

# Cloud Certificate Manager

## User Guide(PCA)

**Issue** 01  
**Date** 2023-12-15



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

---

# Contents

---

<b>1 Overview of Private Certificate Application.....</b>	<b>1</b>
<b>2 Private CA Management.....</b>	<b>3</b>
2.1 Creating a Private CA.....	3
2.2 Activating a Private CA.....	7
2.3 Viewing Private CA Details.....	9
2.4 Configuring a CRL.....	11
2.5 Exporting a Private CA Certificate.....	13
2.6 Disabling a Private CA.....	14
2.7 Enabling a Private CA.....	15
2.8 Deleting a Private CA.....	15
2.9 Canceling the Deletion of a Private CA.....	17
<b>3 Private Certificate Management.....</b>	<b>19</b>
3.1 Applying for a Private Certificate.....	19
3.2 Downloading a Private Certificate.....	23
3.3 Installing a Private Certificate.....	24
3.3.1 Trusting a Private Root CA.....	24
3.3.2 Installing a Private Certificate on a Client.....	28
3.3.3 Installing a Private Certificate on a Server.....	30
3.3.3.1 Installing a Private Certificate on a Tomcat Server.....	30
3.3.3.2 Installing a Private Certificate on an Nginx Server.....	33
3.3.3.3 Installing a Private Certificate on an Apache Server.....	36
3.3.3.4 Installing a Private Certificate on an IIS Server.....	38
3.3.3.5 Installing a Private Certificate on a WebLogic Server.....	41
3.3.3.6 Installing a Private Certificate on a Resin Server.....	47
3.4 Revoking a Private Certificate.....	50
3.5 Viewing Details of a Private Certificate.....	52
3.6 Deleting a Private Certificate.....	53
<b>4 Permissions Management.....</b>	<b>55</b>
4.1 Creating a User and Granting CCM Permissions to the User.....	55
4.2 CCM Custom Policies.....	56
<b>A Change History.....</b>	<b>58</b>

# 1 Overview of Private Certificate Application

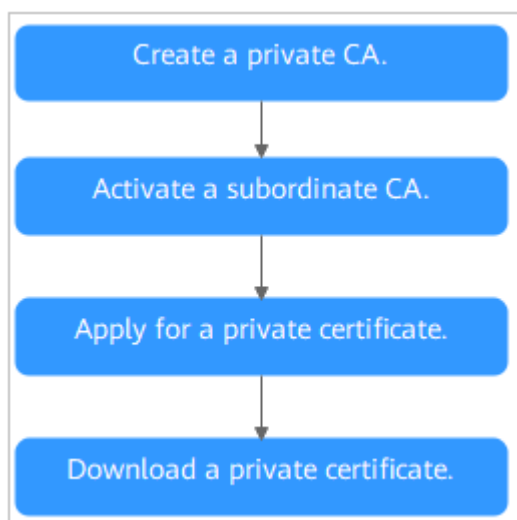
---

Cloud Certificate Manager (CCM) is a private CA and certificate management platform. You can use CCM to set up a complete CA hierarchy and use it to issue and manage private certificates for your organization. It is used to authenticate application identities and encrypt and decrypt data within your organization.

Certificates issued by a private CA are trusted only within your organization, but not the Internet.

For details, see [Figure 1-1](#) and [Table 1-1](#).

**Figure 1-1** Private certificate application procedure



**Table 1-1** Application procedure

Step	Operation	Description
1	<b>Creating a Private CA</b>	Create a private CA as required. If this is your first time creating a private CA, you must create a root CA. You can create multiple subordinate CAs under the existing root CA.
2	<b>Activating a Private CA</b>	A private root CA can be used to issue private certificates once it is created. A private subordinate CA must be activated before it is used to issue certificates.
3	<b>Applying for a Private Certificate</b>	Apply for a private certificate with the activated private CA.
4	<b>Downloading a Private Certificate</b>	After the application is approved, you can download the private certificate and install it on the server.

# 2 Private CA Management

---

## 2.1 Creating a Private CA

CCM helps you set up an internal CA for your organization with low costs and use it to issue certificates with ease.

This topic describes how to create a private root CA and subordinate CA.

### Overview


- Private CAs are classified into root CAs and subordinate CAs (intermediate CAs). A subordinate CA belongs to a root CA. A root CA can have multiple subordinate CAs.
- If this is your first time creating a private CA, you must create a root CA.
- A maximum of 100 CAs can be created for each user. Private CAs in the pending deletion state are also counted in the private CA quota until the private CAs are deleted.

### Prerequisites

The account for creating a private CA has the **PCA FullAccess** permission.

### Procedure

**Step 1** Log in to the [management console](#).

**Step 2** Click  in the upper left corner of the page and choose **Security & Compliance > Cloud Certificate Management Service**. In the navigation pane on the left, choose **Private Certificate Management > Private Certificate**. The **Private Certificate** page is displayed.

**Step 3** In the upper right corner of the private CA list, click **Create CA** to switch to the **Create CA** page.

**Step 4** Configure the CA information.

You need to specify the basic information, distinguished name, and certificate revocation configuration.

1. Configure the basic information. [Table 2-1](#) describes the parameters.

**Figure 2-1** Basic Information

The screenshot shows the 'Basic Information' configuration interface. It includes the following elements:

- CA Type:** Two radio button options. 'Root CA' is selected, with the description 'Creates a root CA and new CA hierarchy.' 'Subordinate CA' is unselected, with the description 'Creates a subordinate CA and adds a layer to the existing CA hierarchy.'
- Key Algorithm:** A dropdown menu currently showing 'RSA2048'.
- Signature Algorithm:** A dropdown menu currently showing 'SHA256'.
- Validity Period:** A spinner control showing the value '1' and a unit dropdown showing 'y...'.
- Expiration Time:** A timestamp at the bottom of the form: 'Nov 10, 2023 17:16:07 GMT+08:00'.

**Table 2-1** Basic information parameters

Parameter	Description	Example Value
CA Type	<p>Indicates the type of the CA to be created.</p> <p>The values can be:</p> <ul style="list-style-type: none"> <li>- <b>Root CA:</b> Select this option if you want to create a CA hierarchy.</li> </ul> <p><b>NOTE</b> If you create a private CA for the first time, you must create a root CA.</p> <ul style="list-style-type: none"> <li>- <b>Subordinate CA:</b> Select this option if you want to add a layer to the existing CA hierarchy.</li> </ul>	Root CA
Key Algorithm	<p>Indicates the key algorithm. The values can be:</p> <ul style="list-style-type: none"> <li>- <b>RSA2048</b></li> <li>- <b>RSA4096</b></li> <li>- <b>EC256</b></li> <li>- <b>EC384</b></li> </ul>	<b>RSA2048</b>

Parameter	Description	Example Value
Signature Algorithm	This parameter is displayed when <b>CA Type</b> is set to <b>Root CA</b> . You can select any of the following hash algorithms: <ul style="list-style-type: none"> <li>- <b>SHA256</b></li> <li>- <b>SHA384</b></li> <li>- <b>SHA512</b></li> </ul>	<b>SHA256</b>
Validity Period	This parameter is displayed when <b>CA Type</b> is set to <b>Root CA</b> . Indicates the validity period of a private certificate issuer. The longest period is 30 years.	3 years

2. Configure the certificated distinguished name. [Table 2-2](#) describes the parameters.

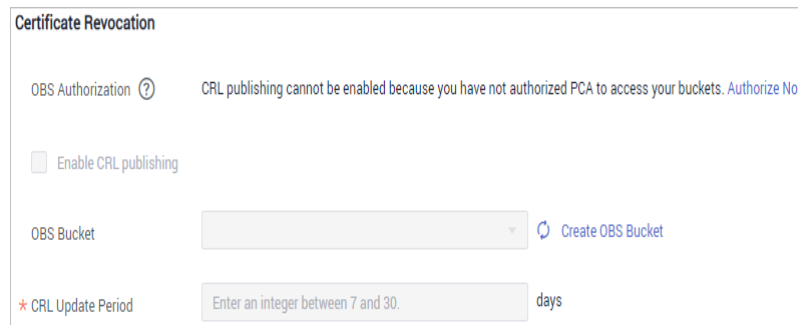
**Table 2-2** Parameters

Parameter	Description	Example Value
Common Name	Indicates the CA name.	N/A
Country/Region	Indicates the country or region where your organization belongs. Enter the two-letter code of the country or region.	FR
State/Province	Indicates the name of the province or state where your organization is located.	Paris
Locality	Indicates the name of the city where your organization is located.	Paris
Organization	The legal name of your company.	-
Organizational Unit	Indicates the department name.	Cloud Dept.

3. (Optional) Configure certificate revocation.  
If you want to use PCA to publish the certificate revocation list (CRL) for a private CA, you can configure parameters in this pane.  
Otherwise, skip this step.  
[Table 2-3](#) describes the parameters.



**Figure 2-2** Certificate Revocation



**Table 2-3** Certificate revocation parameters

Parameter	Description
OBS Authorization	Whether to authorize PCA to access your OBS bucket and upload the CRL file. If you want to authorize, click <b>Authorize Now</b> and complete the authorization as prompted. If you want to cancel the authorization, go to the IAM console to delete the PCAAccessPrivateOBS agency from the agency list. After the permission has been granted, follow-up operations do not require the permission to be granted again.
Enable CRL publishing	Indicates whether to enable CRL publishing.
OBS Bucket	Select an existing OBS bucket or click <b>Create OBS Bucket</b> to create an OBS bucket.
CRL Update Period	Indicates the CRL update period. PCA will generate a new CRL at the specified time. You can set the period to an integer between 7 and 30. If you do not specify a value, it is set to 7 days by default.

**Step 5** Click **Next** to enter the confirmation page.

**Step 6** After confirming the information about the private CA, click **Confirm and Create**.

If you create a root CA, the root CA is automatically activated after being created.  
If you create a subordinate CA, you need to manually activate it.

After you create a subordinate CA, you can choose **Activate Now** or **Activate Later**.

----End

## Follow-up Procedure

After a root CA is created, it can be used to issue private certificates. For details about how to apply for a private certificate, see [Applying for a Private Certificate](#).

After a subordinate CA is created, you need to install a certificate and activate the CA. For details, see [Activating a Private CA](#).

## 2.2 Activating a Private CA

A subordinate private CA must be activated after it is created. A subordinate private CA takes effect and can be used to issue private certificates only after it is activated.

This topic describes how to activate a subordinate CA. You can use either an internal private CA or external private CA to activate the subordinate CA.

- Internal private CA: Use a private CA in CCM to activate a subordinate CA.
- External private CA: Use a private CA from a third party to activate a subordinate CA.

### Prerequisites

- You have created a subordinate private CA. For details, see [Creating a Private CA](#).
- The subordinate CA is in the **Pending activation** state.

### Activating a Subordinate Private CA with an Internal Private CA

**Step 1** Log in to the [management console](#).

**Step 2** Locate the row of the subordinate CA and click **Activate** in the **Operation** column. In the **Install CA Certificate and Activate CA** page, configure the required parameters.

1. Configure **Issued From**.  
Select **Internal private CA**.
2. Configure the required parameters.

**Table 2-4** Parameters

Parameter	Description
Common Name	Indicates the name of the CA. The CA can be a root CA or a subordinate CA. After you select the CA, the system automatically displays the type and ID of the CA.

Parameter	Description
Signature Algorithm	Indicates the signature algorithm. The values can be: <ul style="list-style-type: none"> <li>- <b>SHA256</b></li> <li>- <b>SHA384</b></li> <li>- <b>SHA512</b></li> </ul>
Validity Period	Indicates the validity period of a private CA. The longest period is 20 years.
Path Length	The path length of the subordinate CA. The path length controls how many layers of subordinate CAs the current subordinate CA can issue. (The last layer of the certificate chain is a private certificate). <p><b>NOTE</b> A certificate chain is made up of root CAs, subordinate CAs, and private certificates in a fixed sequence to validate the trust of a certificate at a lower layer.</p>

**Step 3** Confirm the configuration and click **OK**.

----End

## Activating a Subordinate Private CA with a Third-Party Private CA

**Step 1** Log in to the [management console](#).

**Step 2** Locate the row of the subordinate CA and click **Activate** in the **Operation** column. In the **Install CA Certificate and Activate CA** page, configure the required parameters.

1. Configure **Issued From**. Select **External private CA**.
2. Export the CSR.

In the **CA CSR** pane, click **Export File**.

The PEM CSR is exported to a file and is signed by a parent CA.

3. Use the external CA to issue a certificate.

Use your private CA to issue a certificate for the subordinate private CA you want to activate.

4. Import the certificate.

Import the certificate and certificate chain in the **Import the Certificate Issued by an External CA** pane.

