



# ENTRUST

## Certificate Hub 4.2.1

### Kubernetes Deployment Guide

**Document issue: 1.0**  
**Issue date: September 4, 2025**

© 2025, 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

|          |                                             |           |
|----------|---------------------------------------------|-----------|
| <b>1</b> | <b>About this guide .....</b>               | <b>6</b>  |
|          | Revision information.....                   | 6         |
|          | Related documentation.....                  | 6         |
|          | Documentation feedback .....                | 6         |
| <b>2</b> | <b>Overview .....</b>                       | <b>7</b>  |
|          | Licensing model .....                       | 7         |
|          | High-level architecture .....               | 7         |
| <b>3</b> | <b>Release Notes.....</b>                   | <b>9</b>  |
|          | New features .....                          | 9         |
|          | Fixed bugs.....                             | 9         |
|          | Known issues.....                           | 9         |
| <b>4</b> | <b>Requirements.....</b>                    | <b>14</b> |
|          | Database requirements .....                 | 14        |
|          | Kubernetes requirements.....                | 15        |
|          | Third-party command-line tools.....         | 18        |
| <b>5</b> | <b>Preparing the deployment .....</b>       | <b>19</b> |
|          | Checking entropy .....                      | 19        |
|          | Setting a local storage in Kubernetes ..... | 19        |
|          | Downloading the installation files .....    | 20        |
|          | Verifying the downloaded files.....         | 22        |
| <b>6</b> | <b>Configuring the deployment .....</b>     | <b>23</b> |
|          | CAGW_TIMEOUT.....                           | 23        |
|          | CERT_HUB_HOSTNAME.....                      | 23        |
|          | CLUSTER_TYPE.....                           | 23        |
|          | DOCKER_REGISTRY .....                       | 24        |
|          | DOCKER_REPOSITORY .....                     | 24        |
|          | IMAGE_PULL_SECRETS_NAME.....                | 24        |
|          | INITIAL_USER .....                          | 24        |

---

|                                                      |           |
|------------------------------------------------------|-----------|
| INITIAL_USER_EMAIL .....                             | 24        |
| INITIAL_USER_PASSWORD .....                          | 24        |
| KUBECTL.....                                         | 24        |
| LATEST_DISCOVERY_SCANNER_VERSION.....                | 25        |
| NAMESPACE.....                                       | 25        |
| POSTGRES_HOST_API .....                              | 25        |
| POSTGRES_PORT .....                                  | 25        |
| POSTGRES_PWD .....                                   | 25        |
| POSTGRES_SSL_ROOT_CRT .....                          | 25        |
| POSTGRES_SSLMODE.....                                | 25        |
| POSTGRES_USER .....                                  | 26        |
| PROXY_EXCLUDE_DOMAINS.....                           | 26        |
| PROXY_HOST .....                                     | 26        |
| PROXY_PORT .....                                     | 26        |
| <b>7 Deploying.....</b>                              | <b>27</b> |
| Loading the images to Docker.....                    | 27        |
| Creating the Kubernetes environment .....            | 27        |
| <b>8 Using the Certificate Manager console .....</b> | <b>31</b> |
| Logging in to the Certificate Manager console .....  | 31        |
| Browsing the Certificate Manager console guide.....  | 31        |
| <b>9 Upgrading to 4.2.1.....</b>                     | <b>32</b> |
| <b>10 Validating the installation .....</b>          | <b>33</b> |
| Checking the process execution.....                  | 33        |
| Checking the online UI.....                          | 33        |
| <b>11 Managing the database .....</b>                | <b>34</b> |
| Backing up the database .....                        | 34        |
| Restoring the database.....                          | 35        |
| Migrating to an external database .....              | 36        |
| <b>12 Managing logs .....</b>                        | <b>38</b> |
| Viewing Certificate Hub logs in Kubernetes.....      | 38        |

|                                                        |           |
|--------------------------------------------------------|-----------|
| Adjusting the acm-api log level .....                  | 38        |
| <b>13 Error reference .....</b>                        | <b>40</b> |
| Authentication and authorization errors .....          | 40        |
| Administration errors.....                             | 42        |
| Automation errors.....                                 | 45        |
| Control errors .....                                   | 49        |
| Certificate errors .....                               | 53        |
| <b>14 Integration report .....</b>                     | <b>62</b> |
| Entrust products compatible with Certificate Hub ..... | 62        |
| Supported Deployment Platforms .....                   | 62        |
| Supported Web Browser .....                            | 62        |
| Databases supported by Certificate Hub .....           | 63        |
| Plugins supported by Certificate Hub .....             | 63        |
| Standards supported by Certificate Hub .....           | 63        |

## 1 About this guide

This document describes how to deploy Certificate Hub 4.2.1 in Kubernetes.

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

### Revision information

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

| Issue | Date     | Section      | Changes                            |
|-------|----------|--------------|------------------------------------|
| 1.0   | Aug 2025 | All sections | The first release of this document |

### Related documentation

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

| Document                                                                                                                                                        | Contents                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <a href="https://api.managed.entrust.com/csp/1.1.0/Using-Certificate-Manager.html">https://api.managed.entrust.com/csp/1.1.0/Using-Certificate-Manager.html</a> | User options of the Certificate Manager console. |

### 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

Certificate Hub has three sets of capabilities:

- The **find capabilities** inventory certificates across your organization (through network discovery) and automated certificate import (from CA databases and cloud services).
- The **control capabilities** centrally manage policy, issuance & access to public and private certificates regardless of vendor. Perform manual operations as necessary to issue, renew, and revoke certificates.
- The **automation capabilities** push keys and certificates to endpoints, with fully managed rotation and certificate profile management.
- The **report capabilities** provide organizational, issue notifications, and reports to remind certificate owners of actions they need to take.

**i** Administrators can customize Certificate Hub to meet enterprise needs like access permissions, system metadata, notifications, or report branding.

- [Licensing model](#)
- [High-level architecture](#)

### Licensing model

Certificate Hub is licensed by capability tier. The find and report tier provides functionalities like the following.

- Certificate discovery with centralized management through the Certificate Hub console
- Automated and customized reporting
- Expiry notifications
- Fixed, per-year subscription
- Plugin management

The control and automate tier provides functionalities like the following.

- Find features
- Single Pane of Glass Authority management
- Certificate manual issuance and renewal
- Automated certificate renewal.
- Certificate lifecycle management
- Push certs to defined Destinations
- Open Plugin Interfaces

**i** When we refer to "you", we mean the customer who has purchased one or more PKIaaS licenses, or one of that customer's internal users, i.e., personnel. In general, the software is licensed for your internal use only. However, you are permitted to assign identities (uniquely identified endpoints) and digital certificates to Users outside your organization solely to enable communications between you and those Users concerning your business.

### High-level architecture

The high-level architecture integrates the following main components.

- [Discovery Scanners](#)
- [Entrust CA Gateway](#)
- [Certificate Hub](#)

## Discovery Scanners

Certificate Hub Discovery Scanners:

- Search your enterprise's networks or portions of networks for the most recent information about deployed certificates.
- Record each certificate's location, type, algorithms, and expiry, regardless of the certificate issuer.

Discovery Scanners are typically deployed on your premises, inside corporate firewalls, to access the internal private servers. However, only Discovery Scanners require this kind of deployment; you can deploy the other Certificate Hub components in a less restrictive environment.

When started, a Discovery Scanner:

1. Contacts Certificate Hub to get the policy and scan configuration.
2. Launches the Certificate Hub scheduling process for scanning.
3. Executes one or more configured scans according to the calendar schedule and priority.
4. Periodically polls Certificate Hub for any policy and or configuration updates.

 Discovery Scanners run a custom-built version of Nmap to scan ports, capture the returned SSL certificate chain, and transmit scan results to Certificate Hub for processing.

## Entrust CA Gateway

Through Entrust CA Gateway, Certificate Hub obtains a direct feed of issued certificates from each supported Certificate Authority (CA).

1. Entrust Authority Security Manager (on-prem and Entrust-managed).
2. Entrust Public Certificate Services (ECS).
3. Microsoft CAs.
4. Entrust PKIaaS.

Thus, Certificate Hub can request certificates from all the CAs managed by a CA Gateway instance.

## Certificate Hub

Certificate Hub is a container-based set of services amenable to either customer premises or commercial cloud hosting. Certificate Hub provides:

- An API interface to the companion Certificate Hub browser UI.
- The underlying certificate database.
- The necessary background processes.



## 3 Release Notes

See below for the Certificate Hub 4.2.1 release notes.

- [New features](#)
- [Fixed bugs](#)
- [Known issues](#)

### New features

This bug-fixing release does not add features.

### Fixed bugs

This release fixes the following bug.

#### 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

This release has the following known issues.

- [Error when taking certificates off hold \(ATEAM-1445\)](#)
- [Issues when changing the display order of custom fields \(ATEAM-15463\)](#)
- [Some endpoint filters display invalid results on report files \(ATEAM-15933\)](#)
- [Certificates without names not synced from source \(ATEAM-16039\)](#)
- [Wildcard certificates not recorded \(ATEAM-16436\)](#)
- [Buttons language not affected when switching language \(ATEAM-16920\)](#)
- [Owner grid column not populated when accessing reports \(ATEAM-16923\)](#)
- [The Verify button does not always trigger a verification \(ATEAM-16950\)](#)
- [The public key ID can refer to the private key ID \(ATEAM-16982\)](#)
- [Error when selecting the Domains widget \(ATEAM-16986\)](#)
- [Archive certificates option missing with FIND licenses \(ATEAM-16988\)](#)
- [Incomplete CA certificate chain \(ATEAM-16997\)](#)
- [Verification fails for IIS destinations \(ATEAM-17030\)](#)
- [Error message on Kubernetes with an external database \(ATEAM-17230\)](#)
- [502 status code when selecting Preview CSV for a report \(ATEAM-18121\)](#)
- [Unexpected Error When Pushing Certificate to SFTP \(ATEAM-18788\)](#)
- ["Weak Hash Algorithm" filter requires resetting before generating a report \(ATEAM-18804\)](#)

#### Error when taking certificates off hold (ATEAM-1445)

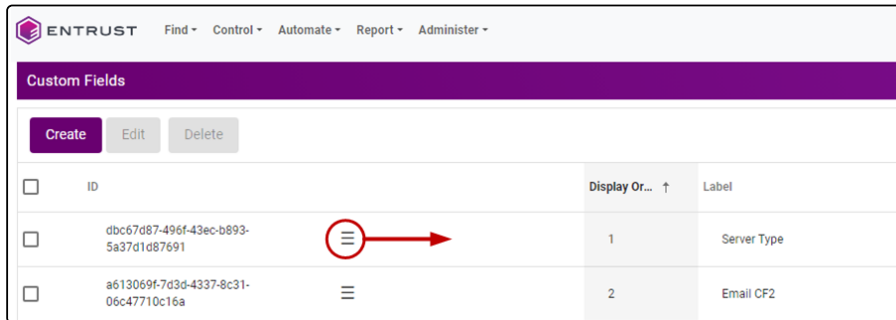
For certificates with **Revocation Reason: On Hold**, attempting to unhold the certificate may fail, or the **Unhold** option may be absent from the **Actions** dropdown.

