.type](#)

 Parameter names are case sensitive.

`action.comment`

A descriptive text on the requested action.

Mandatory: No.

`action.properties.csr`

The base64 encoding of a certificate request.

Mandatory: When requesting or enewing a certificate for a client-generated key pair and CSR.

`action.type`

The type of action requested to the endpoint.

Endpoint	Value	Action
CA actions	RenewAction	Renew a certificate
CA actions	RevokeAction	Permanently invalidate a certificate
CA domain actions	ReverifyDomainAction	Reverify a previously validated domain because the prior validation expired, failed, or must be refreshed to meet CA/Browser Forum or GlobalSign requirements.

Certificate events endpoint

The following endpoint returns a list of events in the certificates issued by a certificate authority.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/{caId}/certificate-events
```

See the table below for a description of each field.


Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter
{group}	One of the groups listed by the Health endpoint

This endpoint returns the following values.

- [events](#)
- [nextPageIndex](#)
- [hasMore](#)

events

A list of events on the issued certificates. For example, certificate revocation.

 The endpoint retrieves 0 events for days without events.

nextPageIndex

An index to track which day and which event should be collected next.

- This index depends on the value assigned to the [default-query-page-size](#) configuration parameter.
- After the first query to the endpoint, all subsequent queries must include this index.


hasMore

A flag indicating if the endpoint has more results to return.

Value	Description
true	The endpoint can return more events from the last <code>nextPageIndex</code> till today
false	All events have been collected, so the caller can stop invoking the endpoint

Disk space endpoint

The following endpoint returns the disk space of the CA Gateway server for a group.

 To enable this endpoint, you must configure the [management.endpoints.web.exposure.include](#) parameter.

```
https://{host}:{port}/{management.endpoints.web.base-path}/health/{group}/diskSpace
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{management.endpoints.web.base-path}	The value of the management.endpoints.web.base-path parameter
{group}	One of the groups listed by the Health endpoint

For example, to check the disk space for the `custom` group.

```
https://localhost:9444/cagw/management/actuator/health/custom/diskSpace
```

If the server is up, this endpoint will return a response like the following.

```
{"status":"UP","details":
{"total":1013309239296,"free":765931622400,"threshold":10485760,"exists":true}}
```

Documentation endpoint

The following endpoint provides documentation on using the CA Gateway API for certificate policy, certificate issuance, and certificate lifecycle management.

```
https://{host}:{port}/{server.servlet.context-path}/doc
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter

Domain actions endpoint

The following endpoint supports domain-related actions.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/  
{caId}/domains/{domain}/actions
```

See the table below for a description of each field.

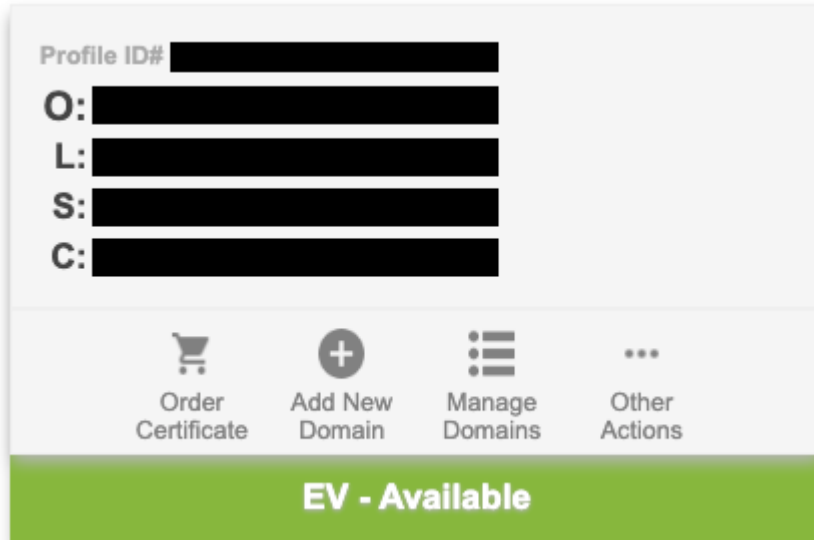
Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter
{caId}	The certificate authority identifier
{domain}	The domain identifier

See below for CA-specific considerations on this endpoint.

- [Verifying a GlobalSign domain with the domain actions endpoint](#)

Verifying a GlobalSign domain with the domain actions endpoint

To check if a GlobalSign profile supports the domains endpoint, navigate to the profile information under **Managed SSL**.



When displayed, the **EV - Available** message indicates that the profile supports requests to add a domain – for example:

```
{
  "name": "mydomain.com"
}
```

The request to the domain actions endpoint must contain the following parameters.

- [action.type](#)
- [action.verificationMethod](#)
- [action.properties.tagLocation](#)

For example:

```
{
  "action": {
    "properties": {
      "tagLocation": "store.example.com,example.com"
    },
    "type": "ReverifyDomainAction",
    "verificationMethod": "dns"
  }
}
```

action.type

The identifier of the domain action. See below for the values supported by this parameter.

Value	Action
ReverifyDomain Action	Verify a previously validated domain because the prior validation expired, failed, or must be refreshed to meet CA/Browser Forum or GCC requirements.


action.verificationMethod

The identifier of the domain verification method. See below for the values supported by this parameter.

Value	Verification
dns	DNS-based domain verification, GCC requires a unique Domain Validation Code (DVC) in a DNS TXT record to confirm domain control.
webServer	HTTP-based verification, also called file-based domain verification. GCC requires a Domain Validation Code (DVC) on the web server to confirm domain control.

action.properties.tagLocation

The location of the domain validation control number.

 When providing multiple values, separate them with commas.

See below for sample values when `verificationMethod` is `dns`.

Value	Validated FQDN
a.b.c.example.com,b.c.example.com,c.example.com,example.com	a.b.c.example.com
store.example.com,example.com	store.example.com
example.com	example.com

See below for sample values when `verificationMethod` is `webServer`.