**Table 2-5** Parameter descriptions

Parameter	Description
Certificate	Open the PEM file in the certificate to be uploaded as a text file with the extension <b>.pem</b> and copy the certificate content to this text box.
Certificate Chain	Open the PEM file in the certificate to be uploaded as a text file with the extension <b>.pem</b> and copy the certificate chain to this text box.

**Step 3** Confirm the configuration and click **OK**.

If the status of the subordinate CA changes to **Activated**, the subordinate CA has been activated.

----End

## Follow-up Procedure

After a subordinate CA is activated, it can be used to issue private certificates. For details about how to apply for a private certificate, see [Applying for a Private Certificate](#).

## 2.3 Viewing Private CA Details

This topic describes how to view the private CA information, including **Common Name**, **Organizational Unit**, **Type**, and **Status**.

### Prerequisites

You have created a private CA. For details, see [Creating a Private CA](#).

### Procedure


**Step 1** Log in to the [management console](#).

**Step 2** View private CA information in the private CA list. [Table 2-6](#) describes the parameters.

**Figure 2-3** Private CA list

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
422	Root CA			2020/06/17 01:55:44 GMT+08:00	2021/06/17 01:56:44 GMT+08:00	Activated	Export CA Certificate   Disable
972	Root CA			2020/06/16 01:52:33 GMT+08:00	2021/06/16 01:53:33 GMT+08:00	Activated	Export CA Certificate   Disable
59	Root CA			2020/06/15 20:21:58 GMT+08:00	2021/06/15 20:22:58 GMT+08:00	Activated	Export CA Certificate   Disable

 **NOTE**

- Select a CA type or status from the type or status search box. CAs of the selected type or status will be displayed in the list.
- Enter a name of a CA in the search box in the upper right corner and click  or press **Enter** to search for a specified CA.

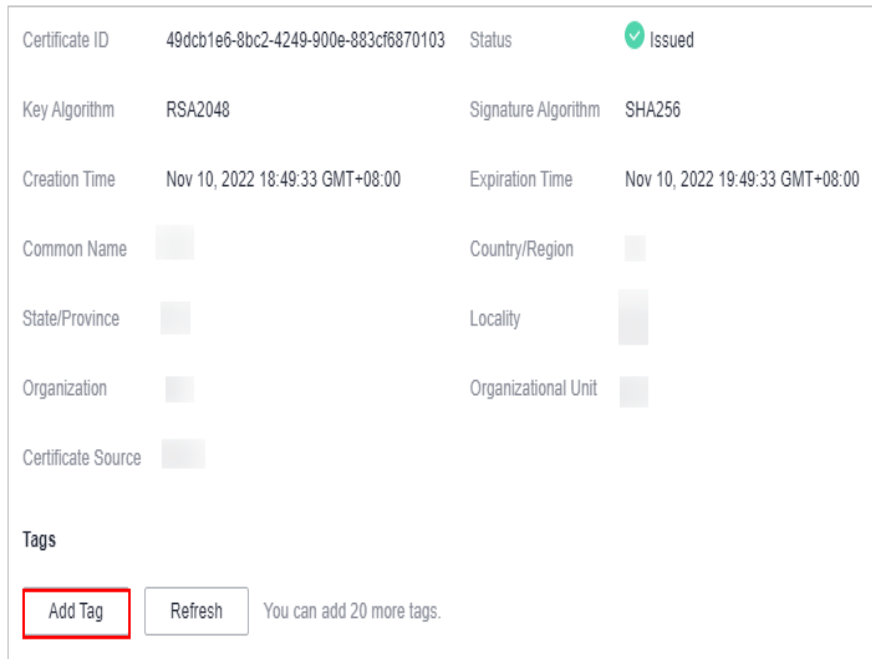
**Table 2-6** CA parameter description

Parameter	Description
Common Name	Indicates the user-defined CA name.
Type	Indicates the private CA type. The value can be: <ul style="list-style-type: none"> <li>• <b>Root CA:</b> The private CA is a root CA and can be used to issue subordinate CAs.</li> <li>• <b>Subordinate CA:</b> The private CA is a subordinate CA.</li> </ul>
Organizational Unit	Indicates the name of the organizational unit to which the private CA belongs.
Issued By	Indicates the name of the CA that issues the private CA.
Creation Time	Indicates the time when a private CA is created.
Expiration Time	Indicates the time when a private CA expires.
Status	Indicates the private CA status. The value can be: <ul style="list-style-type: none"> <li>• <b>Pending activation:</b> The private CA is to be activated.</li> <li>• <b>Activated:</b> The private CA is activated.</li> <li>• <b>Disabled:</b> The private CA is disabled.</li> <li>• <b>Pending deletion:</b> The private CA is to be deleted.</li> <li>• <b>Expired:</b> The private CA is expired.</li> </ul>
Operation	You can activate, enable, or disable a CA.

**Step 3** Click the common name of a private CA to view its details.

You can click **Add Tag** on the CA details page to identify the CA. TMS's predefined tag function is recommended for adding the same tag to different cloud resources.

Figure 2-4 Private CA details



----End

## 2.4 Configuring a CRL

If you want to use PCA to publish the certificate revocation list (CRL) for a private CA, you can enable CRL configuration.

This topic walks you through how to enable or disable CRL configuration.

### Prerequisites

The private CA for which you want to configure a CRL is in the **Activated** or **Disabled** state.

### Enabling CRL Configuration

- Step 1** Log in to the [management console](#).
- Step 2** Click the name of a private CA to go to its details page.
- Step 3** On the private CA details page, click the **CRL Configuration** tab and configure certificate revocation by referring to [Table 2-7](#).

**Figure 2-5** CRL Configuration

**Table 2-7** Certificate revocation parameters

Parameter	Description
OBS Authorization	Whether to authorize PCA to access your OBS bucket and upload the CRL file. If you want to authorize, click <b>Authorize Now</b> and complete the authorization as prompted. If you want to cancel the authorization, go to the IAM console to delete the PCAAccessPrivateOBS agency from the agency list. After the permission has been granted, follow-up operations do not require the permission to be granted again.
Enable CRL publishing	Indicates whether to enable CRL publishing.
OBS Bucket	Select an existing OBS bucket or click <b>Create OBS Bucket</b> to create an OBS bucket.
CRL Update Period	Indicates the CRL update period. PCA will generate a new CRL at the specified time. You can set the period to an integer between 7 and 30. If you do not specify a value, it is set to 7 days by default.

**Step 4** Click **Enable** to enable the CRL. If the system displays a message indicating that the CRL configuration is enabled, the CRL configuration has been enabled.

----End

## Disabling CRL Configuration

- Step 1** Log in to the [management console](#).
- Step 2** Click the name of a private CA to go to its details page.
- Step 3** On the private CA details page, click the **CRL Configuration** tab and click **Disable**. If the system displays a message indicating that the CRL configuration is disabled, the CRL configuration has been disabled.

----End

## 2.5 Exporting a Private CA Certificate

After a private CA is created and activated, you can export the private CA certificate.

If your web services are accessible through browsers, add the root certificate to your browser trust list and install the private certificate issued by the root CA on your web server to implement HTTPS communications between the client and the server.

If your web services are accessible through a client like Java, manually install the root certificate on the client to ensure that the client can validate the encrypted information on the server.

This topic walks you through how to export a private CA certificate.

### Prerequisites

The private CA for which the certificate is to be exported is in the **Activated** state.

### Procedure

- Step 1** Log in to the [management console](#).
- Step 2** Locate the row of the desired private CA and click **Export CA Certificate** in the **Operation** column.

**Figure 2-6** Exporting a CA certificate

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
422	Root CA			2020/06/17 01:55:44 GMT+08:00	2021/06/17 01:56:44 GMT+08:00	Activated	Export CA Certificate   Disable
972	Root CA			2020/06/16 01:52:33 GMT+08:00	2021/06/16 01:53:33 GMT+08:00	Activated	Export CA Certificate   Disable

- Step 3** In the displayed dialog box, click **OK**.

When you click **OK**, CCM will use the download tool provided by the browser to download the private CA certificate to the specified local directory.

Now, you will obtain a private CA certificate file named **root CA name\_certificate.pem**.

----End



## 2.6 Disabling a Private CA

If you no longer need a private CA to issue certificates, you can disable the private CA.

If a private CA is disabled, it cannot be used to issue any private certificates. If you want to use this private CA to issue certificates again, it must be enabled first. For details, see [Enabling a Private CA](#).

This topic describes how to disable a private CA.

### CAUTION

Private CAs will also remain billed while they are disabled.

### Prerequisites

The private CA to be disabled is in the **Activated** or **Expired** state.

### Procedure

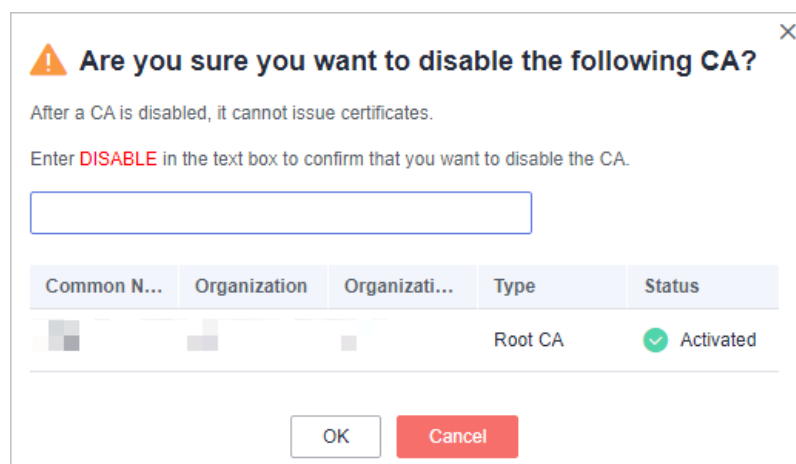
- Step 1** Log in to the [management console](#).
- Step 2** Locate the row of the desired private CA and click **Disable** in the **Operation** column.

**Figure 2-7** Disabling a private CA

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
422	Root CA			2020/06/17 01:55:44 GMT+08:00	2021/06/17 01:56:44 GMT+08:00	Activated	Export CA Certificate <b>Disable</b>
172	Root CA			2020/06/16 01:52:33 GMT+08:00	2021/06/16 01:53:33 GMT+08:00	Activated	Export CA Certificate   Disable

- Step 3** In the displayed dialog box, enter **DISABLE** and click **OK**.

**Figure 2-8** Disable CA



When "CA xxx disabled successfully." is displayed in the upper right corner of the page, and the private CA status changes to **Disabled**, the private CA is disabled successfully.

----End

## 2.7 Enabling a Private CA

If you need to use a disabled private CA to issue certificates, you can restore the certificate to the activated state.

The following walks you through how to enable a private CA so that you can quickly restore a disabled private CA to the activated or expired state.

### Prerequisites

The private CA to be enabled is in the **Disabled** state. For details about how to disable a private CA, see [Disabling a Private CA](#).

### Procedure

- Step 1** Log in to the [management console](#).
- Step 2** Locate the row of the desired private CA and click **Enable** in the **Operation** column.

**Figure 2-9** Enabling a private CA

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
ca-025	Root CA			2020/06/03 01:59:40 GMT+08:00	2021/06/03 02:00:40 GMT+08:00	Disabled	<a href="#">Enable</a> <a href="#">Delete</a>
ca-026	Root CA			2020/02/28 22:18:41 GMT+08:00	2021/02/28 22:17:41 GMT+08:00	Disabled	<a href="#">Enable</a> <a href="#">Delete</a>

When "CA xxx enabled successfully." is displayed in the upper right corner of the page, and the private CA status changes to **Activated**, the private CA is enabled successfully.

----End

## 2.8 Deleting a Private CA

Before deleting a private CA, ensure that it is not in use and will not be used.

If deletion is scheduled for a private CA in the **Disabled** or **Expired** state, the deletion will take effect after a waiting period of 7 to 30 days. If deletion is scheduled for a private CA in the **Pending activation** state, the deletion will take effect immediately. Before the specified deletion date, you can cancel the deletion if you want to use the private CA again. If the specified deletion period expires, the private CA will be permanently deleted. Exercise caution when performing this operation.

**CAUTION**

- Private CAs will also remain billed while they are disabled.
- If you delete a private CA, it takes a few days for the deletion to take effect. It takes at least 7 days for a scheduled deletion to take effect (depending on the delay time you configured). During the scheduled deletion period, you will be billed in accordance with the following rules:
  - If you have not canceled the scheduled deletion and the private CA is deleted, the private CA is not billed for this period.
  - If you cancel the scheduled deletion but the private CA is not deleted during this period, the private CA is still billed for this period.

For example, if you delete a private CA at 00:00 on January 1, 2022 and the private CA is deleted seven days later as scheduled, you will not be billed for the seven days. If you cancel the scheduled deletion at 00:00 on January 4, 2022 and the private CA is not deleted, you will still be billed for the CA for the period from 00:00 on January 1, 2022 to 00:00 on January 4, 2022.