## Issues when changing the display order of custom fields (ATEAM-15463)

In the **Custom Fields** page of the web console, administrators can change the display order of the custom fields. However:

- Reordering a custom field does not change the **Display Order** column value.
- Refreshing the **Custom Field** page reverts all changes.

**Workaround:** After dragging a custom field to a different position, move the ☰ drag icon within its row to make the changes persistent.



## Some endpoint filters display invalid results on report files (ATEAM-15933)

The following endpoint filters do not display correct results on the downloaded report files.

- is empty
- is not empty

## Certificates without names not synced from source (ATEAM-16039)

Certificates without a name are not successfully synced from Sources.

## Wildcard certificates not recorded (ATEAM-16436)

The application does not record wildcard certificates successfully scanned by the Discovery Scanner.

**i** Wildcard certificates are certificates containing the wildcard asterisk in the issuer and subject.

## Buttons language not affected when switching language (ATEAM-16920)

Switching the language before logging in does not affect the language of the **Delete** and **Cancel** buttons in the **Confirm Delete** popup on the **Destinations** page.

## Owner grid column not populated when accessing reports (ATEAM-16923)

The **Owner** grid column is not populated when accessing the **Report Schedules** from the **Report Designer** grid.

**Workaround:** Access the **Report Schedules** grid from the navigation bar.

## The Verify button does not always trigger a verification (ATEAM-16950)

After the failed verification of a Destination, clicking **Verify** again does not trigger a new verification.

**Workaround:** Make any change in the create form – for example, change the **Description**.

## The public key ID can refer to the private key ID (ATEAM-16982)

When issuing a certificate using the **Key Manager (KMIP)** destination, the **public key ID** is also referring to the **private key ID**

## Error when selecting the Domains widget (ATEAM-16986)

When using a FIND license, selecting the **Domains** widget on the **Dashboard** displays the following error.

```
Unable to show information: Forbidden. This request is not allowed.
```

## Archive certificates option missing with FIND licenses (ATEAM-16988)

When using a FIND license, the option to **Archive** certificates is missing in the **Actions** dropdown on the **Certificates** grid.

## Incomplete CA certificate chain (ATEAM-16997)

When creating a new certificate, the downloaded chain only includes the certificate of the CA that issued the new certificate instead of including the entire chain.

## Verification fails for IIS destinations (ATEAM-17030)

Verification fails for IIS destinations if the username includes a domain name – for example:

```
.\user
```

```
domain\user
```

## Error message on Kubernetes with an external database (ATEAM-17230)

When running on Kubernetes with an external database, you can safely ignore the following error message.

```
Error from server (NotFound): configmaps "postgres-config" not found
```

## 502 status code when selecting Preview CSV for a report (ATEAM-18121)

Certificate Hub can return a 502 status code when exporting a report to CSV. That is, when:

1. Navigating to **Report / Designer**.
2. Selecting the **Design** action for a report.
3. Selecting **Preview CSV** in the menu bar.

**Workaround:**

1. Log in to the machine hosting the appliance or the Kubernetes deployment.
2. Run the following command to edit the `acm-api` configuration file.

```
sudo kubectl edit deployment/acm-api -n certhub
```

3. Add `-Xmx4g` to the `JAVA_OPTS` setting.

```
env:  
  - name: JAVA_OPTS  
    value: -Xmx4g
```

4. Run the following command to make the changes effective.

```
sudo kubectl rollout restart deployment/acm-api -n certhub
```

5. Run the following command to edit the `scheduler` configuration file.

```
run ssudo kubectl edit deployment/scheduler -n certhub
```

6. Add `-Xmx4g` to the `JAVA_OPTS` setting.

```
env:  
  - name: JAVA_OPTS  
    value: -Xmx4g
```

7. Run the following command to make the changes effective.

```
sudo kubectl rollout restart deployment/scheduler -n certhub
```

## Unexpected Error When Pushing Certificate to SFTP (ATEAM-18788)

Pushing a certificate to an SFTP destination may fail with an error message that includes the following:

```
Caused by: java.lang.UnsupportedOperationException
```

**Workaround:** This error may occur if the plugin manages an outdated hash of the destination SSH key. To resolve this issue, please follow these steps:

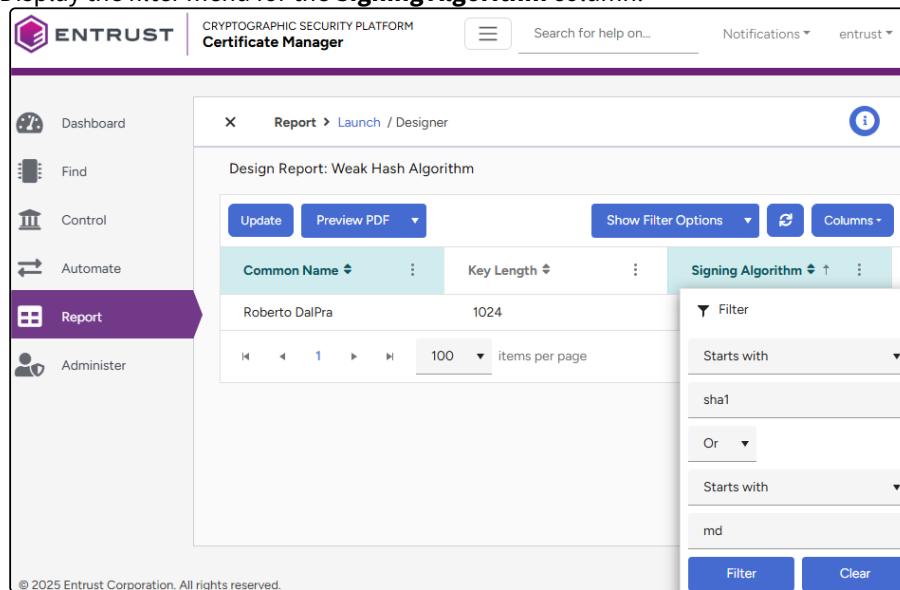
1. Edit the SFTP destination as explained in [Editing a destination](#).
2. In the **Edit** dialog, click **Verify** to force an update of the destination SSH key hash.
3. Ensure the hash displayed by the confirmation message matches the hash of the destination public SSH key.
4. Click **Save** to confirm the changes.

## "Weak Hash Algorithm" filter requires resetting before generating a report (ATEAM-18804)

When using the **Preview** button to export the contents of the **Weak Hash Algorithm** system report, the generated report file includes more certificates than just those with a weak hash algorithm.

### Workaround:

1. Log in to the user console of Certificate Manager.
2. Navigate to **Report > Designer**.
3. On the report grid, click on **Weak Hash Algorithm**.
4. Display the filter menu for the **Signing Algorithm** column.



5. Click **Clear** to remove all the filter settings.
6. Click the **Update** button.
7. Set again the initial filter configuration:

#### Starts with

sha1

#### Or

#### Starts with

md

8. Click the **Update** button.
9. Click **Preview** to confirm the generated report only includes certificates matching the filter.

## 4 Requirements

Certificate Hub has the following requirements.

- [Database requirements](#)
- [Kubernetes requirements](#)
- [Third-party command-line tools](#)

### Database requirements

New Certificate Hub deployments no longer support an internal database. Instead, these deployments use an external database with the following requirements.

- [DBMS](#)
- [Packages](#)
- [Database storage](#)
- [Database permissions](#)
- [Database SSL connection](#)
- [Database names](#)

### DBMS

The external Certificate Hub database must be hosted on the following Database Management System (DBMS).

| DBMS       | Version      |
|------------|--------------|
| PostgreSQL | 15 or higher |

### Packages

Pre-packaged PostgreSQL packages typically include the `postgresql-contrib` subpackage. If not included, install this subpackage to obtain some of the required extensions.

<https://www.postgresql.org/docs/current/contrib.html>

### Database storage

Calculate the required database storage based on the expected certificates and reports. For example, 1G storage is enough for 25,000 certificates and a few weeks of reports.

| Data         | Quantity            | Bytes/Item        | Total  |
|--------------|---------------------|-------------------|--------|
| Certificates | 25,000 certificates | 20 KB/certificate | 500 MB |
| Reports      | 200 reports         | 1 MB/report       | 200 MB |
|              |                     |                   | 700 MB |

## Database permissions

To create an external database user with sufficient permissions, connect to PSQL using the default PostgreSQL user and execute the following commands.

```
CREATE USER ${POSTGRES_USER} WITH NOSUPERUSER CREATEDB ENCRYPTED PASSWORD '${POSTGRES_PWD}';  
\c postgres ${POSTGRES_USER}  
CREATE DATABASE certhub;  
\c certhub ${POSTGRES_USER}  
CREATE EXTENSION IF NOT EXISTS pg_trgm;
```

Where:

- `${POSTGRES_USER}` is the value of the `POSTGRES_USER` configuration setting.
- `${POSTGRES_PWD}` is the value of the `POSTGRES_PWD` configuration setting.

 See [Configuring the deployment](#) for a description of both settings.

## Database SSL connection

Certificate Hub only supports SSL-protected connections with the PostgreSQL database.

## Database names

Database names should not use uppercase letters to avoid case sensitivity problems. Unquoted identifiers in SQL syntax are converted to lowercase, which can lead to problems when mapping to a name with uppercase letters.

## Kubernetes requirements

Certificate Hub runs on a Kubernetes system with the following requirements.

- [Kubernetes versions](#)
- [Kubernetes worker nodes](#)
- [Kubernetes pod instances](#)
- [Kubernetes Ingress](#)
- [Kubernetes persistent volume](#)
- [Third-party command-line tools](#)

## Kubernetes versions

Certificate Hub makes use of fairly generic Kubernetes capabilities. We endeavor to be as generic as possible in our deployment, and we do not require any special Operating System, Kubernetes, or Docker configuration that we are aware of.

- To our knowledge, Certificate Hub use of Kubernetes features is confined to open-source capabilities. Certificate Hub is supported on any active Kubernetes open-source version or commercial derivative.
- We officially support Certificate Hub operation on the current and prior two Kubernetes versions, consistent with generally accepted Kubernetes support policies. However, we have operated Certificate Hub on versions since 2018.

## Kubernetes worker nodes

The number of Kubernetes worker nodes to use is a choice for our customers and depends on Certificate Hub and the other applications sharing the Kubernetes cluster. In our AWS environment, we run multiple instances of Certificate Hub, each in its namespace. Five nodes are sufficient for our roughly 20 development, quality assurance, and special-use instances. Not sure if these numbers are still valid. CertHub has grown, and we've had to scale back the number of environments we host.