Value	Requested domains
<code>https://example.com/.well-known/pki-validation/gsdv.txt</code>	<code>*.example.com</code> <code>example.com</code>
<code>https://example.com/.well-known/pki-validation/gsdv.txt,https://sub.example.com/.well-known/pki-validation/gsdv.txt</code>	<code>*.sub.example.com</code> <code>sub.example.com</code>
<code>https://example.com/.well-known/pki-validation/gsdv.txt,https://www.example.com/.well-known/pki-validation/gsdv.txt</code>	<code>*.www.example.com</code> <code>sub.example.com</code>

Domains endpoint

The following endpoint supports adding domains for EV (Extended Validation) profiles.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/{caId}/domains
```

See the table below for a description of each field.

Field	Value
<code>{host}</code>	The hostname or IP address of the CA Gateway host server
<code>{port}</code>	The port of the CA Gateway service
<code>{server.servlet.context-path}</code>	The value of the servlet.context-path parameter
<code>{caId}</code>	The certificate authority identifier

Enrollments endpoint

The following endpoint allows enrolling and renewing certificates.

```
https://{host}:{port}/{server.servlet.context-path}/v1/certificate-authorities/  
{caId}/enrollments
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter
{caId}	The certificate authority identifier

See below for CA-specific considerations on this endpoint.

- [Renewing GlobalSign certificates with the enrollments endpoint](#)
- [Renewing Sectigo CA certificates with the enrollments endpoint](#)
- [Requesting GlobalSign certificates with the enrollments endpoint](#)

Renewing GlobalSign certificates with the enrollments endpoint

When using the enrollments endpoint to renew a certificate from a GlobalSign CA, the request body must contain one of the following parameters.


- [properties.renewalOrderID](#)
- [properties.renewalOrderId](#)
- [properties.renewalSerialNumber](#)

For example:

```
{  
  "csr": "...",  
  "includeCa": false,  
  "optionalCertificateRequestDetails": {  
    "subjectDn": "cn=grapefruitdesk.com,o=Test Company,c=CA",  
    "useSANFromCSR": false,  
    "validityPeriod": "P1Y0M0D"  
  },  
  "profileId": "profileA",  
  "renewal": true,  
  "requiredFormat": {  
    "format": "pem"  
  },  
  "properties": {  
    "renewalOrderID": "mssl-order-id"  
  }  
}
```

See below for a parameter reference.

- [csr](#)
- [includeCa](#)
- [optionalCertificateRequestDetails.subjectDn](#)
- [optionalCertificateRequestDetails.useSANFromCSR](#)
- [optionalCertificateRequestDetails.validityPeriod](#)
- [profileId](#)
- [properties.renewalOrderID](#)
- [properties.renewalOrderId](#)
- [properties.renewalSerialNumber](#)
- [properties.specialInstructions](#)
- [renewal](#)
- [requiredFormat.format](#)

 Parameter names are case sensitive.

csr

The base64 encoding of a certificate request.

Mandatory: When requesting or renewing a certificate for a client-generated key pair and CSR.

includeCa

`true` to return the certificate and the issuing CA chain; `false` to return only the end-entity certificate.

Mandatory: No. This optional value defaults to `false`.

optionalCertificateRequestDetails.subjectDn

A Distinguished Name string used to override or supplement the DN from the CSR.

 Some GlobalSign issuance modes allow subject override; others require the subject to match the CSR.

Mandatory: No.

optionalCertificateRequestDetails.useSANFromCSR

A boolean controlling whether the CA should pull Subject Alternative Names (SANs) directly from the CSR.


Value	Action
true	Copy SANs from CSR.
false	SANs will not be taken from the CSR (they must come from other profile rules or request attributes).

Mandatory: No. This optional value defaults to `false`.

optionalCertificateRequestDetails.validityPeriod

The duration for which the certificate will be valid, as an ISO-8601 duration. See the table below for examples.

validityPeriod	Start date	Duration	End date
"2018-07-06T13:00Z/ 2019-07-06T13:00:00Z"	July 6, 2018 at 13:00 UTC	Automatically calculated	July 6, 2019 at 13:00 UTC
"2018-07-06T13:00Z/ P1Y3M0DT0H0M"	July 6, 2018 at 13:00 UTC	1 year and 3 months	Automatically calculated
"P0Y3M0DT0H0M/ 2018-12-31T00:00Z"	Automatically calculated	3 months	31, 2018 at 00:00 UTC
"P1Y3M10DT0H0M"	Current date	1 year, 3 months, 10 days	Automatically calculated

 The selected value must comply with the GlobalSign profile maximum validity rules.

Mandatory: No. This optional value defaults to the profile validity period.

profileId

The identifier of the certificate profile to issue or renew the certificate. To request a PKCS #12 containing a certificate and key pair generated by a GlobalSign CA, this value must identify a profile with one of the following product codes.

Product code	Certificate type
PV_INTRA_SKIP	IntranetSSL using AutoCSR signed with SHA1
PV_INTRA_SKIP_SHA2	IntranetSSL using AutoCSR signed with SHA-256

To request a certificate for a client-generated key pair and CSR, this value must identify a profile with any other product codes.

Mandatory: Yes.

properties.renewalOrderID

The existing GlobalSign order identifier of the certificate renewed certificate – for example: "mssl-order-id"

 This parameter name differs from `properties.renewalOrderID` only by the uppercase "D".

Mandatory: certificate renewal requires using one of the following fields.

- `properties.renewalOrderID`

- `properties.renewalOrderId`
- `properties.renewalSerialNumber`

`properties.renewalOrderId`

The existing GlobalSign order identifier of the certificate renewed certificate – for example: "mssl-order-id".

 This parameter name differs from `properties.renewalOrderID` only by the lowercase "d".

Mandatory: certificate renewal requires using one of the following fields.

- `properties.renewalOrderID`
- `properties.renewalOrderId`
- `properties.renewalSerialNumber`

`properties.renewalSerialNumber`

The serial number of the renewed certificate.

Mandatory: certificate renewal requires using one of the following fields.

- `properties.renewalOrderID`
- `properties.renewalOrderId`
- `properties.renewalSerialNumber`

`properties.specialInstructions`

Free-text field for internal notes about the order – for example: “contact ops before issuing”.

Mandatory: No.

`renewal`

`true` to request the **renewal** of an existing certificate; `false` to request a completely new certificate.

Mandatory: No. This optional value defaults to `false`.

`requiredFormat.format`

The certificate output format.

Value	Format
<code>pem</code>	Base64-encoded certificate
<code>der</code>	Binary DER
<code>pkcs12</code>	PKCS #12 containing a certificate and key pair generated by the CA (if supported by the selected profile).

Mandatory: Yes.