### Prerequisites

The private CA to be deleted is in the **Disabled** or **Pending activation** state.

### Procedure

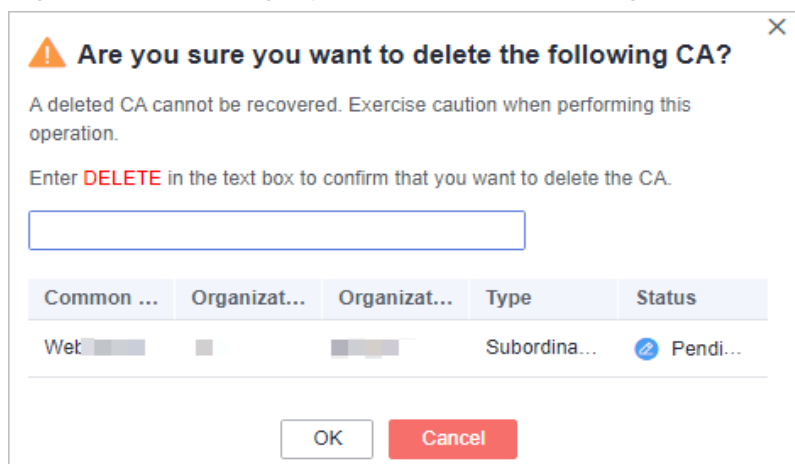
- Step 1** Log in to the [management console](#).
- Step 2** Locate the row of the private CA to be deleted and click **Delete** in the **Operation** column.

**Figure 2-10** Deleting a private CA

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
...	Root CA	...	...	2020/06/03 01:59:40 GMT+08:00	2021/06/03 02:00:40 GMT+08:00	Disabled	Enable Delete
...	Root CA	...	...	2020/02/28 22:16:41 GMT+08:00	2021/02/28 22:17:41 GMT+08:00	Disabled	Enable Delete

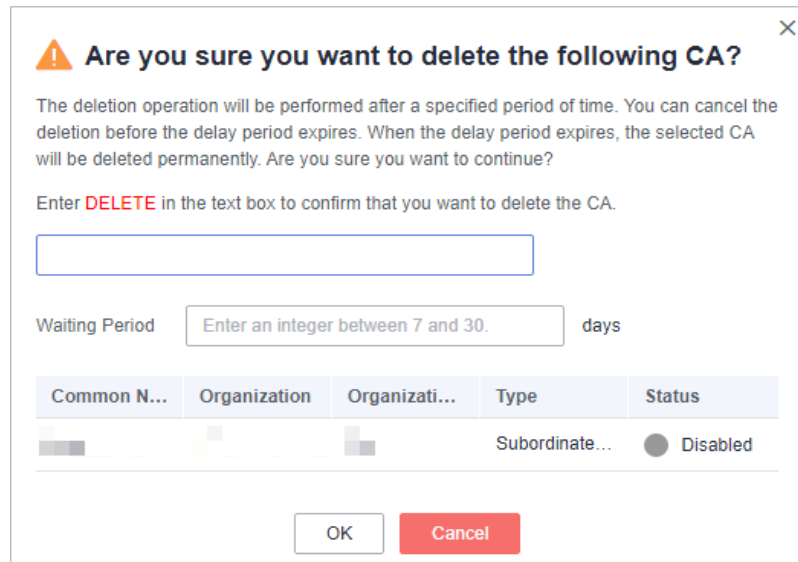
- Step 3** The operations vary according to the private CA status.
  - Private CA in the **Pending activation** state  
In the displayed dialog box, enter **DELETE** in the text box.

**Figure 2-11** Deleting a private CA in the **Pending activation** state



- Private CA in the **Disabled** or **Expired** state  
In the dialog box that is displayed, enter **DELETE** in the text box and configure the waiting period.

**Figure 2-12** Configuring the waiting period



**Step 4** Click **OK**.

- Private CA in the **Pending activation** state: If message "CA xxx deleted successfully." is displayed in the upper right corner of the page, the private CA is deleted successfully.
- Private CA in the **Disabled** or **Expired** state: If the private CA status changes to **Pending deletion**, the private CA will be deleted after the waiting period expires.

----End

## 2.9 Canceling the Deletion of a Private CA

This topic describes how to cancel the scheduled deletion of one or more private CAs prior to the real deletion. After the cancellation, the private CA is in the **Disabled** state.

### Prerequisites

The private CA for which you want to cancel the scheduled deletion is in **Pending deletion** status.

### Procedure

- Step 1** Log in to the [management console](#).
- Step 2** Locate the row of the desired private CA and click **Cancel CA Deletion** in the **Operation** column.

**Figure 2-13** Canceling the deletion of a private CA

Common Name	Type	Organizational Unit	Issued By	Creation Time	Expiration Time	Status	Operation
ca-180	Root CA			2020/06/15 16:36:25 GMT+08:00	2021/06/15 16:37:25 GMT+08:00	Pending deletion	Cancel CA Deletion
ca-18	Root CA			2020/06/11 17:53:59 GMT+08:00	2021/06/11 17:54:59 GMT+08:00	Pending deletion	Cancel CA Deletion

**Step 3** In the displayed dialog box, click **OK**.

If message "Deletion of CA xxx cancelled successfully." is displayed in the upper right corner of the page and the private CA status changes to **Disabled**, the deletion of the private CA is cancelled successfully.

After the deletion is canceled, if you want to use the private CA to issue certificates, you need to enable the private CA. For details, see [Enabling a Private CA](#).

----End

# 3 Private Certificate Management

## 3.1 Applying for a Private Certificate


After you create and activate a private CA, you can apply for private certificates from the private CA and use them for identity authentication, data encryption, and data decryption of internal applications.

This topic walks you through how to apply for a private certificate. You can apply for a maximum of 100,000 certificates.

### Prerequisites

You have created and activated a private CA. For details, see [Creating a Private CA](#) and [Activating a Private CA](#).


### Procedure

- Step 1** Log in to the [management console](#).
- Step 2** Click  in the upper left corner of the page and choose **Security > Cloud Certificate Management Service**. In the navigation pane, choose **Private Certificate Management**.
- Step 3** In the upper right corner of the private certificate list, click **Apply for Certificate**.
1. Select the CSR file generation method.

**Table 3-1** Certificate signing request (CSR)

Parameter	Description
System generated CSR	The system automatically generates a certificate private key. Once the certificate is issued, you can download your certificate and private key on the certificate management page.

Parameter	Description
Upload a CSR	<p>You can use an existing CSR. The procedure is as follows:</p> <ol style="list-style-type: none"> <li>1. You need to manually generate a CSR file and paste the content of the CSR file into the text box.</li> <li>2. Click <b>Parse</b>.</li> </ol>
<p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>- To obtain a certificate, a CSR file needs to be submitted to the CA for review. A CSR file contains a public key and a distinguished name (DN). Typically, a CSR file is generated by a web server, and a pair of public and private keys are created along with the CSR file.</li> <li>- You are advised to select <b>System generated CSR</b> to avoid approval failure caused by incorrect content.</li> <li>- A private key file will be generated when the CSR file is generated manually. Keep and back up your private key properly. A private key maps to a certificate. If a private key is lost, the corresponding certificate becomes invalid.</li> <li>- CCM has strict requirements on the key length of a CSR file. The key length must be 2,048 bits and the key type must be RSA.</li> </ul>	

2. Configure certificate details.  
Perform this step only when you select **System generated CSR** for **CSR**.  
**Common Name:** You can customize the name of the private certificate.
3. Click  on the right of **Advanced Configuration**.  
Perform this step only when you select **System generated CSR** for **CSR**.

**Table 3-2** Advanced settings

Parameter	Description	Example Value
Key Algorithm	<p><b>Key Algorithm:</b> Select the key algorithm and key size for the private certificate.</p> <p>The value can be <b>RSA2048</b>, <b>RSA4096</b>, <b>EC256</b>, or <b>EC384</b>.</p>	RSA2048
Signature Algorithm	<p>Select the signature hash algorithm for the private certificate.</p> <p>The value can be <b>SHA256</b>, <b>SHA384</b>, or <b>SHA512</b>.</p>	SHA256

Parameter	Description	Example Value
Key Usage	<p>Select the key usage of the certificate. You can select more than one option.</p> <ul style="list-style-type: none"> <li>- <b>digitalSignature</b>: The key is used as a digital signature.</li> <li>- <b>nonRepudiation</b>: The key can be used for non-repudiation.</li> <li>- <b>keyEncipherment</b>: The key can be used for key encryption.</li> <li>- <b>dataEncipherment</b>: The key can be used for data encryption.</li> <li>- <b>keyAgreement</b>: The key can be used as a key-agreement protocol.</li> <li>- <b>keyCertSign</b>: The key can be used to issue certificates.</li> <li>- <b>cRLSign</b>: The key can be used for signing blacklists.</li> <li>- <b>encipherOnly</b>: The key can be used for encryption only.</li> <li>- <b>decipherOnly</b>: The key can be used for decryption only.</li> </ul>	digitalSignature
Enhanced Key Usage	<p>Select the enhanced key usage for the certificate. You can select more than one option.</p> <ul style="list-style-type: none"> <li>- <b>Server identity authentication</b></li> <li>- <b>Client identity authentication</b></li> <li>- <b>Code signature</b></li> <li>- <b>Secure email</b></li> <li>- <b>Timestamp</b></li> </ul>	Server identity authentication
Customized Extension Field	Enter customized information.	None



Parameter	Description	Example Value
Configure Certificate AltName	<p>This field is optional. If you want to use the private certificate to multiple subjects, you can add more AltName records.</p> <p>You can configure <b>IP address</b>, <b>DNS</b>, <b>Email</b>, or <b>URI</b> for <b>AltName</b>. When you configure <b>AltName</b>, enter the value according to the value type you select.</p> <ul style="list-style-type: none"> <li>- <b>IP address</b>: Enter an IP address.</li> <li>- <b>DNS</b>: Enter the domain name.</li> <li>- <b>Email</b>: Enter an email address.</li> <li>- <b>URI</b>: Enter the network address.</li> </ul> <p>A maximum of five AltName records can be configured.</p>	None

4. Select a CA.

**Table 3-3** Parameters for selecting a CA

Parameter	Description
Common Name	Select a common name of the private CA you want.
Type	The CA type is autofilled after you specify <b>Common Name</b> .
CA ID	The CA ID is autofilled after you specify <b>Common Name</b> .
Validity Period	Configure the validity period of the private certificate.

**Step 4** Confirm the information and click **OK**.

After you submit your application, the system will return to the private certificate list page. Message "Certificate xxx applied for successfully." is displayed in the upper right corner of the page, indicating that the private certificate application is successful.

----End

### Follow-up Operations

When a private certificate is issued, you can download it and distribute it to the certificate subject for installation. For details, see [Downloading a Private Certificate](#).

## 3.2 Downloading a Private Certificate

Before using a private certificate, you need to download it. Only downloaded certificate can be assigned to the corresponding certificate subject so that they can install and use the certificate.

This topic describes how to download a private certificate. Only certificates in the **Issued** state can be downloaded.

### Prerequisites

Your private certificate is in the **Issued** state. For details, see [Applying for a Private Certificate](#).

### Procedure

- Step 1** Log in to the [management console](#).
  - Step 2** Locate the row of the desired private certificate and click **Download** in the **Operation** column.
  - Step 3** Click the target tab based on your server type and click **Download Certificate**.
- End

### Description of Downloaded Certificate Files

The downloaded certificate files vary depending on the CSR file type (**System generated CSR** or **Upload a CSR**) configured when you apply for a private certificate.

- **System generated CSR**  
[Table 3-4](#) describes the downloaded files.

**Table 3-4** Description of downloaded files (1)

Server Type	Files in the Package
Tomcat	<b>keystorePass.txt</b> : certificate password <b>server.jks</b> : certificate file
Nginx	<b>server.crt</b> : certificate files, containing the server certificate and certificate chain <b>server.key</b> : certificate private key file
Apache	<b>chain.crt</b> : certificate chain file <b>server.crt</b> : certificate file <b>server.key</b> : certificate private key file
IIS	<b>keystorePass.txt</b> : certificate password <b>server.pfx</b> : certificate file

Server Type	Files in the Package
Others	<b>chain.pem</b> : certificate chain file <b>server.key</b> : certificate private key file <b>server.pem</b> : certificate file

- **Upload a CSR**

[Table 3-5](#) describes the downloaded files.