## Kubernetes pod instances

The Kubernetes deployment of Certificate Hub consists of the following pods.

| Pod            | Description                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entry          | The external entry point into the Certificate Hub application. Certificate Hub exposes this pod as a service. You must point to a suitable Ingress at this service.                                         |
| Internal API   | The heart of Certificate Hub implementing all new Certificate Hub features. This pod is implemented in Java and runs inside an embedded Tomcat instance.                                                    |
| CertHub API    | This component provides the externally accessible CertHub REST API.                                                                                                                                         |
| Lemur          | The ephemeral pod migrating and updating the legacy database.                                                                                                                                               |
| PostgreSQL     | An off-the-shelf PostgreSQL docker image.                                                                                                                                                                   |
| Notification   | The ephemeral pod periodically invoking the plugin automation script for the notification plugin to send out notification emails on impending expirations.                                                  |
| Flyway Scripts | The ephemeral pod containing the Flyway tool and bespoke scripts. This simple run-once container connects to the Certificate Hub database, applies the scripts, and is not required again until an upgrade. |
| User Create    | The ephemeral pod bootstrapping initial users into the database.                                                                                                                                            |
| Role Update    | The ephemeral pod bootstrapping initial roles into the database.                                                                                                                                            |
| UI             | This pod hosts the static UI assets.                                                                                                                                                                        |

In most deployments, one instance of each pod will be sufficient to handle the load. Certificate Hub load is proportional to the number of active users. Background tasks, such as discovery, source certificate retrieval, reporting, and renewal, do not constitute a heavy load.



The deployment scripts instantiate a single instance of each pod. You can modify this in the Kubernetes deployment by updating the replicas settings in the following file.

```
acm-deployment.yaml
```

However, there are many other considerations for high availability.

## Kubernetes Ingress

Certificate Hub assumes that your Kubernetes deployment includes an Ingress Controller. You must point this controller to the Entry Service, for example, by simply defining an Ingress. The Certificate Hub installation script will create an Ingress entry with the following name.

```
certhub-ingress
```

That should work in most cases. But, depending on your type of Ingress Controller, you may need to replace or modify this Ingress.

Certificate Hub expects all URL paths prefixed with a namespace. Traditionally this namespace is the Kubernetes Namespace of the Certificate Hub installation, but you can use any string that contains only letters and numbers. Certificate Hub will not function without this namespace prefix. The Ingress installed by the installation scripts defines a path with such a namespace prefix.

 Your Ingress Controller must handle TLS termination. Certificate Hub does not handle TLS termination.

## Kubernetes persistent volume

Set a persistent volume for the database where Certificate Hub saves the configuration, certificate data, and reports.

- In AWS deployments, you can generally resize this volume. Kubernetes provides volume expansion support, which we have verified to be effective in our Entrust AWS deployments of Certificate Hub.
- On Kubernetes platforms such as Azure, the storage setup templates may need to be modified.
- In other deployments, like on-premises deployments or deployments using stock PostgreSQL, set the volume size in advance because you cannot resize the database.

For example, a PersistentVolume for 1G is enough storage for 25,000 certs and a few weeks of reports.

| Data         | Quantity            | Bytes/Item        | Total  |
|--------------|---------------------|-------------------|--------|
| Certificates | 25,000 certificates | 20 KB/certificate | 500 MB |
| Reports      | 200 reports         | 1 MB/report       | 200 MB |
|              |                     |                   | 700 MB |

**i** Report size is highly variable, and reports will accumulate quickly if you run them daily. It is probably best to be conservative here. By default, saved reports will be removed after one year. You can remove them earlier by changing the retention value in the report settings.

#### To set the persistent volume

1. Define a persistent volume with enough storage size when [Setting a local storage in Kubernetes](#).
2. Provide persistent volume claim size accordingly when [Creating the Kubernetes environment](#).

## Third-party command-line tools

Certificate Hub assumes the following third-party command-line tools are already installed.

| Tools    | Usage                                                                   |
|----------|-------------------------------------------------------------------------|
| htpasswd | Manage usernames and passwords of HTTP users                            |
| kubectl  | Manage Kubernetes clusters                                              |
| OpenSSL  | Generate application secrets like the Java Web Token (JWT) signing key. |

## Third-party command-line tools

Certificate Hub assumes the following third-party command-line tools are already installed.

| Tools    | Usage                                                                   |
|----------|-------------------------------------------------------------------------|
| htpasswd | Manage usernames and passwords of HTTP users                            |
| kubectl  | Manage Kubernetes clusters                                              |
| OpenSSL  | Generate application secrets like the Java Web Token (JWT) signing key. |

## 5 Preparing the deployment

Prepare the Certificate Hub deployment as explained in the following sections.

- [Checking entropy](#)
- [Setting a local storage in Kubernetes](#)
- [Downloading the installation files](#)
- [Verifying the downloaded files](#)

### Checking entropy

Certificate Hub data encryption operations require a robust entropy source to improve randomness. 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 has not enough entropy, and you must install the `rngd` daemon.

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

**i** Run the above commands in the host machine because the Docker containers use the entropy provided by the host machine.

### Setting a local storage in Kubernetes

Create the persistent volumes defined in the following file.

```
cluster/local-persistent-volume.yaml.template
```

Specifically, this template defines the following persistent volumes.

| Persistent volume         | Contents                                                                                                | Size                 |
|---------------------------|---------------------------------------------------------------------------------------------------------|----------------------|
| postgres-pv-volume-<node> | PostgreSQL                                                                                              | Selected on creation |
| plugins-pv-volume         | Plugins downloaded from <a href="https://trustedcare.entrust.com">https://trustedcare.entrust.com</a> . | 1 Gigabyte           |

To create both persistent volumes, run the following script.

```
./acmPreReqsWithLocalStorage.sh --nodes <nodes> -r <repository> --pv-volume <volume>
```

See below for the supported options.

- `-n, --nodes <nodes>`
- `--pv-volume <volume>`
- `-r, --repo <repository>`

To check that the persistent volumes are available, run:

```
kubectl get pv
```

### `-n, --nodes <nodes>`

Make persistent disks for PostgreSQL on the Kubernetes nodes with the `<nodes>` hostnames.

### `--pv-volume <volume>`

Allocate a persistent volume of `<volume>` (for example, 10Gi). This volume is for the database. It does not include the 1 Gigabyte for the plugin's persistent volume.

 See [Requirements](#) on calculating the required size of the persistent volume for the database.

### `-r, --repo <repository>`

Use the `<repository>` repository of the nginx image – for example, [quay.io](#)

 This repository option will often be different from the one you set up for the Certificate Hub images.

## Downloading the installation files

Download the Certificate Hub installation files for Docker.

### To download the installation files

1. Log in to <https://trustedcare.entrust.com>
2. Go to **PRODUCTS > PKI > Certificate Hub**.
3. Click on the Certificate Hub version you want to download.
4. Select the **SOFTWARE DOWNLOADS** tab to download the installation files, as explained below.
  - [Downloading the Docker images](#)
  - [Downloading the deployment scripts](#)
5. Select the **DOCUMENTS** tab to download the product documentation.

## Downloading the Docker images

On the **SOFTWARE DOWNLOADS** tab, click the **Download** link for the image files.

 You must download and deploy the entire set of images.

| Title                | File Name                                       | Image                       | Notes                          |
|----------------------|-------------------------------------------------|-----------------------------|--------------------------------|
| Entry                | cm-entry-<version>.docker.tar.gz                | cm/cm-entry                 |                                |
| Internal API         | cm-api-<version>.docker.tar.gz                  | cm/api                      | Added in Certificate Hub 2.0   |
| Certificate Hub API  | cm-external-api-<version>.docker.tar.gz         | cm/external-api             |                                |
| Approval             | cm-approval-<version>.docker.tar.gz             | cm/approval                 | Added in Certificate Hub 4.1.0 |
| DB Schema Management | cm-db-schema-management-<version>.docker.tar.gz | cm/acm-db-schema-management |                                |
| UI                   | cm-ui-<version>/docker.tar.gz                   | cm/ui                       | Added in Certificate Hub 2.0   |

Entrust also distributes the Discovery Scanner through [trustedcare.entrust.com](https://trustedcare.entrust.com), along with the Certificate Hub images.

| Title             | File Name                                    | Image                | Notes                                                               |
|-------------------|----------------------------------------------|----------------------|---------------------------------------------------------------------|
| Discovery Scanner | cm-discovery-scanner-<version>.docker.tar.gz | cm/discovery-scanner | The Discovery Scanner 2.1 collects additional endpoint information. |

## Downloading the deployment scripts

On the **SOFTWARE DOWNLOADS** tab, click the **Download** link for **Kubernetes Configuration Scripts**. This compressed file contains the following installation scripts.

| Script                                     | Use                                                   |
|--------------------------------------------|-------------------------------------------------------|
| cluster/acmPreReqsWithLocalStorage.sh      | <a href="#">Setting a local storage in Kubernetes</a> |
| environment/config.sh                      | <a href="#">Configuring the deployment</a>            |
| environment/environmentCreationOrUpdate.sh | <a href="#">Creating the Kubernetes environment</a>   |
| environment/loadImagesToDocker.sh          | <a href="#">Loading the images to Docker</a>          |

| Script               | Use                                   |
|----------------------|---------------------------------------|
| environment/dbctl.sh | <a href="#">Managing the database</a> |

The compressed file also contains configuration files evaluated by the installation scripts. For example, the `acmPreReqsWithLocalStorage.sh` script evaluates the following configuration files.

| Configuration file                            | Settings                                   |
|-----------------------------------------------|--------------------------------------------|
| cluster/local-persistent-volume.yaml.template | Kubernetes persistent volume provisioning. |
| cluster/localstorageclass.yaml                | Local storage class definition.            |

The `yaml.template` files contain placeholders for evaluating the `config.sh` parameters. The installation scripts source these parameters and deploy the `yaml.template` files as `yaml`.

## 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.

## 6 Configuring the deployment

To configure Certificate Hub in Kubernetes, edit the `config.sh` file and configure the following static parameters.

- `CAGW_TIMEOUT`
- `CERT_HUB_HOSTNAME`
- `CLUSTER_TYPE`
- `DOCKER_REGISTRY`
- `DOCKER_REPOSITORY`
- `IMAGE_PULL_SECRETS_NAME`
- `INITIAL_USER`
- `INITIAL_USER_EMAIL`
- `INITIAL_USER_PASSWORD`
- `KUBECTL`
- `LATEST_DISCOVERY_SCANNER_VERSION`
- `NAMESPACE`
- `POSTGRES_HOST_API`
- `POSTGRES_PORT`
- `POSTGRES_PWD`
- `POSTGRES_SSL_ROOT_CRT`
- `POSTGRES_SSLMODE`
- `POSTGRES_USER`
- `PROXY_EXCLUDE_DOMAINS`
- `PROXY_HOST`
- `PROXY_PORT`

**i** After creating the Certificate Hub environment in Kubernetes, you can still modify the proxy settings by editing the configuration maps with `kubectl` or similar.

### CAGW\_TIMEOUT

The timeout value for the CA Gateway calls, in seconds.

