UNICORN Systems

entsoe

ENTSO-E

Energy Communication Platform

Installation Guide v4.4.0

Jankovcova 1037/49, CZ – 170 00 Prague 7

Project:	ENTSO-E
Project – Subject:	Energy Communication Platform
Document Title:	Installation Guide v4.4.0
Date:	1.11.2016
Author:	Jakub Eliáš, Aleš Holý, Zdeněk Pospíšil, Josef Brož, Jiří Dudek, Jiří Neuman
Contact:	E-mail: info@unicornsystems.eu Tel.: (+420) 221 400 111

1. Executive Summary

This document provides installation instructions for all ECP components.

2. Content

3. Revision History

Version	Date	Author	Description
4.1.4	05.02.2018	Laura Betcherová	Documentation review
4.2.0	23.03.2018	Petr Novák	Automatic start of service after server startup
4.3.0	26.06.2018	Josef Brož	Version update
4.3.0	24.09.2018	Barbora Klabačková	Reverse proxy usage for ECP CD Endpoint/CD uninstallation via administration console Java upgrade description
4.3.0	05.10.2018	Barbora Klabačková	Documentation review
4.3.0	18.10.2018	Barbora Klabačková	Documentation review
4.4:0	05.06.2019	Drahoslav Stejskal	External MySQL Database – Create Schema update
4.4.0	05.06.2019	Barbora Kánská	Support forward proxy update

4. General Information

4.1 ECP introduction

4.1.1 ECP network overview

The ECP network consists of multiple ECP endpoints, component directories, and brokers. This guide is focused on installation of these components.

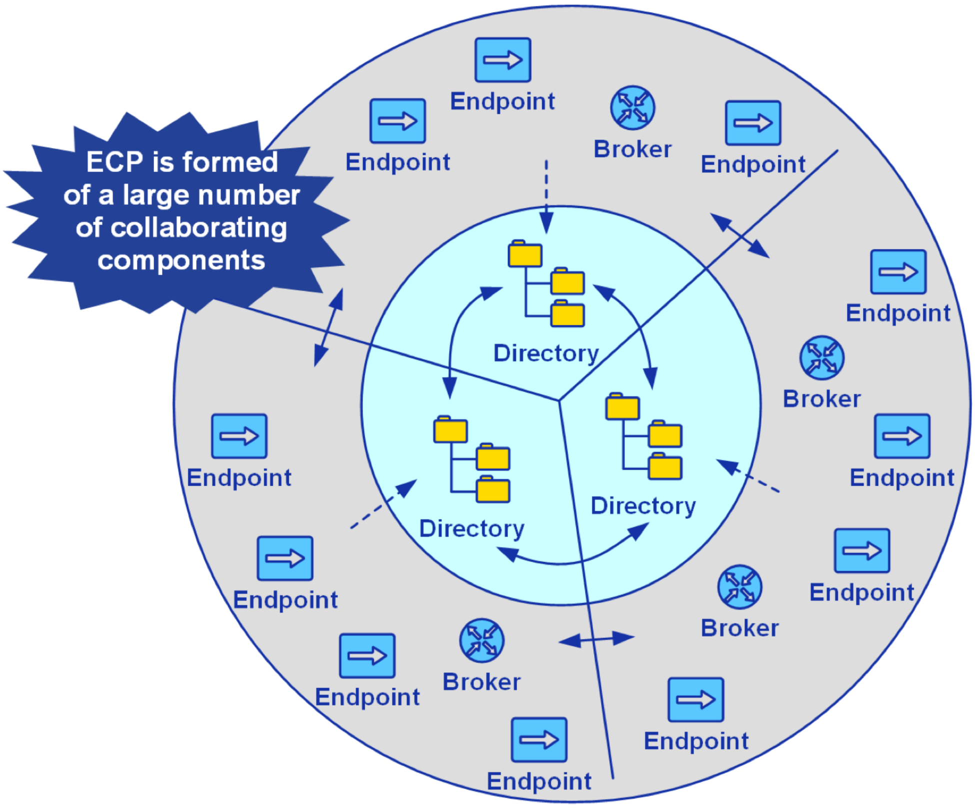


Figure 1: Distributed architecture

4.1.2 ECP network communication protocols

ECP uses two main protocols for messaging – AMQPS and HTTPS. Please note that the port numbers specified below are default, however different ports may be explicitly configured for ECP depending on the installation environment.