**Table 3-5** Description of downloaded files (2)

Server Type	Files in the Package
Tomcat	<b>server.crt</b> : certificate file <b>chain.crt</b> : certificate chain file
Nginx	<b>server.crt</b> : certificate file
Apache	<b>server.crt</b> : certificate file <b>chain.crt</b> : certificate chain file
IIS	<b>server.crt</b> : certificate file <b>chain.crt</b> : certificate chain file
Others	<b>cert.pem</b> : certificate file <b>chain.pem</b> : certificate chain file

## 3.3 Installing a Private Certificate

### 3.3.1 Trusting a Private Root CA

Before installing a private certificate, you need to add the root CA to the trusted root certificate authorities of the client or server.

#### Prerequisites

You have created and exported a private root CA. For details, see [Exporting a Private CA Certificate](#).

#### Constraints

- One-way authentication  
To win more trust from the client for your server, you need to add the root CA that issue the server certificate to the client-end trusted CA store.
- Two-way authentication  
To enable two-way authentication between a server and a client, each side needs to add the root CA of the other side to their own trusted root CA store.

## Procedure

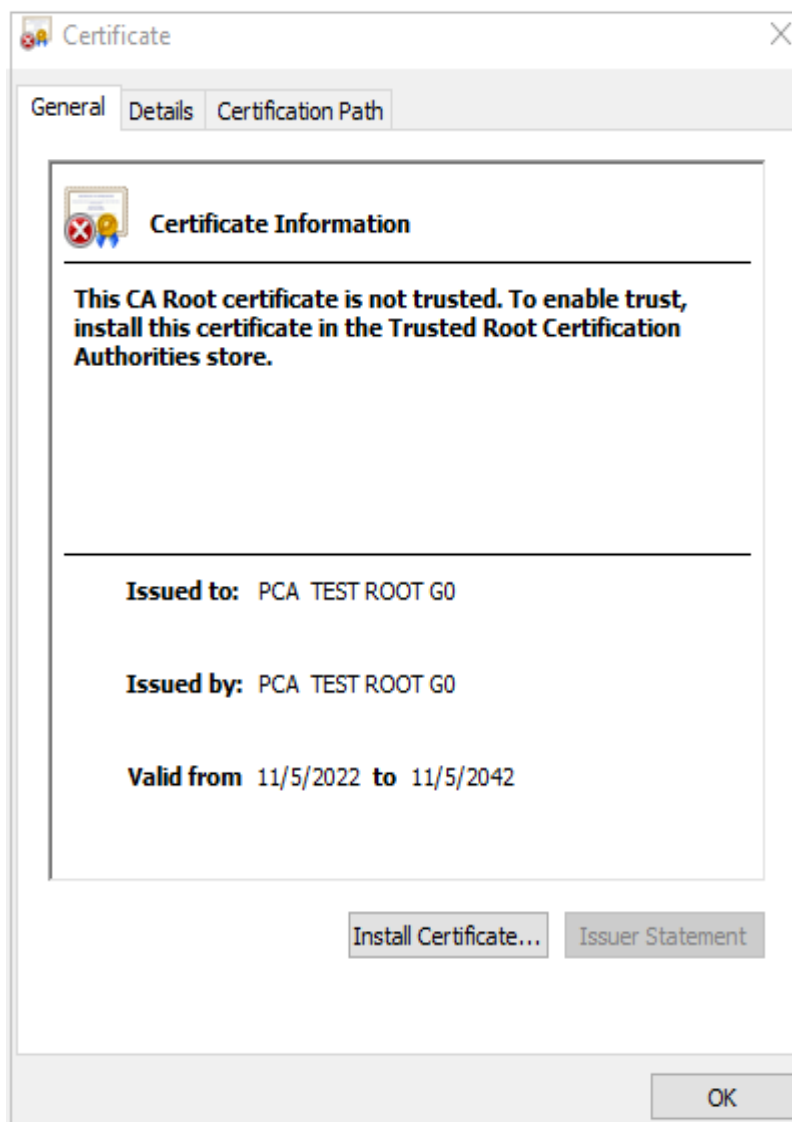
Use either of the following methods to add the root CA to trusted root certification authorities based on the operating system:

 **NOTE**

Root CA **PCA TEST ROOT G0** is used as an example.

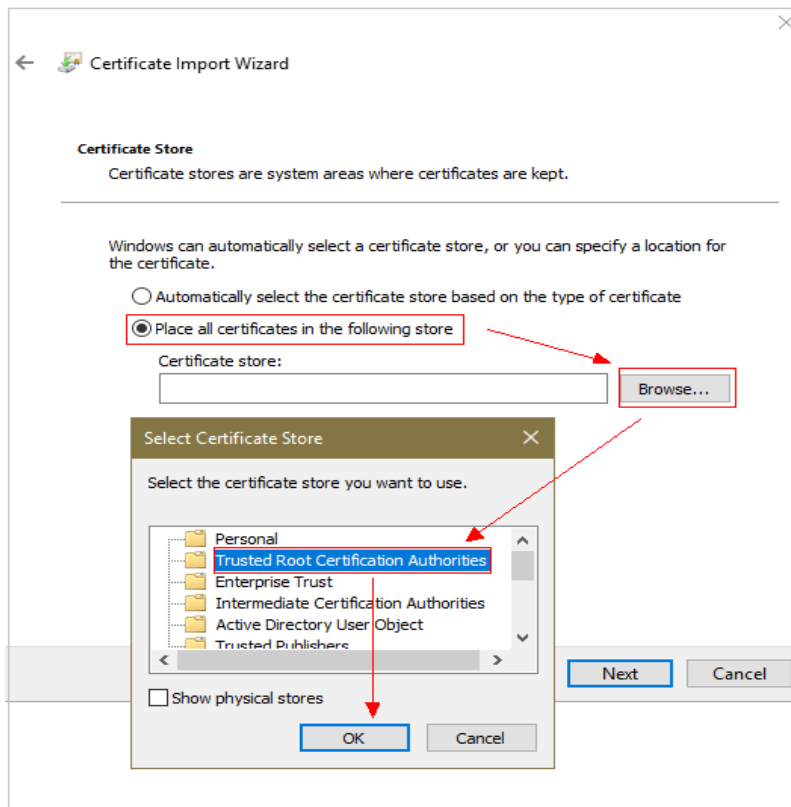
- **Windows**
  - a. Change the file name extension of the root CA certificate from .pem to .crt. and double-click the certificate file. The root CA certificate information shows that the root certificate is untrusted.

**Figure 3-1** Root CA not trusted



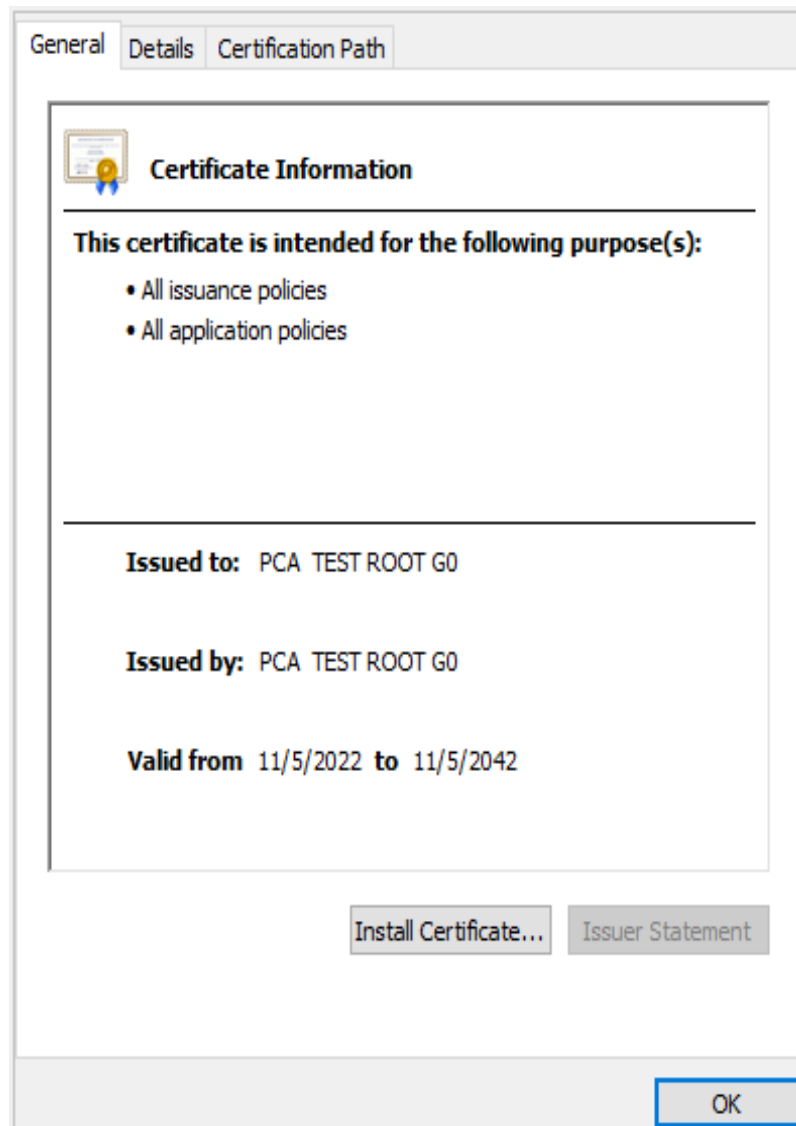
- b. Click **Install Certificate**, select a certificate storage location based on the certificate usage, and click **Next**.
- c. As shown in [Figure 3-2](#), select **Place all certificates in the following store** and click **Browse**. Then, select **Trusted Root Certification Authorities** and click **OK**.

Figure 3-2 Storing a root certificate



- d. Click **Next**, and then click **OK**. A dialog box is displayed, indicating that Windows will trust all certificates issued by the private root CA. Click **Yes**.
- e. Double-click the root CA certificate file. If the **Certificate Information** area shows that the system trusts the root CA certificate, the root CA is added to the trusted root CAs.

Figure 3-3 Trusted root CA



- **Linux**

The path for and method of storing root CA certificates vary depending on Linux OS versions. The following procedure use CentOS 6 as an example:

  - a. Copy the root CA certificate file to the **/home/** directory.
  - b. If **ca-certificates** is not installed on the server, run the following command to install **ca-certificates**:  
**yum install ca-certificates**
  - c. Copy the root CA certificate to the **/etc/pki/ca-trust/source/anchors/** directory:  
**cp /home/root.crt /etc/pki/ca-trust/source/anchors/**
  - d. Add the root CA certificate to the trusted root certificate file:  
**update-ca-trust extract**
  - e. Check whether the information about the newly added root CA certificate is included in the command output:  
**view /etc/pki/tls/certs/ca-bundle.crt**

Figure 3-4 Root CA certificate added to the trusted CA list



**NOTE**

If the OpenSSL version is too old, the configuration may not take effect. You can run the **yum update openssl -y** command to update the OpenSSL version.

- **macOS**
  - a. Open the macOS startup console and select **Keychain Access**.
  - b. Enter the password to log in to **Keychain Access**.
  - c. Drag and drop the target root CA certificate into **Keychain Access**. The root CA certificate now is untrusted by the system.
  - d. Right-click the root CA certificate to load its details.
  - e. Click **Trust**, select **Always Trust for When using this certificate**, and click **Close**.
  - f. Enter the password to make the configuration of the trusted root CA certificate take effect.
  - g. View the root CA certificate on the Keychain Access window. If the certificate is trusted by the system, the root CA is successfully added to the trusted root CA store.

### 3.3.2 Installing a Private Certificate on a Client

This topic describes how to install a private certificate on the client.

#### Prerequisites

You have downloaded an OpenSSH issued private certificate. For details about how to download a certificate, see [Downloading a Private Certificate](#).

## Constraints

If the server needs to verify the client certificate, you need to add the root CA of the client certificate to the trusted root CA store on the server. For details, see [Trusting a Private Root CA](#).

## Procedure

This procedure uses Windows as an example.