Renewing Sectigo CA certificates with the enrollments endpoint

When using the enrollments endpoint to renew a certificate issued by a Sectigo CA, the JSON request must include the `properties.renewalSerialNumber` setting. For example:

```
{
  "csr":
  "MIIC2zCCACMCAQAwJjEKMCIgA1UEAwbbWF3aWEudGVzdC5zZWN0aWdvLnJlbnV3LjA2MlIIBIjANBgkqhkiG
  9w0BAQEFAAOCAQ8AMIIBCgKCAQEAg6xeRXTqPSvmTVCTLi17pvGH2d5zQP2uiU5CR1r7ofGjJVumktr9MUf4DW
  mAgaaumkmjn0xRMSew4viE99nONP7XfciJLBRKgjBBnwr0fLCCIgRVKLR3PiTQ/
  N9J60gt8KkVYZ94Z1HmiGtXye6cZHX2ibnqfcW//
  tN5ewUrzQTFlyIAiw0QqRPkt10M4xFrRtFv2d07bf6rLPS75VCjYdfUaerHf8a0mllFwzmciaYaZkGM2/
  Uds+rel+0eVT8gzBqAqvqxCRXf20n67acXV1x2gStaa+uJmk46EacLtL+b/
  DHXigoNSqruchMUdHqfnd8DtVVCjPy2nTBrHmBvl8wIDAQABoHAwbgYJKoZIhvcNAQkOMWEwXzAJBGNVHRME
  AjAAMAsGA1UdDwQEAwIFoDADBgNVHSUEFjAUBgggrBgEFBQcDAQYIKwYBBQUHAwIwJgYDVRORBBSwHYIbbWF3a
  WEudGVzdC5zZWN0aWdvLnJlbnV3LjA2MA0GCSqGSIsb3DQEBcwUAA4IBAQCdKzAHBavqRIDzVu1VSj0bI/
  Vjg0cF982p6/lPnFKfL+13fxslUN003kG/uuh4Zdxo/00VMH6VS+iT4I06Gu0/
  7LoMg7oAxqN1FaBjK5VwLNAuro2uS54FGz+2ubYS8F+cTQxfM0uerNgVJXGFKaaHtrcoDynfp9PgyJg+I4zB
  aPR/5E051Am2t5K+wQHts0+ThoInnk5iwsL+SUnzCZwGwkf50Uf450/
  pyjKnUKsjXpfdPSoTrkVEFTniNkM+FzFwjhhECP536EzWLLsX/C3d/
  QGTndH4c4GN9qFGaybDmWZHHJHaj52/CeORJmkrI0REQi1R0FSGW8Zb66Dz9",
  "renewal": true,
  "properties": {
    "RenewalSerialNumber": "4c56db9f8f07af1e93c66a37e041ef8e"
  },
  "profileId": "profileA-local",
  "requiredFormat": {
    "format": "PEM"
  }
}
```

See below for a parameter reference.

- `csr`
- `profileId`
- `properties.renewalSerialNumber`
- `renewal`
- `requiredFormat.format`

 Parameter names are case sensitive.

CSR

The base64 encoding of a certificate request.

Mandatory: When requesting or renewing a certificate for a client-generated key pair and CSR.

profileId

The identifier of the certificate profile to use to issue or renew the certificate.

Mandatory: Yes.

properties.renewalSerialNumber

The serial number of the renewed certificate.

Mandatory: Yes.

renewal

`true` to request the **renewal** of an existing certificate; `false` to request a completely new certificate.

Mandatory: No. This optional value defaults to `false`.

requiredFormat.format

The certificate output format.

Value	Format
pem	Base64-encoded certificate
der	Binary DER
pkcs12	PKCS #12 containing a certificate and key pair generated by the CA (if supported by the selected profile).

Mandatory: Yes.

Requesting GlobalSign certificates with the enrollments endpoint


When using the enrollments endpoint to request a certificate from a GlobalSign CA, the `profileId` parameter must identify a GlobalSign MSSL (Managed SSL) profile. For example:

```
{
  "csr": "...",
  "includeCa": false,
  "optionalCertificateRequestDetails": {
    "subjectDn": "cn=grapefruitdesk.com,o=Test Company,c=CA",
    "useSANFromCSR": false,
    "validityPeriod": "P1Y0M0D"
  },
  "profileId": "profileA",
  "renewal": false,
  "requiredFormat": {
    "format": "pem"
  }
}
```

```
}  
}
```

See below for a parameter reference.

- [csr](#)
- [includeCa](#)
- [optionalCertificateRequestDetails.subjectDn](#)
- [optionalCertificateRequestDetails.useSANFromCSR](#)
- [optionalCertificateRequestDetails.validityPeriod](#)
- [profileId](#)
- [properties.specialInstructions](#)
- [renewal](#)
- [requiredFormat.format](#)

 Parameter names are case sensitive.

csr

The base64 encoding of a certificate request.

Mandatory: When requesting or renewing a certificate for a client-generated key pair and CSR.


includeCa

`true` to return the certificate and the issuing CA chain; `false` to return only the end-entity certificate.

Mandatory: No. This optional value defaults to `false`.

optionalCertificateRequestDetails.subjectDn

A Distinguished Name string used to override or supplement the DN from the CSR.

 Some GlobalSign issuance modes allow subject override; others require the subject to match the CSR.

Mandatory: No.

optionalCertificateRequestDetails.useSANFromCSR

A boolean controlling whether the CA should pull Subject Alternative Names (SANs) directly from the CSR.


Value	Action
true	Copy SANs from CSR.
false	SANs will not be taken from the CSR (they must come from other profile rules or request attributes).

Mandatory: No. This optional value defaults to `false`.

optionalCertificateRequestDetails.validityPeriod

The duration for which the certificate will be valid, as an ISO-8601 duration. See the table below for examples.

validityPeriod	Start date	Duration	End date
"2018-07-06T13:00Z/ 2019-07-06T13:00:00Z"	July 6, 2018 at 13:00 UTC	Automatically calculated	July 6, 2019 at 13:00 UTC
"2018-07-06T13:00Z/ P1Y3M0DT0H0M"	July 6, 2018 at 13:00 UTC	1 year and 3 months	Automatically calculated
"P0Y3M0DT0H0M/ 2018-12-31T00:00Z"	Automatically calculated	3 months	31, 2018 at 00:00 UTC
"P1Y3M10DT0H0M"	Current date	1 year, 3 months, 10 days	Automatically calculated

 The selected value must comply with the GlobalSign profile maximum validity rules.