Following is the list of all ports that must be open to allow ECP communication:

- ECP endpoint
  - WAN
    - Outgoing
      - 5671 - to ECP brokers
      - 5671 - to endpoints with enabled direct communication (only if direct communication is enabled)

        - 8443 - to ECP component directories
    - Incoming
        - 5671 – only when direct communication is enabled
- Internal network
    - Incoming
        - 5672 – AMQP API (only when the AMQP channel is enabled)
        - 8080 – Webservice API, GUI and monitoring
    - Outgoing
        - <db-port> – optionally to the selected external database in HA deployment

ECP component directory

- WAN
    - Outgoing
        - 8443 – to other ECP component directories in the network
    - Incoming
        - 8443 – synchronization REST API exposed for other components
- Internal network
    - Incoming
        - 8080 – GUI and monitoring
    - Outgoing
        - <db-port> – optionally to the selected external database in HA deployment

ECP broker

- WAN
    - Outgoing
        - 8443 - to ECP component directories
    - Incoming
        - 5671 – AMQPS port for messaging
- Internal network
    - Incoming
        - 8161 – Management GUI and monitoring
    - Outgoing
        - <db-port> – optionally to the selected external database in HA deployment

4.1.3 ECP network deployment modes

The ECP components can be installed using several modes. Before starting an installation, it is important to choose one of the following installation modes:

> Standalone deployment
> High availability (HA) deployment
> Clustered high availability deployment

4.1.3.1 Standalone deployment

In standalone mode, no other services are required (such as a database, however it is still possible to configure use of an external database). This is the simplest installation option. Please note that high availability is not possible in this mode.

4.1.3.2 High availability deployment

High availability deployment consists of a shared external database and two ECP component deployments. In this mode, when one ECP component goes down, the system will continue to function. Please note that for HA deployment two IPs in WAN are required in the following scenarios:

- ECP broker
- ECP endpoint with enabled direct communication

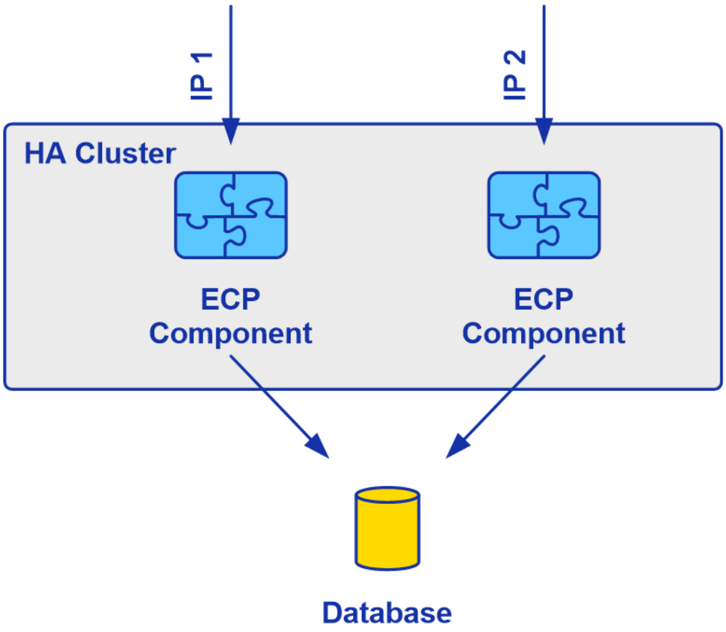


Figure 2 : HA deployment

If you also want to have HA at the database level it is possible to deploy MySQL to the cluster using Galera cluster or by using other solution that works for your database.