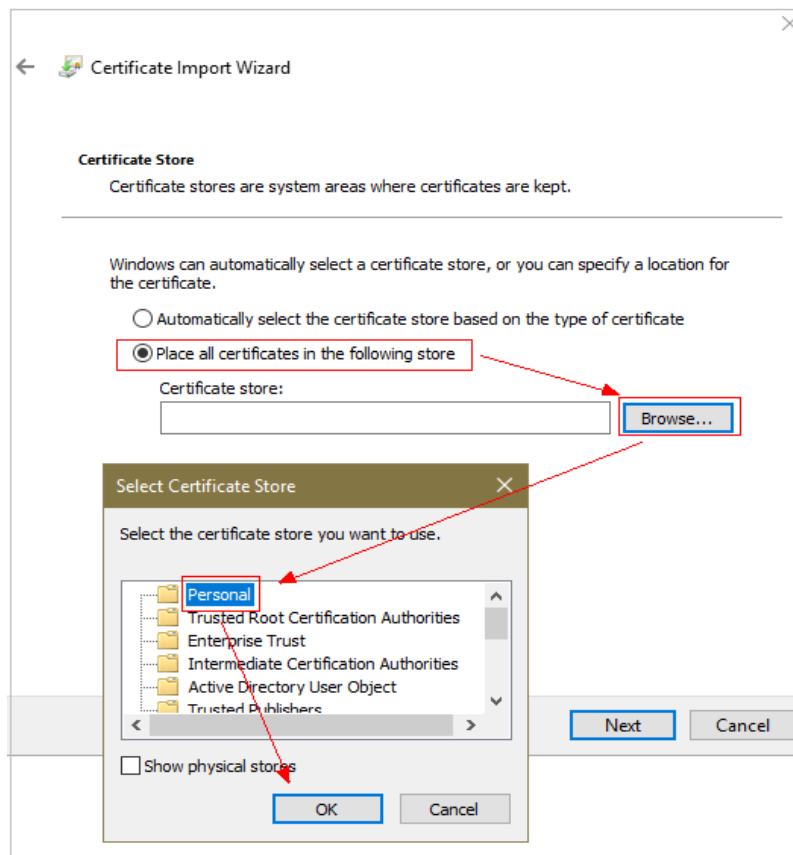
- Step 1** Log in to the [management console](#).
- Step 2** Click  in the upper left corner of the page and choose **Security > Cloud Certificate Management Service**. In the navigation pane, choose **Private Certificate Management > Private Certificate**.
- Step 3** Click **Service List** at the top of the page, and choose **Security > Cloud Certificate Manager**.
- Step 4** Click  in the upper left corner of the page and choose **Security > Cloud Certificate Management Service**. In the navigation pane, choose **Private Certificate Management**.
- Step 5** Locate the row containing the desired certificate. In the **Operation** column, click **Download**.
- Step 6** Select the **IIS** tab and click **Download Certificate**.
- Step 7** Decompress the downloaded certificate file package **client\_iis.zip** to obtain certificate file **server.pfx** and private key password file **keystorePass.txt**.
- Step 8** Double-click certificate file **server.pfx**, select a certificate storage location based on its usage, and click **Next**.
- Step 9** Confirm the name of the certificate file you want to import and click **Next**.
- Step 10** Enter the password obtained from private key password file **keystorePass.txt** and click **Next**.
- Step 11** Select **Place all certificates in the following store**, click **Browse**, select **Personal**, and click **OK**, as shown in [Figure 3-5](#).



Figure 3-5 Storing a private certificate



**Step 12** Click **Next** and **Finish**. The certificate is installed when a dialog box is displayed indicating that the certificate is imported successfully.

----End

### 3.3.3 Installing a Private Certificate on a Server

#### 3.3.3.1 Installing a Private Certificate on a Tomcat Server

This topic describes how to install a private certificate on a Tomcat 7 server running a Linux OS.

**NOTE**

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

#### Prerequisites

- The certificate has been issued.
- You have downloaded the private certificate in the format that is supported by Tomcat. For details, see [Downloading a Certificate](#).
- You have used a system-generated CSR to apply for the certificate.

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.
- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

The installation process is as follows (for Tomcat 7 servers):

[Step 1: Obtaining Files](#) → [Step 2: Creating a Directory](#) → [Step 3: Modifying Configuration Files](#) → [Step 4: Restarting the Tomcat](#) → [Verifying the Result](#)

### Step 1: Obtaining Files

Decompress the downloaded Tomcat certificate file to obtain the certificate file **server.jks** and password file **keystorePass.txt**.

### Step 2: Creating a Directory

Create a **cert** directory in the Tomcat installation directory, and copy the **server.jks** and **keystorePass.txt** files to the **cert** directory.

### Step 3: Modifying Configuration Files

---

#### NOTICE

Before modifying the configuration file, back up the configuration file. You are advised to deploy the configuration file in the test environment and then configure it on the production environment to avoid service interruptions caused by incorrect configurations.

---

The installation process is as follows (for Tomcat 7 servers):

1. Find the following parameters in the **server.xml** file in the Tomcat installation directory **conf**:

```
<!--  
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"  
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
    clientAuth="false" sslProtocol="TLS" />  
-->
```

2. Find the preceding parameters and delete the comment characters **<!--** and **-->**.
3. Add the following parameters. Change the values of the parameters according to [Table 3-6](#).

```
keystoreFile="cert/server.jks"
keystorePass="Certificate key"
```

The complete example configuration is as follows. Modify other parameters based on your needs.

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11Protocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    keystoreFile="cert/server.jks"
    keystorePass="Certificate key"
    clientAuth="false" sslProtocol="TLS" />
```

### NOTICE

Do not directly copy all configuration. Only parameters **keystoreFile** and **keystorePass** need to be added. Set other parameters based on site requirements.

**Table 3-6** Parameter description (1)

Parameter	Description
port	Port number to be used on the server. You are advised to set the value to <b>443</b> .
protocol	HTTP protocol. Retain the default value.
keystoreFile	Path for storing the <b>server.jks</b> file. The value can be an absolute path or a relative path. Example: <b>cert/server.jks</b>
keystorePass	<p>Password of <b>server.jks</b>. Set this parameter to the password provided in the <b>keystorePass.txt</b> file.</p> <p><b>NOTICE</b></p> <p>If the password contains <b>&amp;</b>, replace it with <b>&amp;amp;</b> to avoid configuration failure.</p> <p>An example command is provided as follows:</p> <p>If the password is <b>keystorePass="Ix6&amp;APWgCHf72DMu"</b>, change it to <b>keystorePass="Ix6&amp;amp;APWgCHf72DMu"</b>.</p>
clientAuth	Whether to require all customers to show the security certificate and authenticate their identity. Retain the default value.

- Find the following parameters in the **server.xml** file in the Tomcat installation directory **conf**:

```
<Host name="localhost" appBase="webapps"
    unpackWARs="true" autoDeploy="true">
```

- Change the value of **Host name** to the domain name bound to the certificate.

The complete configuration is as follows (**www.domain.com** is used as an example):

```
<Host name="www.domain.com" appBase="webapps"
    unpackWARs="true" autoDeploy="true">
```

6. Save the configuration file.

## Step 4: Restarting the Tomcat

Run the `./shutdown.sh` command in the `bin` directory of Tomcat to stop the Tomcat service.

After 10 seconds, run the `./startup.sh` command to start the Tomcat service. If the process is automatically started by the daemon process, you do not need to manually start the process.

## Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter `https://Domain name` and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

### 3.3.3.2 Installing a Private Certificate on an Nginx Server

This topic describes how to install a private certificate on an Nginx 1.7.8 server running CentOS 7.

#### NOTE

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

## Prerequisites

- The certificate has been issued.
- You have downloaded the private certificate in the format that is supported by Nginx. For details, see [Downloading a Certificate](#).
- You have used a system-generated CSR to apply for the certificate.

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.
- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

The installation process is as follows (for Nginx 1.7.8 servers running CentOS 7):

[Step 1: Obtaining Files](#) → [Step 2: Creating a Directory](#) → [Step 3: Modifying Configuration Files](#) → [Step 4: Verifying the Configuration](#) → [Step 5: Restarting Nginx](#) → [Verifying the Result](#)

## Step 1: Obtaining Files

Decompress the downloaded certificate file on your local PC.

You will obtain certificate file **server.crt** and private key file **server.key**.

- The **server.crt** file contains two segments of certificate code: -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, which are the server certificate and intermediate CA certificate respectively.
- **server.key** contains one segment of private key code: -----BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY-----.

## Step 2: Creating a Directory

Create a **cert** directory in the Nginx installation directory, and copy the **server.key** and **server.crt** files to the **cert** directory.

## Step 3: Modifying Configuration Files

### NOTICE

Before modifying the configuration file, back up the configuration file. You are advised to deploy the configuration file in the test environment and then configure it on the production environment to avoid service interruptions caused by incorrect configurations.

Configure the **nginx.conf** file in the **conf** directory of Nginx.

1. Find the following configuration:

```
#server {
#  listen    443 ssl;
#  server_name localhost;
#  ssl_certificate cert.pem;
#  ssl_certificate_key cert.key;
#  ssl_session_cache shared:SSL:1m;
#  ssl_session_timeout 5m;
#  ssl_ciphers HIGH:!aNULL:!MD5;
#  ssl_prefer_server_ciphers on;
#  location / {
#    root html;
#    index index.html index.htm;
#  }
#}
```

2. Delete comment tags (#) at the beginning of the lines.

```
server {
    listen    443 ssl;
    server_name localhost;
    ssl_certificate cert.pem;
    ssl_certificate_key cert.key;
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 5m;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;
```

```
location / {
    root    html;
    index  index.html index.htm;
}
}
```

3. Modify the following parameters according to [Table 3-7](#).

```
ssl_certificate    cert/server.crt;
ssl_certificate_key cert/server.key;
```

The complete configuration is as follows. Modify other parameters based on your needs.

```
server {
    listen    443 ssl; # Set the default HTTPS port to 443. If the default HTTPS port is not
    configured, Nginx may fail to start.
    server_name www.domain.com; #Replace www.domain.com with the domain name associated
    with your certificate.
    ssl_certificate    cert/server.crt; #Replace cert/server.crt with the path of the certificate file.
    ssl_certificate_key cert/server.key; #Replace cert/server.key with the path of the private key.
    ssl_session_cache  shared:SSL:1m;
    ssl_session_timeout 5m;
    ssl_ciphers HIGH:!aNULL:!MD5; #Encryption suite
    ssl_prefer_server_ciphers on;
    location / {
        root    html; #Site directory
        index  index.html index.htm; #Add attributes.
    }
}
```

**NOTICE**

Do not directly copy all configuration. Only attributes starting with **ssl** are directly related to the certificate configuration. Modify other parameters based on site requirements.

**Table 3-7** Parameters

Parameter	Description
listen	SSL access port number. Set the value to <b>443</b> . Set the default HTTPS port to 443. If the default HTTPS port is not configured, Nginx may fail to start.
server_name	Domain name which the certificate is used for. Example: www.domain.com
ssl_certificate	Certificate file <b>server.crt</b> Set the value to the path of the <b>server.crt</b> file. An example of the path is <b>cert/server.crt</b> .
ssl_certificate_key	Private key file <b>server.key</b> Set the value to the path of the <b>server.key</b> file. An example of the path is <b>cert/server.key</b> .

4. Save the configuration file.

## Step 4: Verifying the Configuration

Go to the execution directory of Nginx and run the following command:

```
sbin/nginx -t
```

If the following information is displayed, the configuration is correct.

```
nginx.conf syntax is ok  
nginx.conf test is successful
```

## Step 5: Restarting Nginx

Run the following command to restart Nginx to make the configuration take effect:

```
cd /usr/local/nginx/sbin  
./nginx -s reload
```

## Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter **https://*Domain name*** and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

### 3.3.3.3 Installing a Private Certificate on an Apache Server

This topic describes how to install a private certificate on an Apache 2.4.6 server running CentOS 7.

#### NOTE

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

## Prerequisites

- The certificate has been issued.
- You have downloaded the private certificate in the format that is supported by Apache. For details, see [Downloading a Certificate](#).
- You have used a system-generated CSR to apply for the certificate.

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.

- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

The installation process is as follows (for Apache 2.4.6 servers running CentOS 7):

[Step 1: Obtaining Files](#) → [Step 2: Creating a Directory](#) → [Step 3: Modifying Configuration Files](#) → [Step 4: Restarting Apache](#) → [Step 5: Verifying the Result](#)

### Step 1: Obtaining Files

Decompress the downloaded certificate file on your local PC.

You will obtain certificate files **ca.crt** and **server.crt** and private key file **server.key**.

- **ca.crt** contains one segment of intermediate CA certificate code: -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.
- **server.crt** contains one segment of server certificate code: -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.
- **server.key** contains one segment of private key code: -----BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY-----.

### Step 2: Creating a Directory

Create a **cert** directory in the Apache installation directory, and copy the **server.key**, **server.crt**, and **ca.crt** files to the **cert** directory.

### Step 3: Modifying Configuration Files

#### NOTICE

Before modifying the configuration file, back up the configuration file. You are advised to deploy the configuration file in the test environment and then configure it on the production environment to avoid service interruptions caused by incorrect configurations.

1. Open the **conf.d/ssl.conf** file in the Apache root directory.
2. Configure the domain name associated with the certificate.

Find and modify the following parameter:

```
ServerName www.example.com:443
```

The complete configuration is as follows (**www.domain.com** is used as an example):

```
ServerName www.domain.com:443 #Replace www.domain.com with the domain name of your server.
```

3. Configure the public key for the certificate.

Find and modify the following parameter:

```
SSLCertificateFile "${SRVROOT}/conf/server.crt"
```



Set the value to the path of the **server.crt** file. An example of the path is **cert/server.crt**.

The complete configuration is as follows:

```
SSLCertificateFile "cert/server.crt"
```

4. Configure the private key for the certificate.

Find and modify the following parameter:

```
SSLCertificateKeyFile "${SRVROOT}/conf/server.key"
```

Set the value to the path of the **server.key** file. An example of the path is **cert/server.key**.

The complete configuration is as follows:

```
SSLCertificateKeyFile "cert/server.key"
```

5. Configure the certificate chain.

Find and modify the following parameter:

```
#SSLCertificateChainFile "${SRVROOT}/conf/server-ca.crt"
```

Delete the comment tag **#** at the beginning of the line. Set this parameter to the path of the **ca.crt** file. An example of the path is **cert/ca.crt**.

The complete configuration is as follows:

```
SSLCertificateChainFile "cert/ca.crt"
```

6. Save the **ssl.conf** file and exit.

## Step 4: Restarting Apache

Restart the Apache service for the configuration to take effect:

1. Run the **service named stop** command to stop the Apache server.
2. Run the **service httpd start** command to start the Apache server.

## Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter **https://Domain name** and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

### 3.3.3.4 Installing a Private Certificate on an IIS Server

This topic describes how to install a private certificate on an IIS server.

#### NOTE

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

## Prerequisites

- The certificate has been issued.
- You have downloaded the private certificate in the format that is supported by IIS. For details, see [Downloading a Certificate](#).

- You have used a system-generated CSR to apply for the certificate.

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.
- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

To install a private certificate on an IIS server, perform the following steps:

[Step 1: Obtaining Files](#) → [Step 2: Configuring IIS](#) → [Step 3: Verifying the Result](#)

### Step 1: Obtaining Files

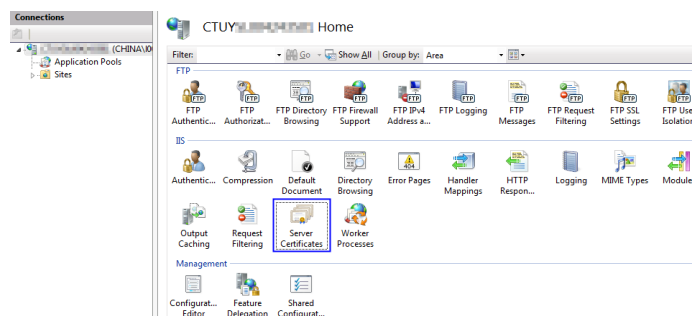
Decompress the downloaded certificate file on your local PC.

You will obtain certificate file **server.pfx** and password file **keystorePass.txt**.

### Step 2: Configuring IIS

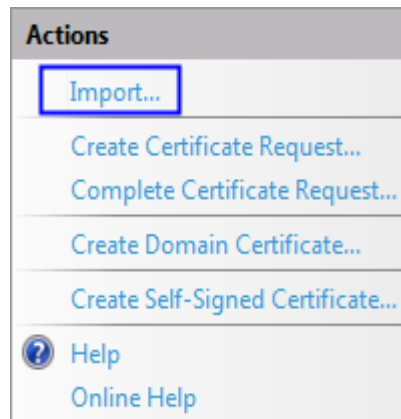
1. Install IIS as instructed by IIS guides.
2. Open the IIS management console, double-click **Server Certificates**.

**Figure 3-6** Double-clicking Server Certificates



3. In the displayed dialog box, click **Import**.

Figure 3-7 Import

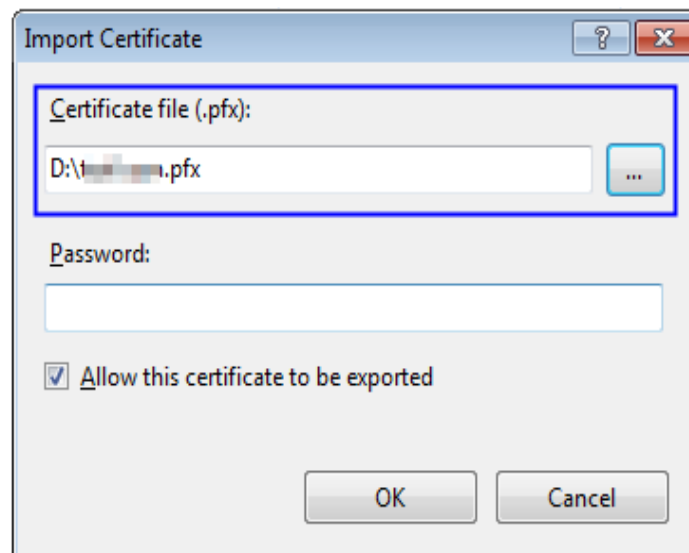


4. Import the **server.pfx** certificate file. Then click **OK**.

 **NOTE**

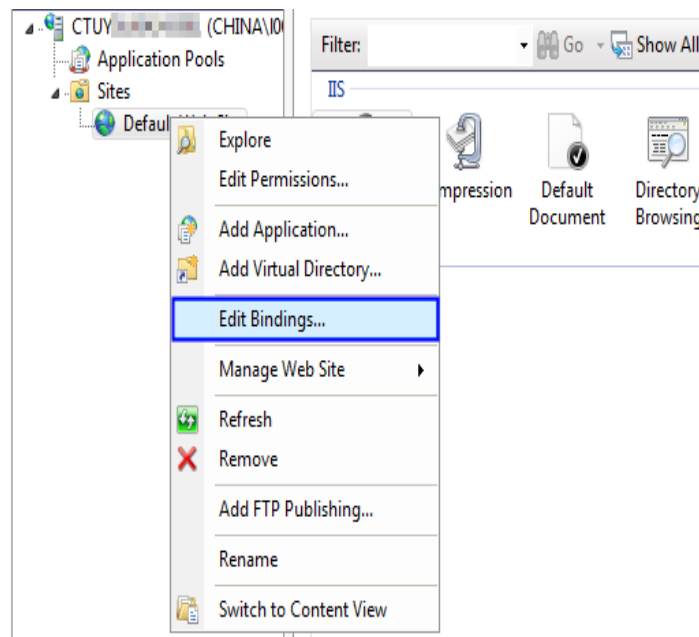
In the **Password** box, enter the password provided in the **keystorePass.txt** file.

Figure 3-8 Importing a PFX certificate file



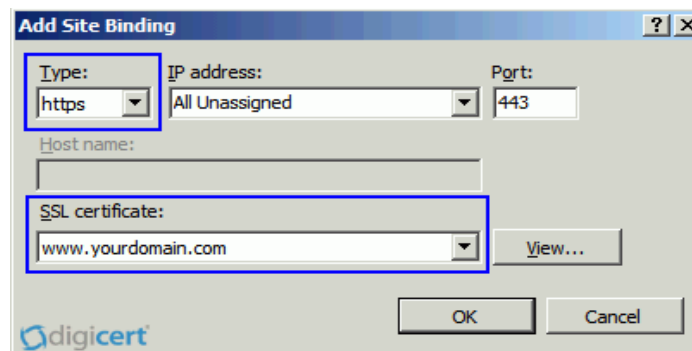
5. Right-click the target site (the default site is used as an example). Choose **Edit Bindings** from the shortcut menu.

Figure 3-9 Choosing Edit Bindings



6. In the dialog box that is displayed, click **Add**. Then enter the following information.

Figure 3-10 Binding a website



- **Type:** Select **https**.
  - **Port:** Retain the default port **443**.
  - **SSL certificate:** Select the certificate imported in **4**.
7. Click **OK**.

### Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter **https:// Domain name** and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

### 3.3.3.5 Installing a Private Certificate on a WebLogic Server

WebLogic is a Java EE application server, used to develop, integrate, deploy, and manage large-scale distributed Web apps, network apps, and database apps. It

applies dynamic functions of Java and security of the Java Enterprise standard to the development, integration, deployment, and management of large-scale network applications.

Currently, WebLogic 10.3.1 and later versions support SSL certificates of all mainstream brands. Versions earlier than WebLogic 10.3.1 do not support SSL certificates of brands.

This topic describes how to install a private certificate on a Weblogic server.

#### NOTE

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

## Prerequisites

- The certificate is in the **Issued** status.
- You have downloaded the private certificate in the format that is supported by Tomcat. For details, see [Downloading a Certificate](#).
- You have used a system-generated CSR to apply for the certificate.
- The JDK has been installed.

The JDK has been installed after WebLogic installation is complete. If the JDK has not been installed, install the [Java SE Development Kit \(JDK\)](#).

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.
- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

To install a private certificate on a WebLogic server, perform the following steps:

[Step 1: Obtaining Files](#) → [Step 2: Configuring WebLogic](#) → [Verifying the Result](#)

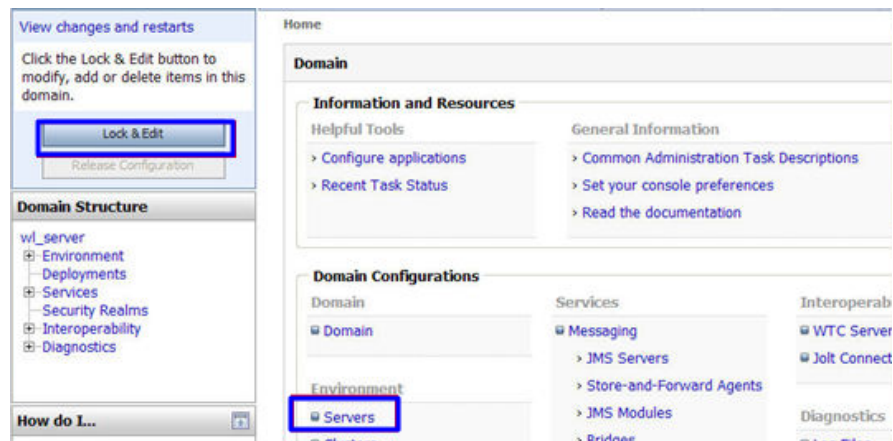
### Step 1: Obtaining Files

Decompress the downloaded Tomcat certificate file to obtain the certificate file **server.jks** and password file **keystorePass.txt**.

## Step 2: Configuring WebLogic

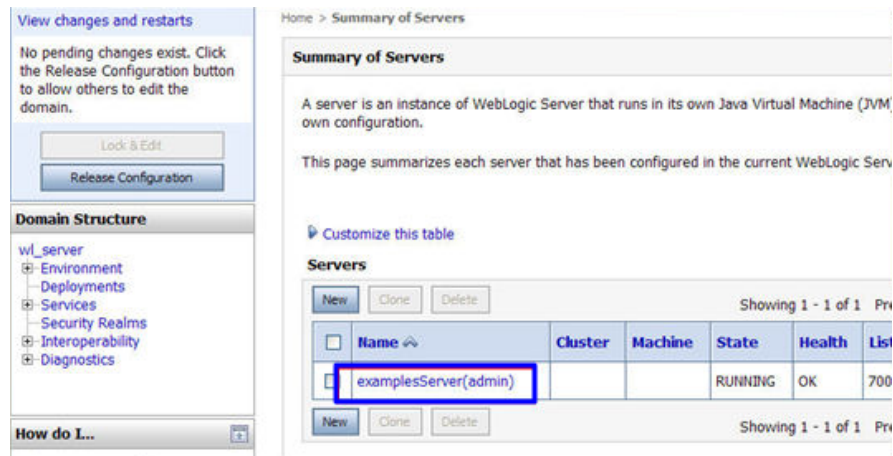
1. Log in to the management console of the WebLogic server.
2. Click **Lock & Edit** in the upper left corner of the page to unlock the configuration.
3. Click **Servers** in **Domain Configurations**.

Figure 3-11 Server



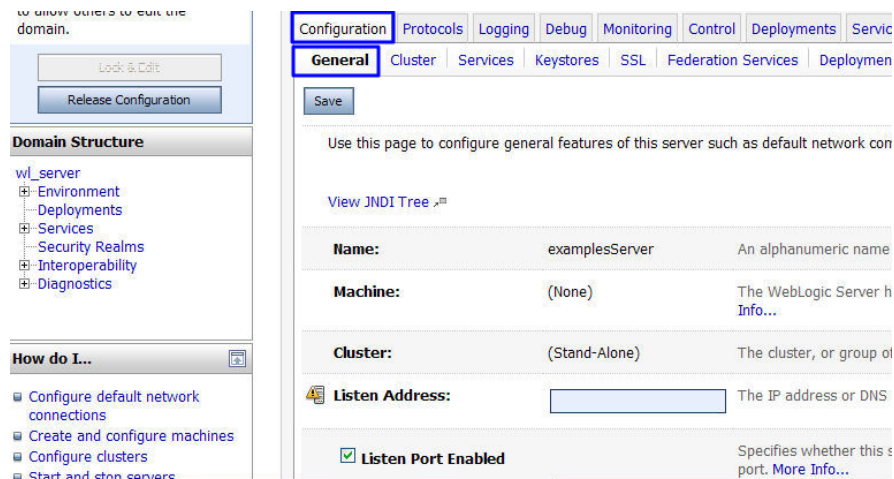
4. In the server list, select the server for which you want to configure the server certificate. The server configuration page is displayed.