**Mandatory:** No. When omitted, this value defaults to 60 seconds.

### CERT\_HUB\_HOSTNAME

The base hostname of the Kubernetes Ingress routing to the application.

**Mandatory:** Yes.

### CLUSTER\_TYPE

Make sure this parameter has an empty value.

```
CLUSTER_TYPE= ' '
```

You can also simply omit the parameter.

**Mandatory:** No.

## DOCKER\_REGISTRY

The registry from which to pull the private Entrust Docker images.

**Mandatory:** Yes.

## DOCKER\_REPOSITORY

The name in the Docker registry of the repository for pushing the Certificate Hub images. This name will be prepended as a directory to all the images. It may contain multiple path components. In an OpenShift environment, this equates to a project.

**Mandatory:** Yes.

## IMAGE\_PULL\_SECRETS\_NAME

The name of a preconfigured imagePullSecrets registry credential to use for the Entrust images.

**Mandatory:** No.

## INITIAL\_USER

The username of the initial administrator.

**Mandatory:** Yes.

## INITIAL\_USER\_EMAIL

The email address of the initial administrator.

**Mandatory:** Yes.

## INITIAL\_USER\_PASSWORD

A temporary password for the initial administrator. Ensure that this temporary password does not include special characters such as '#', '!', or '\*'.

 After the initial login, the administrator will be prompted to create a new password that meets a specific set of password strength requirements.

**Mandatory:** Yes.

## KUBECTL

The Kubernetes command-line tool. This parameter is handy in environments like OpenShift with a different command name.

**Mandatory:** No. This optional value defaults to kubectl.



## LATEST\_DISCOVERY\_SCANNER\_VERSION

The version identifier communicated to the Discovery Scanner instances. The instances will compare this value with the local version and show a local warning if an upgrade is available.

**Mandatory:** Yes.

## NAMESPACE

The Kubernetes namespace to deploy Certificate Hub under. The application will be available at:

```
https://<CERT_HUB_HOSTNAME>/<NAMESPACE>
```

**Mandatory:** Yes.

## POSTGRES\_HOST\_API

The host of an external PostgreSQL database.

**Mandatory:** No. When omitting this value, Certificate Hub uses an internal database instead.

## POSTGRES\_PORT

The connection port with an external PostgreSQL database.

**Mandatory:** When using an external database.

## POSTGRES\_PWD

The user password of the internal or external database.

**Mandatory:** Yes.

## POSTGRES\_SSL\_ROOT\_CRT

The path of the file that contains the root CA certificate for connecting with an external PostgreSQL database.

**Mandatory:** When using an external database and `POSTGRES_SSLMODE` is `verify-ca` or `verify-full`.

## POSTGRES\_SSLMODE

The SSL mode for connecting with an external PostgreSQL database. Supported values are:

- `require`
- `verify-ca`
- `verify-full`

See <https://www.postgresql.org/docs/current/libpq-ssl.html> for a description of each mode.



Any of the supported PostgreSQL modes requires enabling SSL.

**Mandatory:** When using an external database.

## POSTGRES\_USER

The user name of the internal or external database.

**Mandatory:** Yes.

## PROXY\_EXCLUDE\_DOMAINS

The external hosts you want to access without the proxy, as a comma-separated list of IP addresses or domain names.

 This field does not support wildcards.

**Mandatory:** No. When omitting this value, Certificate Hub does not use a proxy.

## PROXY\_HOST

The IP or domain name of the proxy server.

**Mandatory:** No. When omitting this value, Certificate Hub does not use a proxy.

## PROXY\_PORT

The port number of the proxy server (if any).

**Mandatory:** No. When omitting this value, Certificate Hub does not use a proxy.

## 7 Deploying

Deploy Certificate Hub as explained below.

- [Loading the images to Docker](#)
- [Creating the Kubernetes environment](#)

### Loading the images to Docker

The deployment script assumes that the required [Docker images](#) are in a private Docker registry accessible by Kubernetes. If you do not log all nodes onto the Docker registry, you can configure imagePullSecrets when [Configuring Certificate Hub in Kubernetes](#).

You will need to manually load these images or run the following script provided with the [deployment scripts](#).

```
./loadImagesToDocker.sh -r <docker_registry> -n <docker_repository> [-a]
```

When executed, the script prompts for the location of the directory containing the Certificate Hub images. Press **ENTER** to use the parent directory. The directory path should not contain any spaces.

#### -a, --auth

Load the registry authentication from:

```
$HOME/.docker/config.json
```

When this file is not present, the script prompts the username and password. Skip this option if the docker client has already logged in on the registry.

#### -n, --docker repository <docker\_repository>

Push the Certificate Hub images to the `<docker_repository>` repository of the Docker registry. If omitted, the script assumes the `DOCKER_REPOSITORY` value set when [Configuring the deployment](#).

#### -r, --docker-registry <docker\_registry>

Push the Certificate Hub images to the `<docker_registry>` Docker registry. If omitted, the script assumes the `DOCKER_REGISTRY` value set when [Configuring Certificate Hub in Kubernetes](#).

### Creating the Kubernetes environment

To install or update Certificate Hub in Kubernetes, run the following script provided with the deployment scripts.

```
environmentCreationOrUpdate.sh
```

You can use this script as-is or modify it to fit your Kubernetes policies. The script will run through the flow below.

1. Source the parameters described in [Configuring the deployment](#).
2. Create the Kubernetes YAML files from the corresponding templates by replacing parameters.

3. Create the environment namespace.
4. Generate (optionally regenerate) the application secrets.
5. Deploy PostgreSQL 11 from public Docker. If you have a private PostgreSQL 11 image that you prefer to use, point to a private registry in the following template distributed with the Deployment scripts.

```
/environment/postgres/postgres-deployment.yaml
```

6. Wait until the PostgreSQL deployment completes.
7. Create or update the schema.
8. Deploy the Java ACM-API application.
9. Deploy the Ingress definitions.

To deploy a new environment, run the following command.

```
./environmentCreationOrUpdate.sh --tag <tag> --create --password <password> --  
postgres-pwd <postgres-pwd> --smtp-pwd <smtp-pwd> --pvc-storage <pvc-storage>
```

To upgrade a deployment, run the following command.

```
./environmentCreationOrUpdate.sh --tag <tag>
```

**i** Intellitrust is now "Entrust Identity as a Service (IDaaS)", and the redirect URL changed after the upgrade 2.0.1. Thus, if you added Intellitrust as an Identity provider in Certificate Hub, you must update the IDaaS to use the new redirect URL.

As detailed in the following sections, the script prompts for the value of some parameters when omitted.

- -a, --allow-untrusted-CAGW
- --add-namespace
- -c, --create
- -g, --generate-JWT-secret
- -k, --encryption-key-file <key-file>
- -n, --namespace <namespace>
- --password <password>
- --postgres-user <postgres-user>
- --postgres-pwd <postgres-pwd>
- --pvc storage <size>
- -t, --tag <tag>

## -a, --allow-untrusted-CAGW

Accept self-signed or private CA Gateway server certificates when consuming CA Gateway's HTTPS API.

**⚠** This option is for testing purposes, not for production mode. Will make the system vulnerable to Man in the Middle attacks if added.

Certificate Hub supports uploading separate trust stores with each CA Gateway and Source Configuration.

## --add-namespace

Generate the namespace environment when using the `--create` option. To set the namespace, you can either:

- Use the `--namespace` option.
- Set the `NAMESPACE` parameter when [Configuring the deployment](#).

Omit this option if the namespace already exists.

## -c, --create

On first execution, trigger the generation of the secrets and the database schema.

**i** If you omit this option on the first execution, delete the namespace and run again.

## -g, --generate-JWT-secret

Trigger the regeneration of the JWT secret for client session authentication. This advanced switch invalidates all existing sessions and is rarely required.

## -k, --encryption-key-file <key-file>

Use the key in the `<key-file>` file to encrypt the sensitive data of the databases. When omitting this option, the Certificate Hub deployment generates a random encryption key.

**i** If you use the embedded database option, the backup and restore operations will include the encryption key. You should only consider generating your key if you use an external database and external services for the backup and restore operations.

The `<key-file>` file must contain the base-64 encoding of a 32-byte-long key. Save the file without the newline character, for example:

```
echo -n "This is a custom encryption key." | base64 > encryption.key
```

You can also generate a securely random encryption key using:

```
openssl rand -base64 32 > encryption.key
```

## -n, --namespace <namespace>

Set the `<namespace>` environment namespace when using the `--add-namespace` option.

## --password <password>

Set `<password>` as the password of the initial administrative user. Some special characters could fail the user creation job without quoting or escaping the password. After the first login, you will be redirected to the **Change**

**Password** page for replacing this temporary one-time password with a password meeting the password strength requirements.

**Default value:** Prompted to the user.

**--postgres-user <postgres-user>**

Authenticate in the Certificate Hub database with the `<postgres-user>` user.

**Default value:** `postgres`

**--postgres-pwd <postgres-pwd>**

Authenticate in the Certificate Hub database with the `<postgres-pwd>` password.

**Default value:** Prompted to the user.

**--pvc storage <size>**

Allocate `<size>` for the Kubernetes Persistent Volume claim storage. Where `<size>` is:

- A plain integer
- A fixed-point number using one of these suffixes: E, P, T, G, M, K.
- The power-of-two equivalents: Ei, Pi, Ti, Gi, Mi, Ki.

See <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers>

**Default value:** 1Gi

**-t, --tag <tag>**

Use the `<tag>` tag version of the Entrust Docker containers.

 When upgrading, the provided tag must be different from the tag of the current deployment. Kubernetes will detect a difference and deploy the new version.

**Default value:** `latest`

## 8 Using the Certificate Manager console

After deploying Certificate Hub, you can use the Certificate Manager to manage certificate issuance and lifecycle.

- [Logging in to the Certificate Manager console](#)
- [Browsing the Certificate Manager console guide](#)

### Logging in to the Certificate Manager console

See below for opening a session in the Certificate Manager console of Certificate Hub.

#### To log in to the Certificate Manager console

1. Open a web browser at the following URL.

```
https://<CERT_HUB_HOSTNAME>/certhub
```

Where `<CERT_HUB_HOSTNAME>` is the value of the `CERT_HUB_HOSTNAME` parameter described in [Configuring the deployment](#).

2. Authenticate using the `INITIAL_USER` and `INITIAL_USER_PASSWORD` credentials described in [Configuring the deployment](#).

### Browsing the Certificate Manager console guide

The Certificate Manager console user guide is published as part of the Cryptographic Security Platform documentation at:

<https://api.managed.entrust.com/csp/1.1.0/Using-Certificate-Manager.html>

## 9 Upgrading to 4.2.1

For upgrading your Certificate 4.1.0 installation to 4.2.1, upgrade the Certificate Hub Docker image as follows.