4.1.3.3 Clustered high availability deployment

Clustered high availability deployment is possible only for endpoints that are required to send and receive so many messages that two application servers do not have enough performance to process them. Please note that in this mode it is not possible to use direct messaging.

Clustered HA deployment consists of:

- N application servers with ECP endpoint installed
  These endpoints cannot use embedded active ActiveMQ
- Two servers with ActiveMQ – replace the embedded ActiveMQ broker
- An external database (e.g. MySQL)

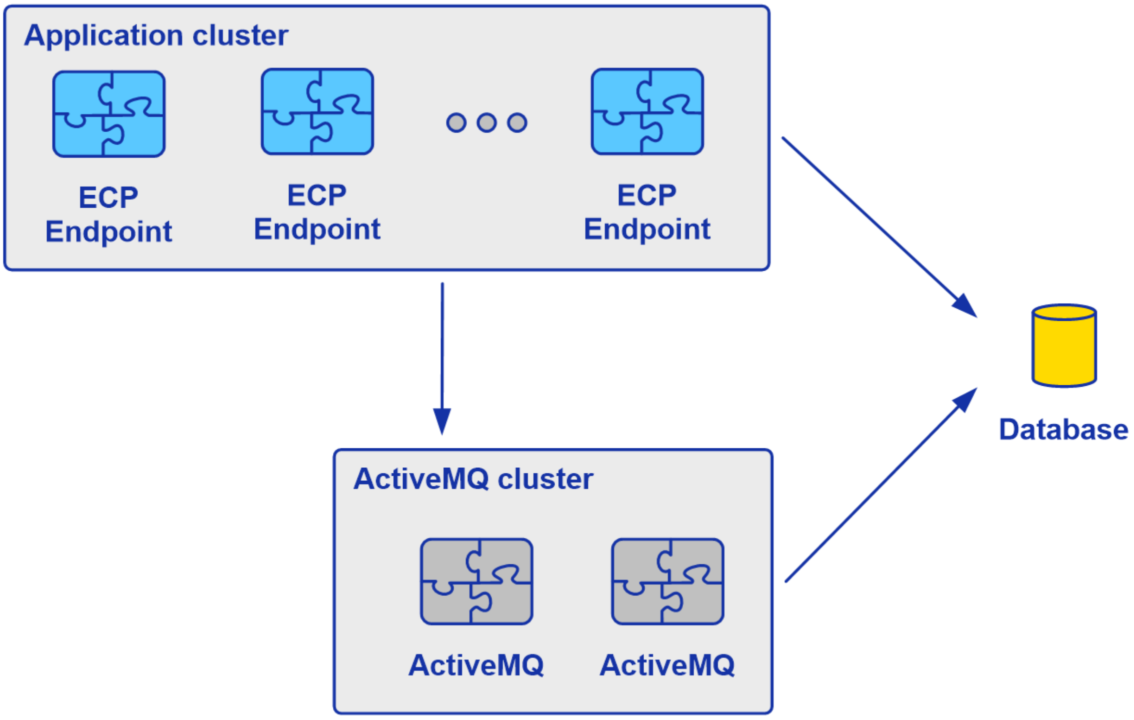


Figure 3 : Clustered HA deployment

5. ECP Endpoint Installation

5.1 Common prerequisites

5.1.1 General prerequisites before installing an ECP endpoint

- ECP endpoint installer (jar or rpm package, depending on the deployment and OS type)
- Registration keystore (this keystore will be provided by your home component directory administrator)
- Hardware prerequisites can be found in **ECP 4 System Design, Appendix C: Hardware Requirements**.
- General software requirements can be found in **ECP4 System Design, chapter 14: Service Level Requirements**.