Figure 3-12 Target server



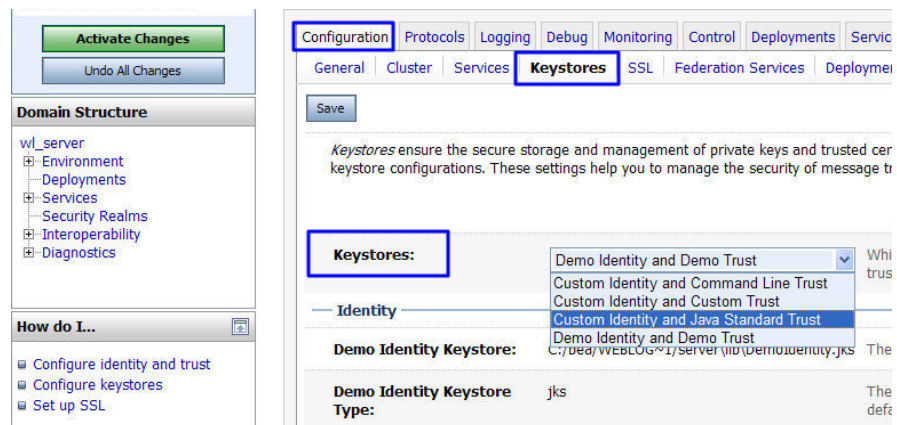
5. Modify the HTTPS port.  
On the server configuration page, click the **General** tab and configure whether to enable HTTP and HTTPS and the access port number.  
Select **Listen SSL Port Enabled** and change the port number to **443**.

Figure 3-13 port



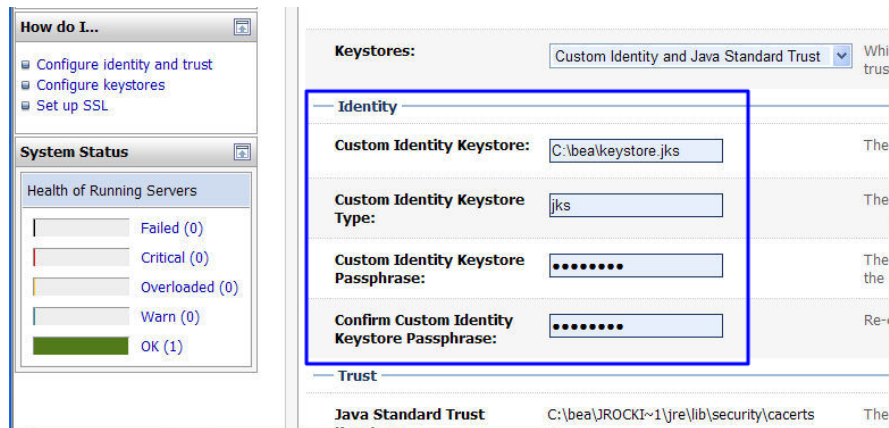
6. Configure an authentication mode and a key.
  - a. On the server configuration page, click the **Keystores** tab and configure an authentication mode.

Figure 3-14 Authentication mode



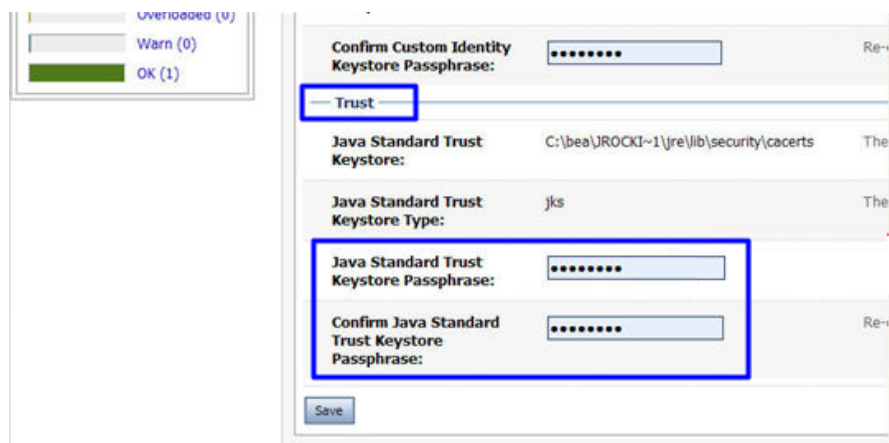
- Select **Custom Identity and Java Standard Trust** for server authentication.
    - Select **Custom Identity and Custom Trust** for bidirectional authentication.
  - b. Configure a key in the **Identity** area.  
Configure the path for storing the keystore file **server.jks** on the server and enter the password of the keystore file.

Figure 3-15 Key



- **Custom Identity Keystore:** Enter the path for storing the .jks file.  
Example: C:\bea\server.jks
  - **Custom Identity Keystore Type:** Set the file format to **jks**.
  - **Custom Identity Keystore Passphrase:** Enter the certificate password, that is, the password in **keystorePass.txt**.
  - **Confirm Custom Identity Keystore Passphrase:** Re-enter the certificate password.
- c. In unidirectional authentication, configure the default trust store file **cacerts** of the JRE.  
The default password of **cacerts** is **changeit**.

Figure 3-16 Trust store file

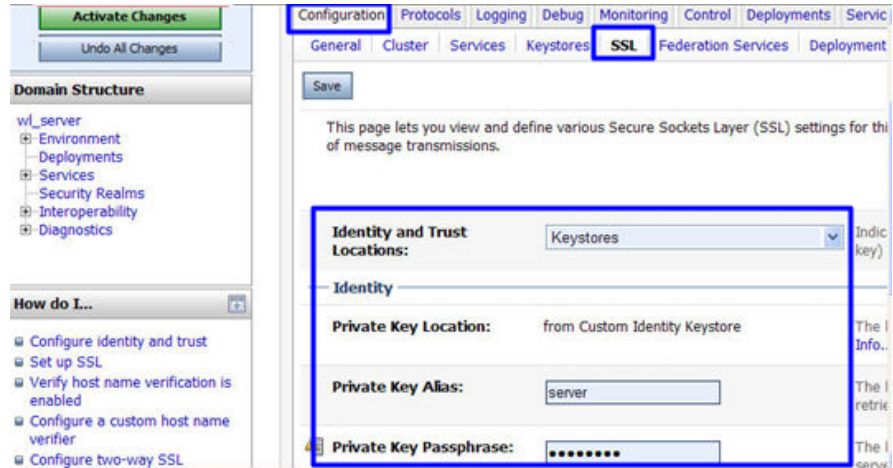


- **Java Standard Trust Keystore Passphrase:** Enter the default password **changeit**.
  - **Confirm Java Standard Trust Keystore Passphrase:** Re-enter the default password.
7. Configure the private key alias of the server certificate.



On the server configuration page, click the **SSL** tab and set the following parameters:

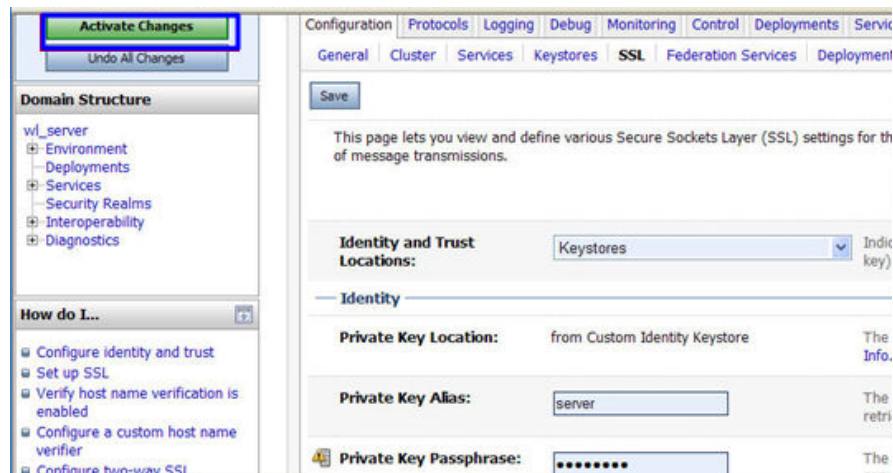
**Figure 3-17** Private key



- **Identity and Trust Locations:** Select **Keystores**.
- **Private Key Alias:** Configure a private key alias in the private key library. You can run the **keystool -list** command to view the private key alias.
- **Private Key Passphrase:** Enter the private key protection password. Generally, the private key protection password is the same as the keystore file protection password.
- **Confirm Private Key Passphrase:** Enter the private key protection password again.

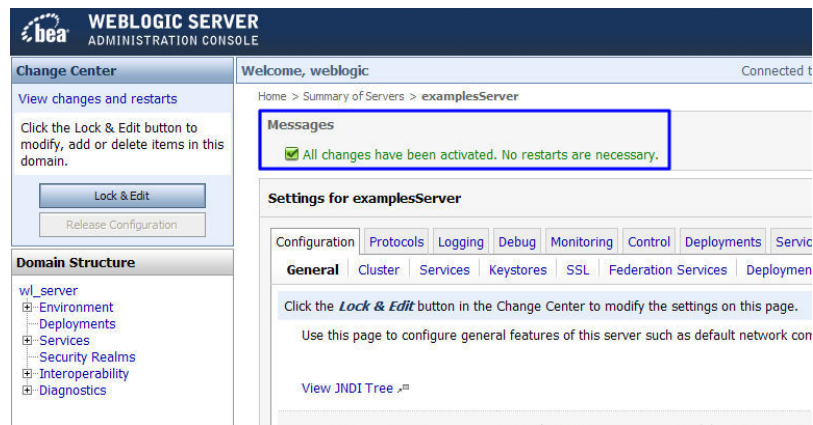
8. Click **Active Changes** to save the settings.

**Figure 3-18** Saving the settings



9. (Optional) If the system prompts you to restart the WebLogic server, restart the WebLogic server for the settings to take effect. As shown in **Figure 3-19**, you do not need to restart the WebLogic server.

Figure 3-19 Message displayed



## Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter **https:// Domain name** and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

### 3.3.3.6 Installing a Private Certificate on a Resin Server

This topic describes how to install a private certificate on a Resin server.

#### NOTE

The installation procedure in this topic is for your reference only as the commands executed and configuration file modified during the installation may vary depending on OS types and server configurations.

## Prerequisites

- The certificate is in the **Issued** status.
- You have downloaded the private certificate in the format that is supported by Tomcat. For details, see [Downloading a Certificate](#).
- You have used a system-generated CSR to apply for the certificate.

## Constraints

- Before installing the certificate, enable port 443 on the server where the private certificate is installed and add port 443 to the security group. Otherwise, HTTPS cannot be enabled after the installation.
- A root CA must be added to the trusted client CA list so that all server certificates issued by the root CA can be trusted by the client. For details, see [Trusting a Private Root CA](#).
- If a domain name maps to multiple servers, deploy the certificate on each server.
- A private certificate can only be installed on the server that maps to the domain name associated with the certificate. Otherwise, the web browser will display a message indicating that the domain name is insecure.

## Procedure

To install a private certificate on a Resin server, perform the following steps:

[Step 1: Obtaining Files](#) → [Step 2: Configuring Resin](#) → [Verifying the Result](#)

### Step 1: Obtaining Files

Decompress the downloaded Tomcat certificate file to obtain the certificate file **server.jks** and password file **keystorePass.txt**.

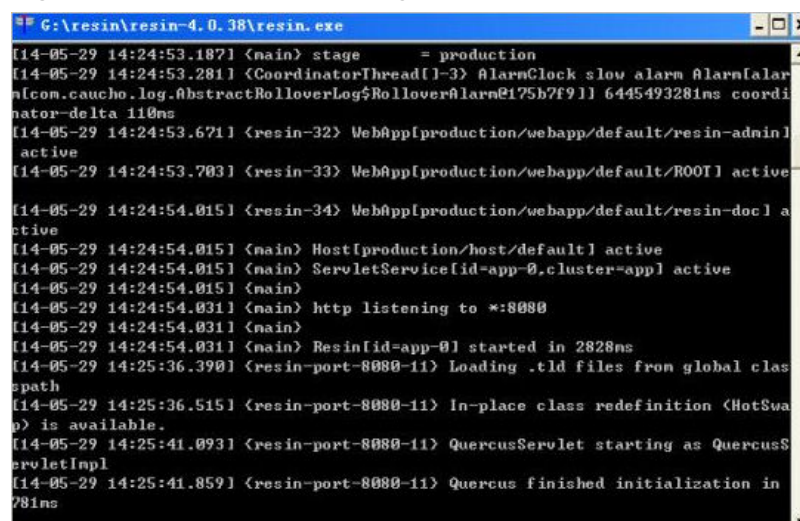
### Step 2: Configuring Resin

#### NOTICE

Before modifying the configuration file, back up the configuration file. You are advised to deploy the configuration file in the test environment and then configure it on the production environment to avoid service interruptions caused by incorrect configurations.

1. (Optional) Install Resin.  
If you have installed Resin, skip this step.
  - a. Log in to the [Resin](#) official website and download the appropriate application packages for your operating system.  
The following uses **Resin-4.0.38** for Windows as an example.
  - b. Decompress the downloaded Resin software package.
  - c. Access the root directory of Resin-4.0.38 and find the **resin.exe** file.
  - d. Run the **resin.exe** file. During the execution, the command prompt window [Figure 3-20](#) will display.

Figure 3-20 Information dialog box