### To upgrade the Certificate Hub image to 4.2.1

1. Push to Docker the images described in [Downloading the installation files](#).
2. Set the new configuration as explained in [Configuring the deployment](#).
3. Run the `environmentCreationOrUpdate.sh` script as follows.

```
./environmentCreationOrUpdate.sh --tag 4.2.1
```



## 10 Validating the installation

Validate the Certificate Hub installation deployed or updated on Kubernetes.

- [Checking the process execution](#)
- [Checking the online UI](#)

### Checking the process execution

Run the following command.

```
kubectl get pods -n <NAMESPACE>
You should get something like the following
```

| NAME                         | READY | STATUS    | RESTARTS | AGE |
|------------------------------|-------|-----------|----------|-----|
| acm-api-564d85889d-ttj6x     | 1/1   | Running   | 0        | 3m  |
| cm-entry-78784c6b44-f7kk5    | 1/1   | Running   | 0        | 3m  |
| external-api-9cbc67d6b-7scdt | 1/1   | Running   | 0        | 3m  |
| flyway-5pch9                 | 0/1   | Completed | 0        | 3m  |
| lemur-864d6fdd6-qhsjj        | 0/1   | Completed | 0        | 3m  |
| postgres-6dc7bb76db-tx9jh    | 1/1   | Running   | 0        | 3m  |
| role-update-vxvnf            | 0/1   | Completed | 0        | 3m  |
| ui-556fd4dc7d-2cjx2          | 1/1   | Running   | 0        | 3m  |

Optionally, run the following command to free space by deleting completed pods and any associated stopped container.

```
kubectl delete pod -n <namespace> --field-selector=status.phase=Succeeded
```

Check that the following persistent processes are running:

- cm-entry
- external-api
- ui
- acm-api
- postgres

Other pods are ephemeral.

### Checking the online UI

Open your browser in the URL defined when [Configuring the deployment](#).

```
https://CERT_HUB_HOSTNAME/NAMESPACE
```

You should see the Certificate Hub login page.

## 11 Managing the database

See below for the main database management operations.

- [Backing up the database](#)
- [Restoring the database](#)
- [Migrating to an external database](#)

**i** To backup and restore the database encryption key, the below operations use the `dbctl.sh` script obtained when [Downloading the installation files](#).

### Backing up the database

See below for backing up the Certificate Hub database.

- [Vacuuming the database](#)
- [Backing up the database contents](#)
- [Backing up the database encryption key](#)

### Vacuuming the database

Run this command once before any backup to release orphaned pages from the database and decrease its size.

```
sudo dbctl.sh vacuum -n <namespace> [--kubectl-cmd <kubectl-cmd>]
```

Where each parameter has the value described below.

| Option        | Value                                                              | Mandatory                                     |
|---------------|--------------------------------------------------------------------|-----------------------------------------------|
| <namespace>   | The namespace of the Certificate Hub instance                      | Always                                        |
| <kubectl-cmd> | The name of the <code>kubectl</code> client or equivalent command. | When the name of the command is not "kubectl" |

For example:

```
sudo ./dbctl.sh vacuum -n cm --postgres-user postgres
```

### Backing up the database contents

Back up the Certificate Hub database using the tools provided by the DBMS.

### Backing up the database encryption key

Run the following command to back up the database encryption key.

 Back up the databases regularly to restore your data in case of disaster recovery.

```
sudo dbctl.sh backup -n <namespace> [--kubectl-cmd <kubectl-cmd>]
```

Where each parameter has the value described below.

| Option        | Value                                                              | Mandatory                                     |
|---------------|--------------------------------------------------------------------|-----------------------------------------------|
| <namespace>   | The namespace of the Certificate Hub instance                      | Always                                        |
| <kubectl-cmd> | The name of the <code>kubectl</code> client or equivalent command. | When the name of the command is not "kubectl" |

For example:

```
sudo ./dbctl.sh backup -n cm --postgres-user postgres
```

## Restoring the database

To restore the Certificate Hub database, follow the steps below in the same Certificate Hub version used when [Backing up the database](#).

- [Restoring the database contents](#)
- [Restoring the database encryption key](#)
- [Completing the database restoration](#)

### Restoring the database contents

Restore the Certificate Hub database using the Database Management System (DBMS) tools.

### Restoring the database encryption key

Run the following command to restore the database encryption key.

```
sudo dbctl.sh restore -n <namespace> --backup-file <backup-file> [--kubectl-cmd <kubectl-cmd>]
```

Where each parameter has the value described below.

| Option        | Value                                         | Mandatory |
|---------------|-----------------------------------------------|-----------|
| <namespace>   | The namespace of the Certificate Hub instance | Always    |
| <backup-file> | The path of the backup file.                  | Always    |

| Option        | Value                                                              | Mandatory                                     |
|---------------|--------------------------------------------------------------------|-----------------------------------------------|
| <kubectl-cmd> | The name of the <code>kubectl</code> client or equivalent command. | When the name of the command is not "kubectl" |

For example:

```
sudo ./dbctl.sh restore -n cm --backup-file backup-06_19_2023.tar.gz.gpg
```

 Before running this command, you can ignore or delete the `user-creation` and `role-update` jobs in ERROR state.

## Completing the database restoration

Redeploy Certificate Hub to make effective the restoration of the database encryption key.

## Migrating to an external database

Only Certificate Hub installations migrated from older versions support an internal database—that is, a database running on the same host as the application. See below for switching from an internal database to an external one.

- [Exporting the database contents](#)
- [Restoring the database contents](#)
- [Applying the database changes to Certificate Hub](#)

 Run the steps below in the same version of Certificate Hub that will use the database.

## Exporting the database contents

Set the following environment variables in the host running the internal database.

| Variable     | Value                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------|
| KUBECTL      | The value of the <code>KUBECTL</code> parameter described in <a href="#">Configuring the deployment</a> .      |
| NAMESPACE    | The value of the <code>NAMESPACE</code> parameter described in <a href="#">Configuring the deployment</a> .    |
| POSTGRES_DB  | <code>certhub</code>                                                                                           |
| POSTGRES_PWD | The value of the <code>POSTGRES_PWD</code> parameter described in <a href="#">Configuring the deployment</a> . |

| Variable      | Value                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------|
| POSTGRES_USER | The value of the <code>POSTGRES_USER</code> parameter described in <a href="#">Configuring the deployment</a> . |

Run the following commands.

```
now=$(date +"%m_%d_%Y")
```

```
$KUBECTL exec deployment/postgres -n "$NAMESPACE" -- bash -c "export  
PGPASSWORD=$POSTGRES_PWD; pg_dump -d $POSTGRES_DB -U $POSTGRES_USER" > "db-backup-  
$now.sql"
```

## Restoring the database contents

Perform the following steps on an external host to restore the database contents.

### To restore the database contents on an external host

1. Install the PostgreSQL version described in [Database requirements](#).
2. Copy the `db-backup-<date>.sql` backup file generated when [Exporting the database contents](#).
3. Set the following environment variables to the values described in [Exporting the database contents](#).
  - `POSTGRES_DB`
  - `POSTGRES_PWD`
  - `POSTGRES_USER`
4. Run the following command.

```
export PGPASSWORD=$POSTGRES_PWD; psql -d $POSTGRES_DB -U $POSTGRES_USER -f db-  
backup-<date>.sql
```

## Applying the database changes to Certificate Hub

Complete the database migration by performing the following operations on the Certificate Manager host.

### To apply the database changes to Certificate Hub

1. Update the database configuration parameters described in [Configuring the deployment](#) to match the new external database.
2. Redeploy Certificate Hub to make the changes effective.
3. Wait while the redeployment with the external database is complete.
4. Manually remove the PostgreSQL pod.

## 12 Managing logs

In the `acm-api` pod, Certificate Hub has a main container writing logs with an SLF4J logger.

- [Viewing Certificate Hub logs in Kubernetes](#)
- [Adjusting the acm-api log level](#)

### Viewing Certificate Hub logs in Kubernetes

List the pod names:

```
kubectl get pods -n <namespace>
```

Where `<namespace>` is the name of the namespace. For example:

- `qalatest`
- `dev`

View logs on Discovery, Reports, Administrators, Certificates, Sources, or Destinations:

```
kubectl logs <pod_name> -n <namespace>
```

Where `<pod_name>` is the pod name with the following prefix:

```
acm-api-
```

### Adjusting the acm-api log level

Edit the ConfigMap. You can use the tool provided by your platform or run the following command.

```
kubectl -n <namespace> edit cm acm-api-config
```

Append the log level under `jpa.hibernate.ddl-auto: none\n`.

```
"\ jpa.hibernate.ddl-auto: none\nlogging:\n  level:\n    root: <log_level>"
```

Where `<log_level>` is one of the following:

- `ERROR`
- `WARN`
- `INFO`
- `DEBUG`
- `TRACE`

Run the following command.

```
kubectl -n <namespace> rollout restart deployment acm-api
```

## 13 Error reference

When executed, Certificate Hub can print the following errors.

- [Authentication and authorization errors](#)
- [Administration errors](#)
- [Automation errors](#)
- [Control errors](#)
- [Certificate errors](#)

 See the Certificate Hub user guide for how to browse the audit logs.

### Authentication and authorization errors

The application throws the following authentication and authorization errors.

| Code     | Message                                                                                                                               |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1006 | Failed to hash the password for user: <code>&lt;Username&gt;</code>                                                                   |
| ERR_1010 | <code>hasPermission</code> unexpectedly invoked for <code>&lt;Permission&gt;</code>                                                   |
| ERR_1011 | The util command must have a <code>--cmd</code> argument.                                                                             |
| ERR_1012 | Unknown command <code>&lt;Command&gt;</code>                                                                                          |
| ERR_1013 | <code>--username</code> , <code>--password</code> , and <code>--email</code> must be supplied to the <code>createUser</code> command. |
| ERR_1014 | Unexpected crypto error:                                                                                                              |
| ERR_1015 | Error creating default cert expiry rule for initial user:                                                                             |
| ERR_1016 | <code>--username</code> and <code>--role</code> must be provided.                                                                     |
| ERR_1017 | Unexpected crypto exception:                                                                                                          |



| Code         | Message                                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_104<br>0 | Unexpected parsing error while loading auth request:                                                                                         |
| ERR_104<br>1 | Unexpected parsing error while saving auth request:                                                                                          |
| ERR_104<br>2 | Unexpected parsing error while removing auth request:                                                                                        |
| ERR_104<br>6 | Could not find password auth provider entry.                                                                                                 |
| ERR_104<br>7 | Failed to hash the password for user: <code>&lt;Username&gt;</code>                                                                          |
| ERR_104<br>8 | Cannot update non-existent user. User must have existing id.                                                                                 |
| ERR_104<br>9 | Login denied. Tenant id not found for user <code>&lt;Username&gt;</code> .                                                                   |
| ERR_105<br>6 | More than one LDAP auth provider registration found ( <code>&lt;Number of registrations&gt;</code> ).<br>Unexpected behavior may result!     |
| ERR_105<br>7 | More than one PASSWORD auth provider registration found ( <code>&lt;Number of registrations&gt;</code> ).<br>Unexpected behavior may result! |
| ERR_107<br>6 | Unable to create keystore: <code>&lt;CA&gt;</code>                                                                                           |
| ERR_107<br>7 | Cryptography issue when creating user.                                                                                                       |
| ERR_107<br>8 | Cryptographic error processing password.                                                                                                     |
| ERR_107<br>9 | Unable to initialize SSLContext for LDAPS                                                                                                    |

| Code     | Message                                                                              |
|----------|--------------------------------------------------------------------------------------|
| ERR_1080 | More than one LDAP auth provider registration present. Unexpected results may occur. |
| ERR_1081 | LDAP authentication error.                                                           |
| ERR_1082 | Unexpected exception during LDAP lookup.                                             |
| ERR_1083 | Error closing LDAP context.                                                          |
| ERR_1084 | Could not find Active Directory user.                                                |
| ERR_1085 | Error creating the daemon user:                                                      |
| ERR_1086 | Error creating the initial user:                                                     |