Mandatory: No. This optional value defaults to the profile validity period.

profileId

The identifier of the certificate profile to issue or renew the certificate. To request a PKCS #12 containing a certificate and key pair generated by a GlobalSign CA, this value must identify a profile with one of the following product codes.

Product code	Certificate type
PV_INTRA_SKIP	IntranetSSL using AutoCSR signed with SHA1
PV_INTRA_SKIP_SHA2	IntranetSSL using AutoCSR signed with SHA-256

To request a certificate for a client-generated key pair and CSR, this value must identify a profile with any other product codes.

Mandatory: Yes.

properties.specialInstructions

Free-text field for internal notes about the order – for example: “contact ops before issuing”.

Mandatory: No.

renewal

`true` to request the **renewal** of an existing certificate; `false` to request a completely new certificate.

Mandatory: No. This optional value defaults to `false`.

requiredFormat.format


The certificate output format.

Value	Format
pem	Base64-encoded certificate
der	Binary DER
pkcs12	PKCS #12 containing a certificate and key pair generated by the CA (if supported by the selected profile).

Mandatory: Yes.

Health endpoint

The following endpoint returns information on the CA Gateway server health.

 To enable this endpoint, you must configure the [management.endpoints.web.exposure.include](#) parameter.

```
https://{host}:{port}/{management.endpoints.web.base-path}/health
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{management.endpoints.web.base-path}	The value of the management.endpoints.web.base-path parameter

For example:

```
{"status": "UP", "groups": ["custom"]}
```

See below for a description of each value.

- [status](#)
- [groups](#)

status


The ping status of the CA Gateway server.

groups

The list of user groups defined under the [endpoint](#) parameter.

Metrics endpoint

The following endpoint returns CA Gateway metrics in Prometheus-compliant format.

 To enable this endpoint, you must configure the [management.endpoints.web.exposure.include](#) parameter.

`https://{host}:{port}/{management.endpoints.web.base-path}/prometheus`

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{management.endpoints.web.base-path}	The value of the management.endpoints.web.base-path parameter

For example:

```
# HELP jvm_threads_live_threads The current number of live threads including both
daemon and non-daemon threads
# TYPE jvm_threads_live_threads gauge
jvm_threads_live_threads 51.0
# HELP spring_security_filterchains_AnonymousAuthenticationFilter_before_total
# TYPE spring_security_filterchains_AnonymousAuthenticationFilter_before_total
counter
spring_security_filterchains_AnonymousAuthenticationFilter_before_total{security_secu
rity_reached_filter_section="before",spring_security_filterchain_position="0",spring_
security_filterchain_size="0",} 8.0
# HELP jvm_gc_live_data_size_bytes Size of long-lived heap memory pool after
reclamation
# TYPE jvm_gc_live_data_size_bytes gauge
jvm_gc_live_data_size_bytes 8.7626752E7
# HELP executor_completed_tasks_total The approximate total number of tasks that have
completed execution
# TYPE executor_completed_tasks_total counter
executor_completed_tasks_total{name="applicationTaskExecutor",} 0.0
executor_completed_tasks_total{name="taskScheduler",} 2.0
```

```
# HELP system_cpu_count The number of processors available to the Java virtual
machine
# TYPE system_cpu_count gauge
system_cpu_count 8.0
```

Ping endpoint

The following endpoint returns the ping status of the CA Gateway server for a group.

⚠ To enable this endpoint, you must configure the [management.endpoints.web.exposure.include](#) parameter.

```
https://{host}:{port}/{management.endpoints.web.base-path}/health/{group}/ping
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{management.endpoints.web.base-path}	The value of the management.endpoints.web.base-path parameter
{group}	One of the groups listed by the Health endpoint

For example, to check the ping status for the `custom` group.

```
https://localhost:9444/cagw/management/actuator/health/custom/ping
```

If the server is up, this endpoint will return the following response.

```
{"status": "UP"}
```

Swagger endpoint

The following endpoint provides a Swagger UI to visualize and interact with the CA Gateway REST API.

```
https://{host}:{port}/{server.servlet.context-path}/swagger-ui
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter

✘ CA Gateway includes this UI to assist developers with API integrations, but it is not for, nor will it be supported for, production use. It is not a substitute for an administrator UI. We recommend using Entrust Certificate Hub or an equivalent interface provided by another product.

To test CA Gateway with Swagger

1. In the `application.yml` file, configure a tenant, or an integrator.
2. Install the tenant or integrator credential in the browser.
3. Make sure that the certification chain of the CA Gateway TLS certificate is trusted.
4. Navigate to the URL of the Swagger UI. For example:

```
https://localhost:8444/cagw/swagger-ui
```

5. When prompted by the browser, select the tenant or integrator credential.
6. Use the Swagger options to generate curl commands. For example, the following command lists the CAs visible to the tenant or integrator.

```
curl --cert-type P12 --cert tenant.p12:mypassword -X GET "https://cid-cagw.dev.entrust.local/cagw/v1/certificate-authorities" -H "accept: application/json"
```

✘ When running curl commands, some Linux versions do not support authenticating with a P12 file.

Version endpoint

The following endpoint returns version information on CA Gateway.