5.1.2 Firewall configuration

Please ensure that your firewall allows communication as described in 4.1.2 ECP network communication protocols

5.2 Installation on Windows

The application may be installed in either command line or GUI mode. Both modes are described in this chapter.

5.2.1 Variables used in the guide

Several variables are used throughout this installation guide using the <variable_name> format. These variables are used only within this guide and are not to be confused with variables of your OS.

Name	Description
<jre_install_path>	Location of the JRE 8 used by the application
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name
<ecp-jar>	Path to the ECP endpoint installation package, typically ecp-endpoint-<version>.jar

5.2.2 Prerequisites

Before processing with the installation, check that the following prerequisites are met.

- ECP endpoint Windows installer (jar file).
- Java Runtime Environment 8 64-bit with Java Cryptography Extension (JCE) is required.
- Both Oracle's OpenJDK 8 and Oracle's Java SE 8 versions are supported. Oracle's OpenJDK Java is recommended for Organizations which do not have Oracle Java SE binaries available as a part of a commercial product or service from Oracle.
- Check current Java version using the command:

  java -version

- Expected output:

  java version "openjdk 1.8.0.181"

or

  openjdk version "1.8.0_181"

- If the version does not start with "1.8" or the output is anything else:
  - Download and install Java Runtime Environment 8 for your OS from the Oracle website (for OpenJDK visit http://jdk.java.net/ or https://adoptopenjdk.net/).

  - After installation, it is required to set JRE_HOME system variable. Open This PC, select Computer -> Properties -> Advanced system settings -> Environment Variables -> System Variables -> New -> Name: JRE_HOME, value: path to jre (e.g. C:\Program Files\Java\jre1.8.0.181).
- If the version is 1.8.0.181 or later
  - Open the file <jre_install_path>\jre\lib\security\java.security
  - Find the line *#crypto.policy=unlimited*
  - Change the line to *crypto.policy=unlimited* and save the file
- If the version is not 1.8.0.181 or later
  - Download JCE from http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
  - Unzip the file into <jre_install_path>\jre\lib\security. All files should be overwritten.
- For Oracle's OpenJDK Java Cryptography Extension (JCE) is installed by default and no additional installation steps are required

The installer verifies that JCE is available at the beginning of the installation process. If this check fails, a warning message is displayed without aborting the installation.

Please make sure that MMC (Microsoft Management Console) or Services snap-in is not running during the installation as it prevents successful registration of the ECP service.

As the Java is upgraded to the different version after ECP Endpoint installation, the Windows service has to be modified to the new JVM path. To do so, please, use the command and check the Java tab:

> <install_path>/tomcat/bin/tomcat9w.exe //ES//ecp-endpoint

5.2.3 Installation

5.2.3.1 Installation using GUI

1. Double-click the <ecp-jar> file. If Java was registered to execute jar files, an installer window should open. Skip to step 5. If an installer window does not open follow steps 2-4:

2. Open command line with administrator rights:

   2.1. Start > type "cmd" > right click > run as administrator

3. Change directory to location of ecp-endpoint.jar, e.g.

   3.1. cd c:\ecp-installer

4. Run the ecp-endpoint.jar file using

   4.1. java -jar <ecp-jar>

5. ECP endpoint installer window should open.

   5.1. Select installation type, whether ECP Endpoint application should be installed as a standalone application or should be installed to an existing application server.

   5.2. Select target installation path

   5.3. Next

   5.4. Choose whether the ECP Endpoint application should be installed as a Windows service.

   5.5. Next

   5.6. Choose whether the ECP Endpoint application should be automatically started after server start. The preferred option is to not start ECP Endpoint after server start.

   5.7. Next > Next

   5.8. If desired, an automatic installation script can be generated. It can be used in the future to install the application using the same options.