## Administration errors

The application throws the following administration errors.

| Code     | Message                                                   |
|----------|-----------------------------------------------------------|
| ERR_1100 | Internal error occurred                                   |
| ERR_1101 | Error parsing license : <Error message>                   |
| ERR_1102 | Error parsing license: Epm client could not parse license |
| ERR_1103 | Error parsing license : <Error message>                   |
| ERR_1104 | Error parsing license: Epm client could not parse license |

| Code         | Message                                                                                                                                                  |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_11<br>05 | Order Number of <code>&lt;Order number&gt;</code> uploaded license doesn't match the existing license<br><code>&lt;Customer contact reference&gt;</code> |
| ERR_11<br>06 | License revision <code>&lt;Revision&gt;</code> already uploaded.                                                                                         |
| ERR_11<br>07 | Uploaded license revision <code>&lt;Uploaded revision&gt;</code> is outdated. Current license revision :<br><code>&lt;Current revision&gt;</code> .      |
| ERR_11<br>08 | Failed to create the license expiry schedule                                                                                                             |
| ERR_11<br>09 | Failed to send email for license consumption                                                                                                             |
| ERR_11<br>10 | Failed to send email for license expiry                                                                                                                  |
| ERR_11<br>11 | Failed to check the license expiry schedule                                                                                                              |
| ERR_11<br>12 | Failed to delete existing license expiry schedule                                                                                                        |
| ERR_11<br>13 | Failed to create the license expiry schedule                                                                                                             |
| ERR_11<br>14 | Invalid plugin name: <code>&lt;Plugin name&gt;</code>                                                                                                    |
| ERR_11<br>15 | Error executing plugin options for plugin: <code>&lt;Plugin name&gt;</code>                                                                              |
| ERR_11<br>16 | Error loading plugin jar <code>&lt;JAR file name&gt;</code> . Plugin will not be loaded!                                                                 |
| ERR_11<br>17 | Error loading plugin classloader.                                                                                                                        |

| Code         | Message                                                                                                                                                                                                            |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_11<br>18 | Plugin <code>&lt;Canonical name&gt;</code> is missing a language bundle. Plugin will not be loaded!                                                                                                                |
| ERR_11<br>19 | Plugin <code>&lt;Canonical name&gt;</code> has invalid language bundle. No messages section found. Plugin will not be loaded!                                                                                      |
| ERR_11<br>20 | Plugin <code>&lt;Canonical name&gt;</code> has invalid language bundle. No languages found. Plugin will not be loaded!                                                                                             |
| ERR_11<br>21 | Plugin <code>&lt;Canonical name&gt;</code> has an invalid language bundle. Language <code>&lt;Key&gt;</code> is an invalid map. Plugin will not be loaded!                                                         |
| ERR_11<br>22 | Plugin <code>&lt;Canonical name&gt;</code> has an invalid language bundle. Language <code>&lt;Name&gt;</code> , key <code>&lt;Key&gt;</code> is invalid ( <code>&lt;Value&gt;</code> ). Plugin will not be loaded! |
| ERR_11<br>23 | Error initializing plugins! No <code>&lt;Plugin class name&gt;</code> plugins will be loaded until invalid plugin is removed!                                                                                      |
| ERR_11<br>24 | updatePlugin: Error converting global options to Json string from list                                                                                                                                             |
| ERR_11<br>25 | validatePluginStateUpdate : cannot deactivate plugins that don't require license                                                                                                                                   |
| ERR_11<br>26 | validatePluginStateUpdate : cannot deactivate plugin <code>&lt;Name&gt;</code> as its in use by destination : <code>&lt;Label&gt;</code>                                                                           |
| ERR_11<br>27 | validatePluginStateUpdate : cannot deactivate plugin <code>&lt;Name&gt;</code> as its in use by source : <code>&lt;Label&gt;</code>                                                                                |
| ERR_11<br>28 | addPlugin: Error converting global options to Json string from list                                                                                                                                                |
| ERR_11<br>29 | Error converting global options to list from <code>Json byte[]</code>                                                                                                                                              |
| ERR_11<br>30 | addPlugin: Error converting global options to list from <code>Json byte[]</code>                                                                                                                                   |

| Code         | Message                                                                           |
|--------------|-----------------------------------------------------------------------------------|
| ERR_11<br>31 | Error fetching language bundle, Plugin <code>&lt;Plugin name&gt;</code> not found |
| ERR_11<br>32 | Failed to add an entry to the keystore: <code>&lt;TBU&gt;</code>                  |
| ERR_11<br>33 | Plugin update failed, plugin ID <code>&lt;Plugin ID&gt;</code>                    |
| ERR_11<br>49 | Failed importing multiple addresses.                                              |
| ERR_11<br>50 | Failed importing single addresses.                                                |
| ERR_11<br>53 | Failed to check the events retention schedule: <code>&lt;Error&gt;</code>         |
| ERR_11<br>54 | Failed to create the events retention schedule: <code>&lt;Error&gt;</code>        |
| ERR_11<br>99 | Unhandled exception caught                                                        |

## Automation errors

The application throws the following automation errors.

| Code         | Message                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| ERR_12<br>07 | Failed to mapping existing source plugin options.                                                                         |
| ERR_12<br>08 | Failed to process existing source plugin options.                                                                         |
| ERR_12<br>09 | Failed to migrate existing source plugin options.                                                                         |
| ERR_12<br>14 | Failed to send email for report <code>&lt;Report name&gt;</code> , schedule id: <code>&lt;Schedule ID&gt;</code> . Error: |

| Code      | Message                                                                                            |
|-----------|----------------------------------------------------------------------------------------------------|
| ERR_12_15 | Failed to generate missing report: <Report ID>                                                     |
| ERR_12_16 | Failed to generate missing schedule: <Schedule ID>                                                 |
| ERR_12_17 | Failed to return report: <Report ID> . Error: <Error>                                              |
| ERR_12_18 | Error while retrieving report data:                                                                |
| ERR_12_19 | Error while generating report:                                                                     |
| ERR_12_20 | User <Username> does not have permission to edit or delete report <Report name>                    |
| ERR_12_21 | Error while generating report: <Error>                                                             |
| ERR_12_22 | User <Username> does not have permission to access artifact <Artifact ID>                          |
| ERR_12_23 | User <Username> does not have permission to access execution <Execution ID>                        |
| ERR_12_24 | Failed to check the reports retention schedule: <Error>                                            |
| ERR_12_25 | Failed to create the reports retention schedule: <Error>                                           |
| ERR_12_30 | Field ' <Name> ' value ' <Value> ' cannot be parsed as <Type> . Field will be treated as a String. |
| ERR_12_31 | Unexpected exception while processing rule. RULE WILL BE SKIPPED!                                  |

| Code         | Message                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_12<br>32 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The address field <code>&lt;Address field&gt;</code> is empty.                                |
| ERR_12<br>33 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The address field <code>&lt;Address field&gt;</code> is not referring to a text custom field. |
| ERR_12<br>34 | Action plugins not currently supported. THIS ACTION WILL BE SKIPPED!                                                                                                                 |
| ERR_12<br>35 | Exception while executing rule. RULE WILL BE SKIPPED!                                                                                                                                |
| ERR_12<br>36 | Error running rules engine for certificate renewal rule.                                                                                                                             |
| ERR_12<br>37 | Execution of action failed.                                                                                                                                                          |
| ERR_12<br>38 | FAILED processing conditions. RULE WILL BE SKIPPED!                                                                                                                                  |
| ERR_12<br>39 | I/O issue while parsing conditions. RULE WILL BE SKIPPED!                                                                                                                            |
| ERR_12<br>40 | Error running rules engine for event.                                                                                                                                                |
| ERR_12<br>41 | Could not parse plugin config, ACTION WILL BE SKIPPED: <code>&lt;Plugin config&gt;</code>                                                                                            |
| ERR_12<br>42 | FAILED to create the expiration rules schedule! Expiry notifications will not be sent!                                                                                               |
| ERR_12<br>43 | Error while processing event rule conditions. RULE WILL BE SKIPPED!                                                                                                                  |
| ERR_12<br>44 | Only NOTIFICATION actions are supported! ACTION WILL BE SKIPPED!                                                                                                                     |

| Code         | Message                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| ERR_12<br>54 | Unexpected IOException while formatting the certificate. Error:                                                           |
| ERR_12<br>55 | Unexpected IOException while formatting the certificate chain. Error:                                                     |
| ERR_12<br>56 | Unexpected IOException while formatting the certificate. Error:                                                           |
| ERR_12<br>60 | FAILED to create the key manager scan schedule! Key managers will not be scanned!                                         |
| ERR_12<br>61 | Error encountered while scanning key manager.                                                                             |
| ERR_12<br>62 | Error encountered while scanning source.                                                                                  |
| ERR_12<br>71 | User <code>&lt;User ID&gt;</code> does not have permission to view, edit or delete destination <code>&lt;Label&gt;</code> |
| ERR_12<br>72 | Error verifying destination config <code>&lt;Label&gt;</code>                                                             |
| ERR_12<br>73 | Error verifying destination config for plugin <code>&lt;Plugin name&gt;</code>                                            |
| ERR_12<br>74 | Error while generating report.                                                                                            |
| ERR_12<br>75 | Failed to retrieve schedule runtimes for <code>&lt;Schedule name&gt;</code>                                               |
| ERR_12<br>76 | Failed to parse schedule runtimes for <code>&lt;Schedule name&gt;</code>                                                  |
| ERR_12<br>80 | Failed processing conditions for renewal success. RULE WILL BE SKIPPED!                                                   |
| ERR_12<br>81 | I/O issue while parsing conditions for renewal success. RULE WILL BE SKIPPED!                                             |



| Code         | Message                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_12<br>82 | Failed processing conditions for renewal failure. RULE WILL BE SKIPPED!                                                                                                              |
| ERR_12<br>83 | I/O issue while parsing conditions for renewal failure. RULE WILL BE SKIPPED!                                                                                                        |
| ERR_12<br>89 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The address field <code>&lt;Address field&gt;</code> is empty.                                |
| ERR_12<br>90 | Error running rules engine for certificate renewal rule.                                                                                                                             |
| ERR_12<br>91 | Failed processing rule. RULE WILL BE SKIPPED!                                                                                                                                        |
| ERR_12<br>92 | I/O issue while running rule. RULE WILL BE SKIPPED!                                                                                                                                  |
| ERR_12<br>93 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The custom field <code>&lt;Custom field&gt;</code> is empty.                                  |
| ERR_12<br>94 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The address field <code>&lt;Address field&gt;</code> is not referring to a text custom field. |
| ERR_12<br>95 | Expiry notification is dropped for certificate <code>&lt;Certificate name&gt;</code> . The address field <code>&lt;Address field&gt;</code> is empty.                                |
| ERR_12<br>96 | Error running rules engine for certificate renewal rule.                                                                                                                             |
| ERR_12<br>99 | Execution of rule action failed.                                                                                                                                                     |