```
https://{host}:{port}/{server.servlet.context-path}/v1
```

See the table below for a description of each field.

Field	Value
{host}	The hostname or IP address of the CA Gateway host server

Field	Value
{port}	The port of the CA Gateway service
{server.servlet.context-path}	The value of the servlet.context-path parameter

 This endpoint is the main entry point for invoking API capabilities.

14 CA Capabilities reference

The "Get CA Capabilities" endpoint of the CA Gateway API informs on the capabilities supported by each type of CA. The following sections give a complete reference of the returned values.

- [Digicert CA capabilities](#)
- [EJBCA capabilities](#)
- [Entrust CA capabilities](#)
- [GlobalSign capabilities](#)
- [Microsoft CA capabilities](#)
- [Sectigo CA capabilities](#)

Digicert CA capabilities

See below for the capabilities supported by Digicert Certificate Authorities.

- [enrollments](#)
- [certificates](#)
- [certificate-events](#)
- [status](#)
- [ca-properties](#)
- [general](#)

enrollments

See below for the enrollment-supported capabilities

Name	Description	Endpoint	Supported
EnrollmentByCSR	Process certificate signing requests	/v1/certificate-authorities/{cald}/enrollments	✓
X509CertificateResponse	Returns certificate in X509 form	/v1/certificate-authorities/{cald}/enrollments	✓
PKCS12Response	Return certificate and key in PKCS#12 form	/v1/certificate-authorities/{cald}/enrollments	✗
SANInCSR	Process Subject Alternative Names (SANs) in CSR	/v1/certificate-authorities/{cald}/enrollments	✓
SANInRequest	Process Subject Alternative Names (SANs) in enrollment request	/v1/certificate-authorities/{cald}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order	/v1/certificate-authorities/{cald}/enrollments	✓

Name	Description	Endpoint	Supported
CAGeneratedKey	Generate the key and returned it to client in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✘
KeyInRequest	Supports client-generated keys for enrollment in the enrollment request	/v1/certificate-authorities/{caId}/enrollments	✘
CAGeneratedKey Backup	Supports key backup for CA generated keys	/v1/certificate-authorities/{caId}/enrollments	✘
ClientGenerated KeyBackup	Supports key backup for client-generated keys	/v1/certificate-authorities/{caId}/enrollments	✘
ExtensionInCSR	Process extension request in CSR	/v1/certificate-authorities/{caId}/enrollments	✘
ExtensionInRequest	Process extension request in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✘
ValidateProofOf Possesion	Validates proof of possession if requested	/v1/certificate-authorities/{caId}/enrollments	✘
ValidityPeriodIn Request	Supports validity period in request	/v1/certificate-authorities/{caId}/enrollments	✔

certificates

See below for the certificate-supported capabilities

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number	/v1/certificate-authorities/{caId}/certificates/{serial}	✔
CertificateAction	List supported actions on a certificate	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✔
CertificateRevocationReason	List supported revocation reasons for the certificate RevokeAction	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✔

certificate-events

See below for the certificate-event-supported capabilities

Name	Description	Endpoint	Supported
CertificateEvents	Support certificate events	/v1/certificate-authorities/{cald}/certificate-events	✓
MaxCertificateEventsPageSize	Return the maximum size of the certificate events Page	/v1/certificate-authorities/{cald}/certificate-events	✓

status

See below for the status-supported capabilities

Name	Description	Endpoint	Supported
CAStatus	Check whether the CA is up or down	/v1/certificate-authorities/{cald}/status	✗

ca-properties

See below for the CA property-supported capabilities

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieve CA properties	/v1/certificate-authorities/{cald}/properties	✗

general

See below for the general capabilities

Name	Description	Endpoint	Supported
PermitsDefaultCA	Can operate as the default CA		✓
SupportsMultipleCAs	Supports multiple CA configurations		✓

EJBCA capabilities

See below for the capabilities supported by EJB Certificate Authorities (EJB CAs).

- [enrollments](#)
- [certificates](#)

- [certificate-events](#)
- [ca-events](#)
- [recoveries](#)
- [subjects](#)
- [status](#)
- [ca-properties](#)

enrollments

See below for the enrollment-supported capabilities

Name	Description	Endpoint	Supported
EnrollmentByCSR	Process certificate signing requests	/v1/certificate-authorities/{caId}/enrollments	✓
X509CertificateResponse	Returns certificate in X509 form	/v1/certificate-authorities/{caId}/enrollments	✓
PKCS12Response	Return certificate and key in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✓
SANInCSR	Process Subject Alternative Names (SANs) in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
SANInRequest	Process Subject Alternative Names (SANs) in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order	/v1/certificate-authorities/{caId}/enrollments	✓
CAGeneratedKey	Generate the key and returned it to client in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✓
KeyInRequest	Supports client-generated keys for enrollment in the enrollment request	/v1/certificate-authorities/{caId}/enrollments	✗
CAGeneratedKeyBackup	Supports key backup for CA generated keys	/v1/certificate-authorities/{caId}/enrollments	✓
ClientGeneratedKeyBackup	Supports key backup for client-generated keys	/v1/certificate-authorities/{caId}/enrollments	✗

Name	Description	Endpoint	Supported
ExtensionInCSR	Process extension request in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
ExtensionInRequest	Process extension request in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✗
ValidateProofOfPossession	Validates proof of possession if requested	/v1/certificate-authorities/{caId}/enrollments	✓
ValidityPeriodInRequest	Supports validity period in request	/v1/certificate-authorities/{caId}/enrollments	✓

certificates

See below for the certificate-supported capabilities

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number	/v1/certificate-authorities/{caId}/certificates/{serial}	✓
CertificateAction	List supported actions on a certificate	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓
CertificateRevocationReason	List supported revocation reasons for the certificate RevokeAction	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓

certificate-events

See below for the certificate-event-supported capabilities

Name	Description	Endpoint	Supported
CertificateEvents	Support certificate events	/v1/certificate-authorities/{caId}/certificate-events	✓
MaxCertificateEventsPageSize	Return the maximum size of the certificate events Page	/v1/certificate-authorities/{caId}/certificate-events	✓

ca-events

See below for the event-supported capabilities

Name	Description	Endpoint	Supported
CAEvents	Support CA events	/v1/certificate-authorities/{caId}/ca-events	✘

recoveries

See below for the recovery-supported capabilities

Name	Description	Endpoint	Supported
Recover	Recover certificate by DN	/v1/certificate-authorities/{caId}/recoveries/{dn}	✔