5.9. Done

6. Configure the ECP endpoint (see 5.7 ECP endpoint configuration)

5.2.3.2 Installation using the command line

1. Open command line with administrator rights:

> Start > type “cmd” > right click > run as administrator

2. Change directory to the location of <ecp-jar>, e.g.

> cd c:\ecp-installer

3. Run the ecp-endpoint.jar file using

> java -jar <ecp-jar> -console

4. Follow the console installation wizard
5. Configure the ECP endpoint (see 5.7 ECP endpoint configuration)

5.2.4 Starting and stopping ECP endpoint

5.2.4.1 Using GUI

a) If installed as a service:

> Start > Run > “services.msc”
> Find service with name like " ecp-endpoint " and description "ECP endpoint"
> Start / stop service by buttons on the top left

b) If not installed as a service:

> Start / stop application by using
- <install_path>\tomcat\bin\startup.bat
- <install_path>\tomcat\bin\shutdown.bat

5.2.4.2 Using the command line

a) If installed as a service:

> Start service using
- SC start ecp-endpoint

> Stop service using
- SC stop ecp-endpoint

b) If not installed as a service:

> Start application using
- <install_path>\tomcat\bin\startup.bat

> Stop application using
- <install_path>\tomcat\bin\shutdown.bat

5.2.5 Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain configuration files, Tomcat folder, and uninstaller. After the application is successfully started, it creates additional folders for data and log files. Default names and locations of these folders are <install_path>\data and <install_path>\logs.

5.2.5.1 Service status

The status of the service, if installed, can be checked either via command line:

> SC query ecp-endpoint

or in the Windows Services tool (see section 5.2.4.1)

5.2.5.2 Log files

Check the application log files for more information about its status. The default location is <install_path>\logs.

5.3 Uninstallation on Windows

Stop the application and run the command line as an administrator. Navigate to <installation directory>\Uninstaller and execute the following command:

- java -jar uninstaller.jar -c

Uninstaller will remove all application components and deregister the service if present.

5.4 Installation on Linux (non-HA)

The following section describes the installation of ECP endpoint on the Linux platform. Supported Linux distributions:

- CentOS 7
- RHEL 7

5.4.1 Variables

Several variables are used throughout this installation guide using the <variable_name> format. These variables are used only within this guide and are not to be confused with variables of your OS.

Name	Description
<jre-rpm>	Path to the JRE 8 installation package
<jre_install_path>	Location of the JRE 8 used by application
<ecp-rpm>	Path to the ECP endpoint installation package
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name

5.4.2 Prerequisites

- Download the ECP endpoint rpm file and copy it to the server.
- Oracle Java Runtime Environment 8 64-bit with Java Cryptography Extension (JCE) is required.
- Both Oracle's OpenJDK 8 and Oracle's Java SE 8 versions are supported. Oracle's OpenJDK Java is recommended for Organizations which do not have Oracle Java SE binaries available as a part of a commercial product or service from Oracle.
  - Check current Java version using the command:

    java -version
  - Expected output example:

    java version "openjdk 1.8.0.181"

or

    openjdk version "1.8.0_181"

  - If the version does not start with "1.8" or the output is anything else, download and install Java Runtime Environment 8 x64 rpm package from the Oracle website (for Oracle's OpenJDK visit http://jdk.java.net/ or https://adoptopenjdk.net/).
    - Install the JRE:
      rpm -i <jre-rpm>
  - (Oracle`s Java SE 8 only) If the version is 1.8.0.181 or later
    - Open the file <jre_install_path>\jre\lib\security\java.security

  - Find the line *#crypto.policy=unlimited*
  - Change the line to *crypto.policy=unlimited* and save the file
- (Oracle`s Java SE 8 only) If the version is not 1.8.0.181 or later
  - Download JCE from http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
  - Unzip the file into <jre_install_path>\jre\lib\security. All files should be overwritten.
- (Oracle`s OpenJDK 8 only) For Oracle's OpenJDK Java Cryptography Extension (JCE) is installed by default and no additional installation steps are required.
- Please note that ECP rpm file has dependency to specified by jre therefore installed Java must provide jre. Java headless OpenJDK distribution can be also used, however it will not pass the rpm dependency check. To overcome this issue, install rpm with - -nodeps option