## Control errors

The application throws the following control errors.

| Code     | Message                                                                                                                         |
|----------|---------------------------------------------------------------------------------------------------------------------------------|
| ERR_1302 | Error getting authority capabilities from CAGW                                                                                  |
| ERR_1303 | Failed to check the domain sync trigger                                                                                         |
| ERR_1304 | Unable to add domain sync for authority                                                                                         |
| ERR_1305 | Internal error contacting CAGW.                                                                                                 |
| ERR_1306 | Error while reading XML stream from upload.                                                                                     |
| ERR_1307 | Unexpected exception while pushing certificate:                                                                                 |
| ERR_1308 | HTTP Error while uploading certificate: <code>&lt;Error&gt;</code> :\n <code>&lt;Response body&gt;</code>                       |
| ERR_1309 | Error while uploading certificate: <code>&lt;Error&gt;</code>                                                                   |
| ERR_1310 | Unable to parse properties for domain: <code>&lt;Domain name&gt;</code>                                                         |
| ERR_1311 | User <code>&lt;User ID&gt;</code> doesn't have access to authority <code>&lt;Authority ID&gt;</code>                            |
| ERR_1312 | Internal error contacting CAGW.                                                                                                 |
| ERR_1313 | Unable to get profiles for authority                                                                                            |
| ERR_1314 | Unable to get the subject DN for authority                                                                                      |
| ERR_1315 | Unable to get the Capabilities for authority                                                                                    |
| ERR_1316 | Unexpected error contacting CAGW: <code>&lt;Error&gt;</code>                                                                    |
| ERR_1330 | User <code>&lt;User ID&gt;</code> does not have permission to view, edit or delete key manager <code>&lt;Key manager&gt;</code> |
| ERR_1331 | Error verifying key manager config <code>&lt;Key manager label&gt;</code>                                                       |
| ERR_1332 | saveOrUpdateKeyManager: Error converting plugin options to Json string from list                                                |

| Code     | Message                                                                            |
|----------|------------------------------------------------------------------------------------|
| ERR_1333 | Error converting plugin options to list from Json byte[]                           |
| ERR_1334 | Error verifying key manager config for plugin <Plugin name>                        |
| ERR_1349 | Failed to sync domains, Error from CAGW: <Error>                                   |
| ERR_1350 | Unexpected response received from CAGW                                             |
| ERR_1351 | Internal error contacting CAGW                                                     |
| ERR_1352 | Unexpected response received from CAGW: <Error>                                    |
| ERR_1353 | Unexpected response received from CAGW: <Error>                                    |
| ERR_1354 | Unexpected response received from CAGW: <Error>                                    |
| ERR_1355 | Error configuring the SSL client connection to the CAGW APIs.                      |
| ERR_1356 | Error configuring the SSL client connection to the CAGW APIs.                      |
| ERR_1357 | Error configuring the SSL client connection to the CAGW APIs                       |
| ERR_1358 | Error configuring the SSL client connection to the CAGW APIs.                      |
| ERR_1359 | Error configuring the SSL client connection to the CAGW APIs.                      |
| ERR_1362 | Error parsing authority certificate validity period: <Certificate validity period> |
| ERR_1363 | Error parsing authority certificate validity period: <Certificate validity period> |
| ERR_1374 | Error response from CAGW: <Error>                                                  |
| ERR_1375 | Unable to parse properties for domain: <Domain name>                               |
| ERR_1376 | Internal error contacting CAGW: <Error>                                            |

| Code     | Message                                                                          |
|----------|----------------------------------------------------------------------------------|
| ERR_1377 | Internal error contacting CAGW.                                                  |
| ERR_1378 | Internal error contacting CAGW.                                                  |
| ERR_1379 | Internal error contacting CAGW while responding to an authority request.         |
| ERR_1380 | Failed to create the authority domain sync schedule for authority <Authority ID> |
| ERR_1381 | Failed to delete the authority domain sync schedule for authority <Authority ID> |
| ERR_1382 | Certificate Authority <Authority ID> not found                                   |
| ERR_1383 | Unable to parse plugin options for authority <Authority ID> :                    |
| ERR_1384 | Error response from CAGW while getting domain: <Domain name>                     |
| ERR_1385 | Failed to get domain. Error from CAGW: <Error>                                   |
| ERR_1386 | Failed to submit domain, Error from CAGW: <Error>                                |
| ERR_1387 | Unable to fetch whois record from server <Server name> . Error:                  |
| ERR_1388 | Unable to close whois client connection with server <Server name> . Error:       |
| ERR_1389 | Unable to fetch whois record from default host. Error:                           |
| ERR_1390 | Unable to close whois client connection with default server. Error:              |
| ERR_1392 | Error on DNS lookup : <Error>                                                    |
| ERR_1394 | Failed to submit domain, Error from CAGW: <Error>                                |
| ERR_1397 | Certificate Authority <Authority ID> not found                                   |
| ERR_1398 | Unable to parse plugin options for authority <Authority ID>                      |

| Code     | Message                                                                                                                                     |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1399 | Unable to import/update domain id <code>&lt;Domain ID&gt;</code> due to Json parsing error from authority <code>&lt;Authority ID&gt;</code> |

## Certificate errors

The application throws the following certificate errors.

| Code     | Message                                                                                                                                                                       |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1426 | Renewal failed. Missing certificate id.                                                                                                                                       |
| ERR_1427 | Failed auto renewal for certificate <code>&lt;Certificate ID&gt;</code> .                                                                                                     |
| ERR_1428 | Automated renewal failed for certificate <code>&lt;Certificate ID&gt;</code> due to certificate processing error                                                              |
| ERR_1430 | Automated renewal failed for certificate <code>&lt;Certificate ID&gt;</code> due to destination errors: <code>&lt;List of errors&gt;</code>                                   |
| ERR_1431 | Failed to find the renewal daemon user for auto renewal                                                                                                                       |
| ERR_1432 | Failed to create the renewal schedule for cert <code>&lt;Certificate serial&gt;</code> : <code>&lt;Error&gt;</code>                                                           |
| ERR_1433 | Failed to check the renewal schedule <code>&lt;Error&gt;</code>                                                                                                               |
| ERR_1434 | Failed to create the renewal schedule <code>&lt;Error&gt;</code>                                                                                                              |
| ERR_1435 | Adding definition for custom field with duplicate display order : <code>&lt;Label&gt;</code> of type <code>&lt;Type&gt;</code> at position <code>&lt;Display order&gt;</code> |
| ERR_1436 | Deleting definition for custom field with Id : <code>&lt;Metadata ID&gt;</code> failed as it is in use by <code>&lt;Certificates using metadata&gt;</code> certificates       |

| Code         | Message                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>437 | Updating definition for custom field with duplicate display order : <Label> of type <Type> at position <Display order>                  |
| ERR_1<br>438 | Updating definition for custom field with Id : <Metadata ID> failed as it is in use by <Certificates using metadata> certificates       |
| ERR_1<br>439 | Updating definition for custom field with duplicate display order : <Metadata values>                                                   |
| ERR_1<br>440 | Other certificate custom field definitions exists with same display order <List>                                                        |
| ERR_1<br>441 | Updating definition for custom field with Id : <Metadata ID> failed as one of its value <List> is in use by <Certificates> certificates |
| ERR_1<br>442 | Error parsing the value <Value> for custom field <Metadata ID>                                                                          |
| ERR_1<br>443 | Unsupported Operator <Operator> for custom field Id: <Metadata ID>                                                                      |
| ERR_1<br>450 | Could not unarchive certificate because entitlement limit reached.                                                                      |
| ERR_1<br>452 | Error response from CAGW <Error>                                                                                                        |
| ERR_1<br>453 | Error exporting a certificate: <Error>                                                                                                  |
| ERR_1<br>454 | Failed to parse certificate <Certificate name> stored in DB. Error: <Error>                                                             |
| ERR_1<br>455 | Certificate Chain is not available for export                                                                                           |
| ERR_1<br>456 | Error while exporting certificate: <Error>                                                                                              |

| Code         | Message                                                                                                                                                                                                |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>457 | Error saving chain to keystore for export of : <Certificate name>                                                                                                                                      |
| ERR_1<br>458 | Error adding P12 to response stream                                                                                                                                                                    |
| ERR_1<br>459 | Unable to parse response from CAGW to export certificate for : <Certificate name> . Error: <Error>                                                                                                     |
| ERR_1<br>460 | Certificate can not be exported since the issuing Authority is not known                                                                                                                               |
| ERR_1<br>461 | Certificate Authority not found                                                                                                                                                                        |
| ERR_1<br>462 | Error adding P12 to response stream                                                                                                                                                                    |
| ERR_1<br>463 | Unexpected response received from CAGW when exporting a certificate                                                                                                                                    |
| ERR_1<br>464 | Internal error contacting CAGW                                                                                                                                                                         |
| ERR_1<br>465 | Failed to export certificate for <Certificate name> . Error from CAGW: <Error>                                                                                                                         |
| ERR_1<br>466 | Failed to export certificate for <Certificate name> with serial number <Certificate serial number> . Certificate key is not backed up.                                                                 |
| ERR_1<br>467 | Unable to parse response from CAGW to export certificate for : <Certificate name> . Error: <Error>                                                                                                     |
| ERR_1<br>468 | Export private key is not supported for export type <Type> You can uncheck \\\\"Include Private Key\\\\" and try again, however, your exported certificate will not have the private key               |
| ERR_1<br>469 | Export certificate chain is not supported for export type <Type> You can uncheck \\\\"Include Certificate Chain\\\\" and try again, however, your exported certificate will not have certificate chain |