subjects

See below for the subject management-supported capabilities

Name	Description	Endpoint	Supported
SearchBySubjectDN	Lookup certificates by subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}	✔
SubjectDNAction	List supported actions on a subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}/actions	✔

status

See below for the status-supported capabilities

Name	Description	Endpoint	Supported
CAStatus	Check whether the CA is up or down	/v1/certificate-authorities/{caId}/status	✔

ca-properties

See below for the CA property-supported capabilities

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieve CA properties	/v1/certificate-authorities/{cald}/properties	✓

Entrust CA capabilities

See below for the capabilities supported by Entrust Certificate Authorities.

- [enrollments](#)
- [certificates](#)
- [certificate-events](#)
- [subjects](#)
- [domains](#)
- [status](#)
- [ca-properties](#)

enrollments

See below for the enrollment-supported capabilities

Name	Description	Endpoint	Supported
EnrollmentByCSR	Process certificate signing requests	/v1/certificate-authorities/{cald}/enrollments	✓
X509CertificateResponse	Returns certificate in X509 form	/v1/certificate-authorities/{cald}/enrollments	✓
PKCS12Response	Return certificate and key in PKCS#12 form	/v1/certificate-authorities/{cald}/enrollments	✓
SANInCSR	Process Subject Alternative Names (SANs) in CSR	/v1/certificate-authorities/{cald}/enrollments	✗
SANInRequest	Process Subject Alternative Names (SANs) in enrollment request	/v1/certificate-authorities/{cald}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order	/v1/certificate-authorities/{cald}/enrollments	✗
CAGeneratedKey	Generate the key and returned it to client in PKCS#12 form	/v1/certificate-authorities/{cald}/enrollments	✓

Name	Description	Endpoint	Supported
KeyInRequest	Supports client-generated keys for enrollment in the enrollment request	/v1/certificate-authorities/{caId}/enrollments	✘
CAGeneratedKey Backup	Supports key backup for CA generated keys	/v1/certificate-authorities/{caId}/enrollments	✘
ClientGenerated KeyBackup	Supports key backup for client-generated keys	/v1/certificate-authorities/{caId}/enrollments	✘
ExtensionInCSR	Process extension request in CSR	/v1/certificate-authorities/{caId}/enrollments	✔
ExtensionInRequest	Process extension request in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✘
ValidateProofOf Possession	Validates proof of possession if requested	/v1/certificate-authorities/{caId}/enrollments	✘
ValidityPeriodIn Request	Supports validity period in request	/v1/certificate-authorities/{caId}/enrollments	✘

certificates

See below for the certificate-supported capabilities

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number	/v1/certificate-authorities/{caId}/certificates/{serial}	✔
CertificateAction	List supported actions on a certificate	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✔
CertificateRevocationReason	List supported revocation reasons for the certificate RevokeAction	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✔

certificate-events

See below for the certificate-event-supported capabilities

Name	Description	Endpoint	Supported
CertificateEvents	Support certificate events	/v1/certificate-authorities/{caId}/certificate-events	✓
MaxCertificateEventsPageSize	Return the maximum size of the certificate events Page	/v1/certificate-authorities/{caId}/certificate-events	✓

subjects

See below for the subject management-supported capabilities

Name	Description	Endpoint	Supported
SearchBySubjectDN	Lookup certificates by subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}	✓
SubjectDNAction	List supported actions on a subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}/actions	✓

domains

See below for the domain-supported capabilities

Name	Description	Endpoint	Supported
Verification Method	List of supported verification methods	/v1/certificate-authorities/{caId}/domains/{domain}	dns, email, manual, unknown, webServer
Verification Status	Supported verification statuses	/v1/certificate-authorities/{caId}/domains/{domain}	APPROVED, CANCELLED, DECLINED, EXPIRED, EXPIRING, INITIAL_VERIFICATION
Filter	Supported filters for domain search	/v1/certificate-authorities/{caId}/domains	domainName, evEligible, evExpiry, ovEligible, ovExpiry, verificationMethod, verificationStatus

status

See below for the status-supported capabilities

Name	Description	Endpoint	Supported
CASStatus	Check whether the CA is up or down	/v1/certificate-authorities/{caId}/status	✓

ca-properties

See below for the CA property-supported capabilities

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieve CA properties	/v1/certificate-authorities/{caId}/properties	✗

GlobalSign capabilities

See below for the capabilities supported by EJB Certificate Authorities (EJB CAs).

- [enrollments](#)
- [certificates](#)
- [certificate-events](#)
- [domains](#)
- [status](#)
- [ca-properties](#)

enrollments

See below for the enrollment-supported capabilities.

Name	Description	Endpoint	Supported
EnrollmentByCSR	Indicates that the CA supports certificate signing requests.	/v1/certificate-authorities/{caId}/enrollments	✓
X509CertificateResponse	Certificate can be returned in X509 form.	/v1/certificate-authorities/{caId}/enrollments	✓
PKCS12Response	Certificate (and key) can be returned in PKCS#12 form.	/v1/certificate-authorities/{caId}/enrollments	✓
SANInCSR	Subject alternative names in CSR are processed.	/v1/certificate-authorities/{caId}/enrollments	✓

Name	Description	Endpoint	Supported
SANInRequest	Subject alternative names in enrollment request are processed.	/v1/certificate-authorities/{caId}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order. This applies to requests with CSR.	/v1/certificate-authorities/{caId}/enrollments	✓
CAGeneratedKey	Key can be generated by the CA and returned to client(P12). Individual profiles may prevent keys from being generated by the CA.	/v1/certificate-authorities/{caId}/enrollments	✓
KeyInRequest	Client can send a key to be used for the enrollment in the enrollment request.	/v1/certificate-authorities/{caId}/enrollments	✗
CAGeneratedKeyBackup	Key backup can be requested if the CA generated the key.	/v1/certificate-authorities/{caId}/enrollments	✗
ClientGeneratedKeyBackup	Key backup can be requested if the client supplies the key during the request.	/v1/certificate-authorities/{caId}/enrollments	✗
ExtensionInCSR	Extension request in the CSR will be processed.	/v1/certificate-authorities/{caId}/enrollments	✗
ExtensionInRequest	Extension request in enrollment request will be processed.	/v1/certificate-authorities/{caId}/enrollments	✓
ValidateProofOfPossession	Validate proof of possession if requested.	/v1/certificate-authorities/{caId}/enrollments	✗
ValidityPeriodInRequest	A validity period can be supplied in the request	/v1/certificate-authorities/{caId}/enrollments	✓

certificates

See below for the certificate-supported capabilities.

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number.	/v1/certificate-authorities/{caId}/certificates/{serial}	✔
