Install Certificate Manager server components on Linux

This article describes how to install Smart ID Certificate Manager (CM) server components on Linux.

The CM servers comprises several sub-components that may be loaded into the same computer or distributed to several computers in order to obtain higher performance. See this list of CM server components.

This article is valid for Certificate Manager 8.4 and later.

Related information

- Smart ID Certificate Manager
- CM server components
- Install Certificate Manager

Expand/Collapse All

Prerequisites

Make sure you have access to the license file before starting the installation.

Step-by-step instruction

1. To make **install_server.sh** executable, issue this command:
   ```
   chmod a+x install_server.sh
   ```
2. Execute **install_server.sh**.
3. Enter the full installation path, for example:

Example: Full installation path
/opt/cm/server

   Warning

   ⚠️ If CM clients are already installed on this computer, the server installation must not be made to the same directory.
4. Confirm the path.
5. Enter the full configuration files path, for example:

Example: Full configuration file path
/opt/cm/server

6. Confirm the path.
7. Enter the name of the license file.
8. Confirm the file name.
9. Enter the number or name of each server component, for which you have a license, separated by space (" "). See this list of CM server components.

   ⚠️ The CMDB database must already exist and the JDBC component should be included to create the database connection parameters.

   If CF is selected without the CIS component, the CF will automatically be configured to run CIS locally within the CF service. If both the CF and CIS components are selected, CF will be automatically configured to connect to the installed independent CIS service (see Configure the Certificate Factory in Certificate Manager).

Warning

⚠️ Install all server components at the same time. It is not possible to add new components later. Instead you will have to uninstall existing components before you can reinstall. (See Uninstall Certificate Manager server components and clients.)

If you have a license for the functions Key Archive and Recovery or Card Production Manager but are not going to use them, you must set the following parameters in **cm.conf**:

```
CardProductionManager.start=false
KARFactory.start=false
```

10. Select the database engine from the list.
11. Select the machine where the database is installed. Default is *localhost*.
12. Select JDBC port for the database.
13. Enter the database instance name.
14. Enter the name of the database CM user. Default is *lcmreq*.
15. Enter the password for the *lcmreq* user.

Message after installation

1. Await a message from the installation program confirming a successful installation. If an error message occurs, correct the indicated error and restart the installation using the shell script *setup.sh* located in the directory ***<install_root>/install***.
2. Option: Start CM SNMP monitor: `<install_root>/bin/cmsnmp start`.

   If the CM SNMP monitor is installed, set the configuration parameter `*.agent.connectToSupervisor` to 'true' in the configuration files **cis.conf** and **cm.conf**.

   ```
   *.agent.connectToSupervisor = true
   ```

Option: Move configuration files

After the installation, it is possible to manually move the configuration files to any other path. Do this before you start any of the CM server components.

1. Modify each ***<install_root>/bin/*_launch.conf*** file and point to a new configuration path:

Example
`SERVICEARGUMENTS='-p <new_configuration_path>'`

2. The service user used to run the CM components must have both read and write access rights to the new configuration path.

It is recommended to execute the CM services as a separate non-root service user. Do the following:

1. Make sure the CM services are stopped.
2. Create the user (for example, `useradd cmuser`).
3. Make the service user owner of the directory ***<install_root>*** . For example, by issuing the following command:

   ```
   chown -R cmuser:cmuser /opt/cm/server/
   ```

4. Update the files ***<install_root>/bin/*_launch.conf*** so that they specify a line with `SERVICEUSER`, for example:

```
SERVICEUSER=cmuser
```

This can be scripted using a command such as:

```
sed -i -e 's/# SERVICEUSER/SERVICEUSER/' /opt/cm/server/bin/*_launch.
conf
```

The *<install_root>/bin/cmservices* script can be used to install or remove the CF, CIS and CMSNMP services as system services.

`sudo` is required to run the *cmservices* script.

1. Run the following command to install the CM services as system services. You must specify the user and group that the service should be run as.

   Example: Install CM services as system services

   ```
   sudo <install_root>/bin/cmservices install cf cmuser cmuser
   ```

   a. This script can also be used to start, stop or check the status of any of the CM services, see an example here for CF:

      Example: See the status for CF

      ```
      sudo ./cmservices status cf
      ```

2. Start the services:

   ```
   sudo service cmsnmp start
   sudo service cis start
   sudo service cf start
   ```

3. Check the log files in *<configuration_root>/logs* to see that the services started successfully. If not, see the following steps for troubleshooting.
4. The CM server components will use the first found Java executable available on the $PATH. To use a specific Java version, or if $PATH does not include a directory where Java can be found, you can explicitly specify which Java to use by setting a parameter in **<install_root>/bin /*_launch.conf**:

   ```
   JAVA=/usr/jvm/jdk-default/bin/java
   ```

5. When installing CM services on a system that uses systemd, it is possible that the default $PATH as seen by the systemd service wrappers, does not include the expected Java version. If this is the case, then invoking, for example, `sudo service cis start` might not start the service - it will instead be reported as active (exited) by `sudo service cis status`. To remedy this, see the previous step on how to explicitly set the Java executable to use.