| Code         | Message                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>470 | Public certificate must be requested for export type <code>&lt;Type&gt;</code>                                                          |
| ERR_1<br>471 | At least one of public certificate, certificate chain or private key must be requested for export type <code>&lt;Type&gt;</code>        |
| ERR_1<br>472 | At least one of public certificate, certificate chain or private key must be requested for export type <code>&lt;Type&gt;</code>        |
| ERR_1<br>473 | Unable to revoke the authority <code>&lt;Authority name&gt;</code>                                                                      |
| ERR_1<br>474 | Unable to unhold the authority <code>&lt;Authority name&gt;</code>                                                                      |
| ERR_1<br>477 | Error building certificate query with filter : <code>&lt;Filter&gt;</code> . Error <code>&lt;Error&gt;</code>                           |
| ERR_1<br>478 | Error fetching certificates with predicate : <code>&lt;Predicate&gt;</code> . Error <code>&lt;Error&gt;</code>                          |
| ERR_1<br>479 | Certificate Bulk Edit Error: 'certificatesFilter' missing from request body                                                             |
| ERR_1<br>480 | Certificate Bulk Edit Error: If 'clearOutAccessTags' is set, 'accessTags' must be empty.                                                |
| ERR_1<br>481 | Certificate Bulk Edit Error: No updated values provided                                                                                 |
| ERR_1<br>482 | Certificate Bulk Edit Error building certificate query with filter : <code>&lt;Filter&gt;</code> , Error : <code>&lt;Error&gt;</code> . |
| ERR_1<br>483 | Certificate Bulk Edit Error building certificate query with filter : <code>&lt;Filter&gt;</code> , Error : <code>&lt;Error&gt;</code> . |
| ERR_1<br>484 | Certificate Bulk Edit Error updating certificates with filter : <code>&lt;Filter&gt;</code> , Error : <code>&lt;Error&gt;</code>        |



| Code         | Message                                                                                                                                                  |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>486 | Certificate unhold error : Could not find certificate with id: <code>&lt;Certificate ID&gt;</code> .                                                     |
| ERR_1<br>487 | Certificate unhold error : No Authority Id associated with this certificate: <code>&lt;Certificate ID&gt;</code> .                                       |
| ERR_1<br>488 | Certificate unhold error : Cannot unhold certificate <code>&lt;Certificate ID&gt;</code> . Authority is not active : <code>&lt;Authority ID&gt;</code> . |
| ERR_1<br>489 | Certificate unhold error : Cannot unhold certificate <code>&lt;Certificate ID&gt;</code> . No external id found.                                         |
| ERR_1<br>490 | Issue certificate error : Subject DN is required for CSR.                                                                                                |
| ERR_1<br>491 | Issue certificate error : CAGW failed to create certificate for authority <code>&lt;Authority ID&gt;</code>                                              |
| ERR_1<br>492 | Issue certificate error : CAGW Failed to create certificate: <code>&lt;Key manager ID&gt;</code> .                                                       |
| ERR_1<br>493 | Issue certificate error : Subject DN is required for CSR.                                                                                                |
| ERR_1<br>494 | Issue certificate error : Subject DN is required for CSR.                                                                                                |
| ERR_1<br>495 | Issue certificate error : Subject DN is required for CSR.                                                                                                |
| ERR_1<br>496 | Issue certificate error : Subject DN is required for CSR.                                                                                                |
| ERR_1<br>497 | Failed to save certificate: <code>&lt;Error&gt;</code>                                                                                                   |
| ERR_1<br>498 | Failed to upload certificate to the key manager <code>&lt;Key manager ID&gt;</code> . Error <code>&lt;Error&gt;</code>                                   |

| Code         | Message                                                                                                            |
|--------------|--------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>499 | Certificate revoke error : No Authority Id associated with this certificate. <Certificate ID>                      |
| ERR_1<br>500 | Certificate revoke error : Cannot revoke certificate <Certificate ID> .Authority <Authority ID> is not active.     |
| ERR_1<br>501 | Certificate revoke error : Cannot revoke certificate <Certificate ID> . No external id found.                      |
| ERR_1<br>502 | Failed to apply service-level filters on query <Filter>                                                            |
| ERR_1<br>503 | Failed to apply service-level filters on query <Filter>                                                            |
| ERR_1<br>504 | Failed to apply service-level filters on query <Filter>                                                            |
| ERR_1<br>505 | Could not find certificate with id <Certificate ID>                                                                |
| ERR_1<br>506 | Could not find certificate with id <Certificate ID>                                                                |
| ERR_1<br>508 | Failed to issue a certificate from authority <Authority name> . Error <Error>                                      |
| ERR_1<br>509 | Failed to parse the X509 certificate <Certificate body> \n Message: <Error>                                        |
| ERR_1<br>510 | Unable to find certificate <Certificate ID>                                                                        |
| ERR_1<br>511 | Failed to process the certificate <Certificate import request body> \n Message: <Error>                            |
| ERR_1<br>512 | Failed to process the certificate <Certificate body> \n External ID: <Certificate External ID> \n Message: <Error> |

| Code         | Message                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>513 | Failed to apply service-level filters on query <code>&lt;Filter&gt;</code>                                                           |
| ERR_1<br>514 | Failed to run the certificate count query: <code>&lt;Error&gt;</code>                                                                |
| ERR_1<br>521 | Error verifying source config <code>&lt;Source Label&gt;</code>                                                                      |
| ERR_1<br>522 | Error verifying source config for plugin <code>&lt;Plugin name&gt;</code>                                                            |
| ERR_1<br>523 | addOrUpdateSource: Error converting plugin options to Json string from list                                                          |
| ERR_1<br>524 | Error scheduling source sync, sources will not be scanned!                                                                           |
| ERR_1<br>525 | Error creating certificate from certificate request                                                                                  |
| ERR_1<br>526 | Failed to send new external certificate request notification to approver(s). Error: <code>&lt;Notification message&gt;</code>        |
| ERR_1<br>527 | Failed to send external certificate request cancellation notification to requestor. Error: <code>&lt;Notification message&gt;</code> |
| ERR_1<br>528 | Failed to send certificate request approval notification to requestor. Error: <code>&lt;Notification message&gt;</code>              |
| ERR_1<br>529 | Failed to send certificate request rejection notification to requestor. Error: <code>&lt;Notification message&gt;</code>             |
| ERR_1<br>530 | Failed to send new certificate request notification to internal requestor. Error: <code>&lt;Notification message&gt;</code>          |
| ERR_1<br>531 | Failed to send new internal certificate request notification to approver(s). Error: <code>&lt;Notification message&gt;</code>        |

| Code         | Message                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------|
| ERR_1<br>533 | Failed to send new certificate request notification to external requestor. Error: <Notification message>               |
| ERR_1<br>534 | CSR key algorithm <CSR key algorithm> does not match the required key algorithm <Allowed key algorithm>                |
| ERR_1<br>536 | CSR key algorithm keysize <CSR key size> does not meet minimum public key size required : <Allowed key size>           |
| ERR_1<br>537 | Invalid certificate signing request provided                                                                           |
| ERR_1<br>540 | Failed to send new certificate request notification to external requestor. SMTP Notification Plugin not found          |
| ERR_1<br>541 | Failed to send certificate request cancellation notification to external requestor. SMTP Notification Plugin not found |
| ERR_1<br>542 | Failed to send new external certificate request notification to approver(s). SMTP Notification Plugin not found        |
| ERR_1<br>543 | Failed to send new certificate request notification to admin. SMTP Notification Plugin not found                       |
| ERR_1<br>544 | Failed to send new internal certificate request notification to approver(s). SMTP Notification Plugin not found        |
| ERR_1<br>545 | Failed to send notification for certificate request cancellation. SMTP Notification Plugin not found                   |
| ERR_1<br>546 | Failed to send notification for certificate request approval. SMTP Notification Plugin not found                       |
| ERR_2<br>010 | Found invalid certificate with name <Certificate name> .                                                               |
| ERR_2<br>011 | Unexpected exception while processing certificate.                                                                     |

| Code     | Message                                                                                                                   |
|----------|---------------------------------------------------------------------------------------------------------------------------|
| ERR_2012 | Error processing certificate.                                                                                             |
| ERR_2013 | Error creating certificate factory.                                                                                       |
| ERR_2015 | Failed to parse certificate <code>&lt;Certificate name&gt;</code> stored in DB. Error: <code>&lt;Error message&gt;</code> |

## 14 Integration report

Certificate Hub allows you to view and manage certificates across your enterprise, regardless of the issuer.

**i** Certificate Hub Manager uses Entrust CA Gateway as the underlying CA interface.

- [Entrust products compatible with Certificate Hub](#)
- [Supported Deployment Platforms](#)
- [Supported Web Browser](#)
- [Databases supported by Certificate Hub](#)
- [Plugins supported by Certificate Hub](#)
- [Standards supported by Certificate Hub](#)

### Entrust products compatible with Certificate Hub

| Product(s) | Version(s)     | Support Notes                                                                             |
|------------|----------------|-------------------------------------------------------------------------------------------|
| CA Gateway | 3.x            |                                                                                           |
| IDaaS      | Not applicable | IDaaS (Identity as a Service) is supported as an Identity Provider for multi-factor login |

### Supported Deployment Platforms

| Platform   | Version | Support Notes                                                                                                                                                                                                                          |
|------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kubernetes | 1.25    | Certificate Hub has been run on Kubernetes 1.25. We expect it to operate on all supported Kubernetes and Kubernetes variants, such as RedHat OpenShift and Rancher OS. Only upgrade from older releases are supported over Kubernetes. |

### Supported Web Browser

Certificate Hub administration console supports the following web browsers.

| Browser       | Windows     | Mac OS      |
|---------------|-------------|-------------|
| Apple Safari  | 5 or higher | 5 or higher |
| Google Chrome | 8 or higher | 8 or higher |

| Browser         | Windows                                                                                                                                                                                                              | Mac OS        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Microsoft Edge  | The stable versions listed by Microsoft in <a href="https://learn.microsoft.com/en-us/deployedge/microsoft-edge-support-lifecycle">https://learn.microsoft.com/en-us/deployedge/microsoft-edge-support-lifecycle</a> | Not supported |
| Mozilla Firefox | 9 or higher                                                                                                                                                                                                          | 9 or higher   |

Browser compatibility is quite high, so most versions operate without issue. If there is an issue, we will address it using the latest browser version available for the operating system.

## Databases supported by Certificate Hub

The external Certificate Hub database must be hosted on PostgreSQL 15 or higher.

## Plugins supported by Certificate Hub

Certificate Hub supports the following plugins.

| Plugin    | Version                           | Support Notes                                      |
|-----------|-----------------------------------|----------------------------------------------------|
| KMIP KMS  | 2.0                               | Works for any KMS supported by a KMIP 2.0 protocol |
| F5 BIG IP | F5 iControl REST API version 15.1 | Also compatible with newer F5 BIG-IP versions.     |

## Standards supported by Certificate Hub

Certificate Hub supports the following standards.

| Standard                                     | Version | Supported for external IdP providers | Notes                                    |
|----------------------------------------------|---------|--------------------------------------|------------------------------------------|
| OpenID Connect (OIDC)                        | 1.0     | ✓                                    | OIDC 1.0 is a layer on top of OAuth 2.0. |
| Lightweight Directory Access Protocol (LDAP) | v3      | ✓                                    | Includes Active Directory.               |