CertificateAction	Supported actions on a certificate.	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✘
CertificateRevocationReason	Supported revocation reasons for the certificate RevokeAction.	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✘

certificate-events

See below for the certificate-event-supported capabilities.

Name	Description	Endpoint	Supported
CertificateEvents	Supports certificate events.	/v1/certificate-authorities/{caId}/certificate-events	✔
MaxCertificateEventsPageSize	Maximum Size of content in Certificate Events Page	/v1/certificate-authorities/{caId}/certificate-events	✘

domains

See below for the domain-supported capabilities.

Name	Description	Endpoint	Supported
VerificationMethod	Available verification methods.	/v1/certificate-authorities/{caId}/domains/{domain}	✘
VerificationStatus	Possible domain verification status states.	/v1/certificate-authorities/{caId}/domains/{domain}	✘
Filter	Possible domain query filters	/v1/certificate-authorities/{caId}/domains	✘

status

See below for the status-supported capabilities.

Name	Description	Endpoint	Supported
CAStatus	Return whether the CA is up or down.	/v1/certificate-authorities/{caId}/status	✘

ca-properties

See below for the CA property-supported capabilities.

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieves CA properties	/v1/certificate-authorities/{caId}/properties	✘

Microsoft CA capabilities

See below for the capabilities supported by Microsoft Certificate Authorities.

- [enrollments](#)
- [certificates](#)
- [certificate-events](#)
- [recoveries](#)
- [subjects](#)
- [status](#)
- [ca-properties](#)

enrollments

See below for the enrollment-supported capabilities

Name	Description	Endpoint	Supported
EnrollmentByCSR	Process certificate signing requests	/v1/certificate-authorities/{caId}/enrollments	✔
X509CertificateResponse	Returns certificate in X509 form	/v1/certificate-authorities/{caId}/enrollments	✔
PKCS12Response	Return certificate and key in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✔

Name	Description	Endpoint	Supported
SANInCSR	Process Subject Alternative Names (SANs) in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
SANInRequest	Process Subject Alternative Names (SANs) in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order	/v1/certificate-authorities/{caId}/enrollments	✗
CAGeneratedKey	Generate the key and returned it to client in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✓
KeyInRequest	Supports client-generated keys for enrollment in the enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
CAGeneratedKey Backup	Supports key backup for CA generated keys	/v1/certificate-authorities/{caId}/enrollments	✓
ClientGenerated KeyBackup	Supports key backup for client-generated keys	/v1/certificate-authorities/{caId}/enrollments	✓
ExtensionInCSR	Process extension request in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
ExtensionInRequest	Process extension request in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✗
ValidateProofOf Possession	Validates proof of possession if requested	/v1/certificate-authorities/{caId}/enrollments	✗
ValidityPeriodIn Request	Supports validity period in request	/v1/certificate-authorities/{caId}/enrollments	✗

certificates

See below for the certificate-supported capabilities

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number	/v1/certificate-authorities/{caId}/certificates/{serial}	✓
CertificateAction	List supported actions on a certificate	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓
CertificateRevocationReason	List supported revocation reasons for the certificate RevokeAction	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓

certificate-events

See below for the certificate-event-supported capabilities

Name	Description	Endpoint	Supported
CertificateEvents	Support certificate events	/v1/certificate-authorities/{caId}/certificate-events	✓
MaxCertificateEventsPageSize	Return the maximum size of the certificate events Page	/v1/certificate-authorities/{caId}/certificate-events	✓

recoveries

See below for the recovery-supported capabilities

Name	Description	Endpoint	Supported
Recover	Recover certificate by DN	/v1/certificate-authorities/{caId}/recoveries/{dn}	✓

subjects

See below for the subject management-supported capabilities

Name	Description	Endpoint	Supported
SearchBySubjectDN	Lookup certificates by subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}	✓

Name	Description	Endpoint	Supported
SubjectDNAction	List supported actions on a subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}/actions	✓

status

See below for the status-supported capabilities

Name	Description	Endpoint	Supported
CAStatus	Check whether the CA is up or down	/v1/certificate-authorities/{caId}/status	✗

ca-properties

See below for the CA property-supported capabilities

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieve CA properties	/v1/certificate-authorities/{caId}/properties	✗

Sectigo CA capabilities

See below for the capabilities supported by Entrust Certificate Authorities.

- [enrollments](#)
- [certificates](#)
- [ca-events](#)
- [recoveries](#)
- [subjects](#)
- [status](#)
- [ca-properties](#)

enrollments

See below for the enrollment-supported capabilities

Name	Description	Endpoint	Supported
EnrollmentByCSR	Process certificate signing requests	/v1/certificate-authorities/{caId}/enrollments	✓

Name	Description	Endpoint	Supported
X509CertificateResponse	Returns certificate in X509 form	/v1/certificate-authorities/{caId}/enrollments	✓
PKCS12Response	Return certificate and key in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✓
SANInCSR	Process Subject Alternative Names (SANs) in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
SANInRequest	Process Subject Alternative Names (SANs) in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
SubjectNameInRequest	Supply subject name parameters in request to construct a subject DN in the supplied order	/v1/certificate-authorities/{caId}/enrollments	✓
CAGeneratedKey	Generate the key and returned it to client in PKCS#12 form	/v1/certificate-authorities/{caId}/enrollments	✓
KeyInRequest	Supports client-generated keys for enrollment in the enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
CAGeneratedKeyBackup	Supports key backup for CA generated keys	/v1/certificate-authorities/{caId}/enrollments	✓
ClientGeneratedKeyBackup	Supports key backup for client-generated keys	/v1/certificate-authorities/{caId}/enrollments	✓
ExtensionInCSR	Process extension request in CSR	/v1/certificate-authorities/{caId}/enrollments	✓
ExtensionInRequest	Process extension request in enrollment request	/v1/certificate-authorities/{caId}/enrollments	✓
ValidateProofOfPossession	Validates proof of possession if requested	/v1/certificate-authorities/{caId}/enrollments	✓
ValidityPeriodInRequest	Supports validity period in request	/v1/certificate-authorities/{caId}/enrollments	✓

certificates

See below for the certificate-supported capabilities

Name	Description	Endpoint	Supported