```
G:\resin\resin-4.0.38\resin.exe
[14-05-29 14:24:53.187] <main> stage = production
[14-05-29 14:24:53.281] <CoordinatorThread[1-3] AlarmClock slow alarm Alarm[alar
n/com.caucho.log.AbstractRolloverLog$RolloverAlarm@175b7f91] 6445493281ms coordi
nator-delta 110ms
[14-05-29 14:24:53.671] <resin-32> WebApp[production/webapp/default/resin-admin]
active
[14-05-29 14:24:53.703] <resin-33> WebApp[production/webapp/default/ROOT] active
[14-05-29 14:24:54.015] <resin-34> WebApp[production/webapp/default/resin-doc] a
ctive
[14-05-29 14:24:54.015] <main> Host[production/host/default] active
[14-05-29 14:24:54.015] <main> ServletService[id=app-0,cluster=appl] active
[14-05-29 14:24:54.015] <main>
[14-05-29 14:24:54.031] <main> http listening to *:8080
[14-05-29 14:24:54.031] <main>
[14-05-29 14:24:54.031] <main> Resin[id=app-0] started in 2828ms
[14-05-29 14:25:36.390] <resin-port-8080-11> Loading .tld files from global clas
spath
[14-05-29 14:25:36.515] <resin-port-8080-11> In-place class redefinition <HotSwa
p> is available.
[14-05-29 14:25:41.093] <resin-port-8080-11> QuercusServlet starting as QuercusS
ervletImpl
[14-05-29 14:25:41.859] <resin-port-8080-11> Quercus finished initialization in
281ms
```

- e. After the resin.exe file is executed. Start the Microsoft Internet Explorer, enter the default address **http://127.0.0.1:8080** of Resin in the address bar, and then press **Enter**.

If the information similar to **Figure 3-21** is displayed, Resin is installed successfully.

**Figure 3-21** Logging In to Resin



2. Modify the configuration file.
  - a. Find the following parameters in the **Resin.properties** configuration file in the Resin installation directory (the configuration file may be **resin.xml** for different Resin versions):

```
# specifies the --server in the config file
# home_server : app-0

# Set HTTP and HTTPS bind address
# http_address : *

# Set HTTP and HTTPS ports.
# Use overrides for individual server control, for example: app-0.http : 8081
app.http      : 8080
# app.https   : 8443

web.http      : 8080
# web.https   : 8443
```

- b. Delete the comment symbol (#) before **app.https** and **web.https**. Then modify port **8443** to **443**. After the modification:

**app.https** and **web.https**: Port to be used on the server. You are advised to set the value to **443**.

```
# specifies the --server in the config file
# home_server : app-0

# Set HTTP and HTTPS bind address
# http_address : *

# Set HTTP and HTTPS ports.
# Use overrides for individual server control, for example: app-0.http : 8081
app.http      : 8080
app.https     : 443

web.http      : 8080
web.https     : 443
```

- c. Find the following parameters and delete the comment symbol (#) before **jsse\_keystore\_type**, **jsse\_keystore\_file**, and **jsse\_keystore\_password**.

```
# JSSE certificate configuration
# Keys are typically stored in the resin configuration directory.
jsse_keystore_tye : jks
jsse_keystore_file : cert/server/jks
jsse_keystore_password : certificate password
```

- d. Modify certificate-related parameters. For details, see [Table 3-8](#).

```
# JSSE certificate configuration
# Keys are typically stored in the resin configuration directory.
jsse_keystore_tye : jks
jsse_keystore_file : cert/server.jks
jsse_keystore_password: certificate password
```

**Table 3-8** Description

Parameter	Description
jsse_keystore_tye	Type of the Keystore file. Generally, this parameter is set to <b>jks</b> .
jsse_keystore_file	Path for storing the <b>server.jks</b> file. The value can be an absolute path or a relative path. Example: <b>cert/server.jks</b>
jsse_keystore_passwo rd	<p>Password of <b>server.jks</b>. Set this parameter to the password provided in the <b>keystorePass.txt</b> file.</p> <p><b>NOTICE</b></p> <p>If the password contains <b>&amp;</b>, replace it with <b>&amp;amp;</b>; to avoid configuration failure.</p> <p>An example command is provided as follows:</p> <p>If the password is <b>keystorePass="Ix6&amp;APWgcHf72DMu"</b>, change it to <b>keystorePass="Ix6&amp;amp;APWgcHf72DMu"</b>.</p>

- e. Save the configuration file.

3. Restart Resin.

## Verifying the Result

After the deployment succeeds, in the address bar of the browser, enter **https://Domain name** and press **Enter**.

If a security padlock is displayed in the address bar of the browser, the certificate has been installed successfully.

## 3.4 Revoking a Private Certificate

If a private certificate is no longer needed or its private key is lost before it expires, you can revoke it on the console. If a private certificate is revoked, it is no longer trusted within the organization.

If a private certificate is revoked, the billing stops.

The following describes how to revoke a private certificate.


### Prerequisites

The private certificate is in the **Issued** state.

## Constraints

- After you apply for revoking a private certificate, your application cannot be withdrawn. Exercise caution when performing this operation.
- All its records will be cleared and cannot be recovered, including private CA records. Therefore, exercise caution when performing this operation.

## Procedure

- Step 1** Log in to the [management console](#).
- Step 2** Click  in the upper left corner of the page and choose **Security > Cloud Certificate Management Service**. In the navigation pane, choose **Private Certificate Management**.
- Step 3** Locate the row of the desired private certificate and click **Revoke** in the **Operation** column.
- Step 4** In the displayed dialog box, enter **REVOKE** and select the revocation reason to confirm the revocation. The default revocation reason is in the **UNSPECIFIED** field. [Table 3-9](#) describes the revocation reasons you can select.

**Table 3-9** Revocation reasons and meaning

Reason for Revocation	Reason Code in RFC 5280	Description
UNSPECIFIED	0	Default value. No reason is specified for revocation.
KEY_COMPROMISE	1	The certificate key material has been leaked.
CERTIFICATE_AUTHORITY_COMPROMISE	2	Key materials of the CA have been leaked in the certificate chain.
AFFILIATION_CHANGED	3	The subject or other information in the certificate has been changed.
SUPERSEDED	4	The certificate has been replaced.
CESSATION_OF_OPERATION	5	The entity in the certificate or certificate chain has ceased to operate.

Reason for Revocation	Reason Code in RFC 5280	Description
CERTIFICATE_HOLD	6	The certificate should not be considered valid currently and may take effect in the future.
PRIVILEGE_WITHDRAWN	9	The certificate no longer has the right to declare its listed attributes.
ATTRIBUTE_AUTHORITY_COMPROMISE	10	The authority that warrants the attributes of the certificate may have been compromised.

**Step 5** Click **OK**.

When "Certificate xxx revoked successfully" is displayed in the upper right corner of the page, and the private certificate status changes to **Revoked**, the private certificate is revoked successfully.

----End

## 3.5 Viewing Details of a Private Certificate

This topic describes how to view details of a private certificate, including the common name, expiration time, and status.

### Prerequisites


You have applied for a private certificate. For details, see [Applying for a Private Certificate](#).

### Procedure

**Step 1** Log in to the [management console](#).

**Step 2** View the private certificate information. [Table 3-10](#) describes the private certificate parameters.

#### NOTE

- Select a certificate state from the drop-down list of **All statuses**. Then the certificate list displays only the private certificates in the corresponding state.
- Enter a name of a private certificate in the search box in the upper right corner and click  or press **Enter** to search for a specified private certificate.

**Table 3-10** Private certificate parameters

Parameter	Description
Common Name	Indicates the name of the private certificate configured during certificate application.
Issued By	Indicates the name of the private CA that issues the private certificate.
Creation Time	Indicates the time when a private certificate is created.
Expiration Time	Indicates the time when a certificate expires.
Status	Indicates the certificate status. The value can be: <ul style="list-style-type: none"> <li>• <b>Issued</b> The private certificate is issued.</li> <li>• <b>Expired</b> The private certificate is expired.</li> <li>• <b>Revoked</b> The private certificate is revoked.</li> </ul>
Operation	You can download, revoke, or delete the certificate.

**Step 3** Click the common name of a private certificate to view its details.

You can click **Add Tag** on the private certificate details page to identify the private certificate. TMS's predefined tag function is recommended for adding the same tag to different cloud resources.

----End

## 3.6 Deleting a Private Certificate

This topic describes how to delete a private certificate. A deleted private certificate remains valid and trusted.

You can delete a certificate that is no longer needed.

### Prerequisites

The private certificate is in the **Issued**, **Expired**, or **Revoked** state.

### Constraints

- A deleted certificate cannot be restored. Exercise caution with the deletion.
- After you submit a certificate deletion application, you cannot cancel it. Exercise caution when performing this operation.

### Procedure

**Step 1** Log in to the [management console](#).



- Step 2** Locate the row of the private certificate to be deleted and click **Delete** in the **Operation** column.
  - Step 3** In the displayed dialog box, enter **DELETE** to confirm the deletion.
  - Step 4** Click **OK**. If message "Certificate xxx deleted successfully." is displayed in the upper right corner of the page, the private certificate is deleted successfully.
- End

# 4 Permissions Management

---

## 4.1 Creating a User and Granting CCM Permissions to the User

This topic describes how to use IAM to implement fine-grained permissions control for your CCM resources. With IAM, you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials, providing access to CCM resources.
- Grant only the permissions required for users to perform a task.
- Entrust an account or cloud service to perform professional and efficient O&M to your CCM resources.

If your account does not require individual IAM users, skip this chapter.

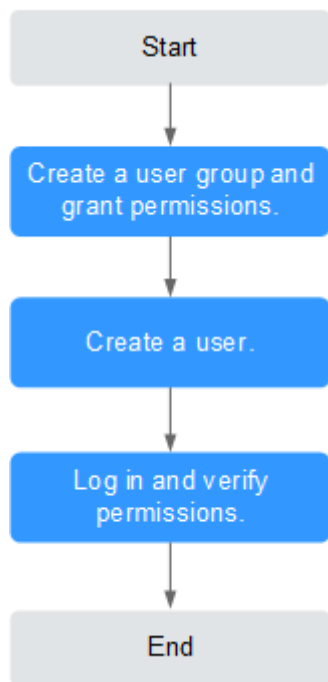
This section provides some methods for you to assign permissions to a user. [Figure 4-1](#) shows the process.

### Prerequisites

Before authorizing permissions to a user group, you need to know which CCM permissions can be added to the user group.

## Process Flow

**Figure 4-1** Process for granting CCM permissions



1. Create a user group and assign permissions.  
Create a user group on the IAM console and grant the user group the **PCA FullAccess**.
2. .  
Create a user on the IAM console and add the user to the group created in **1**.
3. .  
Log in to the CCM console by using the created user, and verify that the user only has read permissions for CCM.  
Choose **Cloud Certificate Management Service** under **Security** in the **Service List**. If no message appears indicating that you have no permissions to access the service, the policy **PCA FullAccess** has already taken effect.

## 4.2 CCM Custom Policies

Custom policies can be created to supplement the system-defined policies of CCM.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

This section contains examples of common CCM custom policies.

## Example CCM Custom Policies

- Example 1: authorizing users to create a CA

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "pca:ca:create"
      ],
      "Effect": "Allow"
    }
  ]
}
```

- Example 2: denying certificate deletion

A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

If you need to assign permissions of the **PCA FullAccess** policy to a user but you want to prevent the user from deleting certificates, you can create a custom policy for denying certificate deletion, and attach both policies to the group that the user belongs to. Then, the user can perform all operations on certificates except deleting certificates. The following is an example of a deny policy:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "pca:ca:delete"
      ],
      "Effect": "Deny"
    }
  ]
}
```

# A Change History

---

Released On	Description
2023-12-15	This issue is the first official release.