SearchBySerial	Lookup certificate by serial number	/v1/certificate-authorities/{caId}/certificates/{serial}	✓
CertificateAction	List supported actions on a certificate	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓
CertificateRevocationReason	List supported revocation reasons for the certificate RevokeAction	/v1/certificate-authorities/{caId}/certificates/{serial}/actions	✓

ca-events

See below for the event-supported capabilities

Name	Description	Endpoint	Supported
CAEvents	Support CA events	/v1/certificate-authorities/{caId}/ca-events	✓

recoveries

See below for the recovery-supported capabilities

Name	Description	Endpoint	Supported
Recover	Recover certificate by DN	/v1/certificate-authorities/{caId}/recoveries/{dn}	✓

subjects

See below for the subject management-supported capabilities

Name	Description	Endpoint	Supported
SearchBySubjectDN	Lookup certificates by subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}	✓
SubjectDNAction	List supported actions on a subject DN	/v1/certificate-authorities/{caId}/subjects/{dn}/actions	✓

status

See below for the status-supported capabilities

Name	Description	Endpoint	Supported
CAStatus	Check whether the CA is up or down	/v1/certificate-authorities/{cald}/status	✓

ca-properties

See below for the CA property-supported capabilities

Name	Description	Endpoint	Supported
CAPropertiesRetrieval	Retrieve CA properties	/v1/certificate-authorities/{cald}/properties	✓

15 Troubleshooting

See below for troubleshooting CA Gateway issues related to EJBCA enrollment.

- [EJBCA authentication failures](#)
- [EJBCA SSL/TLS Connection Failures](#)
- [EJBCA certificate issuance failures](#)

EJBCA authentication failures

If you see authentication errors when enrolling certificates with an EJBCA certificate authority:

1. Verify the client certificate was issued by ManagementCA.
2. Verify the administrator was added as a member of the role with the correct match criteria.
3. Check that the role has access to the required CAs and End Entity Profiles.

See [Configuring and issuing the EJBCA client certificate](#) for details on each operation.

EJBCA SSL/TLS Connection Failures

If you see SSL/TLS connection errors when enrolling certificates with an EJBCA certificate authority:

1. Verify the trust store file path is correct.
2. Verify the EJBCA hostname matches the SSL certificate.
3. Check that the trust store contains the correct CA certificates.
4. For password-less JKS files, ensure `trust-store-password` is empty or omitted.

See [EJBCA properties](#) for details on the configuration parameters.

EJBCA certificate issuance failures

If certificates fail to issue with an EJBCA certificate authority:

1. Verify the End Entity Profile has access to the requested certificate profile.
2. Verify the certificate profile is configured for the desired key usage and extensions.
3. Check that Subject Alternative Names (SANs) are enabled in the End Entity Profile (if using SANs).
4. Review EJBCA logs for detailed error messages.

16 Integration report

A Gateway is a lightweight, container-based module implementing a CA-agnostic Certificate Lifecycle and Policy Management API. Using CA Gateway, your applications can implement certificate issuance, renewal, and revocation actions across all your Entrust-supported Certification Authorities (CAs). CA Gateway provides policy retrieval capabilities that applications can use to customize API and user-facing dialogs to ensure that certificate actions conform to organizational policies.

- [Supported Certificate Authorities](#)
- [Open-source plugins compatible with CA Gateway](#)
- [Supported Platforms](#)
- [Supported HSM](#)
- [Supported TLS configurations](#)

i CA Gateway supports easy upgrades using container technology. We maintain backward compatibility on the API so you can upgrade CA Gateway without worrying that consuming applications will encounter API problems.

Supported Certificate Authorities

CA Gateway is compatible with the following Certificate Authorities.

Product	Version	Support Notes
AWS ACM CA		CA Plugin for cert operations
Digicert CA	Service	Only supports domain-validated (DV) certificates
EJBCA	Community Edition	
Entrust Certificate Authority (ECA)	8.3	The Events API is not supported
	10.1.1	
	10.2	
Entrust PKIaaS	Service	CSRs must be provided on enrollment
GlobalSign	Service	Only GCC API. Only for testing purposes, not ready for production environments
Microsoft Active Directory Certificate Authority	2019, 2016, 2012 R2	CSRs must be provided on enrollment

Product	Version	Support Notes
Sectigo Certificate Authority	Service	Only supports TLS profiles. S/MIME profiles are not supported

Open-source plugins compatible with CA Gateway

CA Gateway is compatible with the following open-source plugin.

 The support is limited to the CA Gateway interoperation with the plugin.

Plugin	Version	Notes
Entrust CA Gateway Vault Client	N/A	This is an Entrust open-source client for Hashicorp Vault that may be obtained from https://github.com/EntrustCorporation/CSP-CA-Gateway-vault-plugin

Supported Platforms

CA Gateway is distributed and operates as a Docker container. Our objective in leveraging Docker is to allow customers to utilize the broad array of Docker capabilities, features, and plug-in drivers. Entrust will make reasonable efforts to support our CA Gateway product on the customer's chosen Docker deployment.


Platform	Version	Note
Docker	20.x	Operation as a Docker container is supported on all OS platforms supporting Docker.

Supported HSM

CA Gateway on Docker supports the following hardware security modules (HSMs).

Hardware	Client driver	Firmware	Guide
Entrust nShield Connect XC	13.9.0 (FIPS 140-2 Level 3 mode supported)	12.60.15 & 12.60.2	CA Gateway 3.x - Entrust nShield Integration Guide
Entrust nShield 5c	13.9.0	13.2.4	
Entrust nShield 5c 10G	Not supported	Not supported	

Hardware	Client driver	Firmware	Guide
Thales Luna HSM 7	10.8.0	7.7.1-20	CA Gateway 3.x - Thales Luna Integration Guide
Thales TCT	10.8.0	7.7.1-20	

 Log in trustedcare.entrust.com to download the integration guides.

Supported TLS configurations

CA Gateway supports the following TLS versions.

TLS versions	Key Encapsulation Mechanisms (KEMs)
TLSv1.2	None
TLSv1.3	None

CA Gateway provides a [server](#) setting to select the supported ciphersuites. See below for the ones supported by default.

Cipher	TLSv1.2	TLSv1.3
TLS_AES_128_CCM_8_SHA256	✘	✓
TLS_AES_128_CCM_SHA256	✘	✓
TLS_AES_128_GCM_SHA256	✘	✓
TLS_AES_256_GCM_SHA384	✘	✓
TLS_CHACHA20_POLY1305_SHA256	✘	✓
TLS_CHACHA_POLY1305_SHA256	✘	✓
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	✓	✘
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	✓	✘
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	✓	✘

Cipher	TLSv1.2	TLSv1.3
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	✓	✗
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	✓	✗
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	✓	✗
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	✓	✗
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	✓	✗
TLS_RSA_WITH_AES_128_GCM_SHA256	✓	✗
TLS_RSA_WITH_AES_256_GCM_SHA384	✓	✗