> ca-certificates – The Mozilla CA root certificate bundle is expected to be installed on OS
- use yum install ca-certificates to install certificate bundle

> tzdata-java – This package contains timezone information for use by Java runtimes.
- Use yum install tzdata-java to install time zone information

5.4.3 ECP application installation

5.4.3.1 Install using RPM package

1. Install ECP endpoint. Change the directory to the folder where the ECP rpm installer is located and then invoke rpm install.

    rpm -i <ecp-rpm>

Installer automatically creates the following folder structure:

Folder	Description
/usr/share/ecp-endpoint	Application binaries folder. This is the default location of <install_path>
/var/lib/ecp-endpoint	Application data folder
/var/log/ecp-endpoint	Application log folder
/var/log/ecp-directory	Application log folder
/etc/ecp-endpoint	ECP configuration files (ecp.properties)
/etc/default/ecp-endpoint/	Default configuration of ECP endpoint. Used to set JVM configuration and location of configuration files. (file /etc/default/ecp-endpoint/ecp.defaults)

5.4.3.2 Install using JAR installer

Installation with JAR installer allows you to install multiple instance of ECP on a single machine. However, in this scenario you must change ports for Tomcat and ActiveMQ so each instance will have different ports.

> Run the installer as a java executable

java –jar endpoint-installer.jar

> Follow the installer wizard.

> Configure the ECP endpoint.

> If required, ECP Endpoint application can be set up to automatically start after server start.
  - chkconfig ecp-endpoint on

Please note that installer does not support registering ECP endpoint as a system service. Please use the RPM installation package in this case.

5.4.4 Starting and stopping ECP endpoint

After ECP endpoint is configured, it can be started. ECP endpoint is registered as a Linux service. The following commands are available:

> To start ECP, use command "*service ecp-endpoint start*"
> To stop ECP, use command "*service ecp-endpoint stop*"
> To display the ECP logs, use command "*service ecp-endpoint log*"

If not installed as a service:

> Start application using bash script "*<install_path>/tomcat/bin/startup.sh*"
> Stop application using bash script "*<install_path>/tomcat/bin/shutdown.sh*"

5.4.5 Installation verification

> Check the status of the ecp-endpoint service:

- *service ecp-endpoint status*
- The service should be running

> Display application log (ecp-endpoint.log) output:

- *service ecp-endpoint log*

> Display all available log files:

- *service ecp-endpoint log -ls*

> Display messages from a selected file:

- *service ecp-endpoint log catalina.out*
- A similar line should appear in the ecp.log file: Started Endpoint in 25.001 seconds (JVM running for 31.924)

> Display the web interface of ECP

- Open URL "http://<endpoint-ip>:8080/ECP_MODULE" in your web browser (replace the <endpoint-ip> with the correct IP address of your endpoint server)
- The page should display a registration page

5.4.6 Log files

ECP endpoint log files are available in the /var/log/ecp-endpoint folder.

5.4.7 Application accounts

RPM ECP endpoint installation automatically creates the user account *ecp-endpoint* that belongs to the *ecp-endpoint* group. This account must be used for all file-based integration like fssf integration.

To be able to login using ecp-endpoint user, it is necessary to set the password for user ecp-endpoint:

passwd ecp-endpoint

5.5 Installation on Linux (HA)

ECP endpoint can be deployed in HA mode on two or more application servers. This setup consists of the following infrastructure components:

> Two application servers with ECP endpoint.

> Shared external database.

5.5.1 Setup external database

Please see section 11. External Databases.

5.5.2 Setup application servers

Please follow section 5.4 and install prerequisites and application on both application servers in the same way as for non-HA deployment.

5.5.3 Configure application

The application must be configured according to section 5.7.1.1 for HA deployment.

5.5.4 Start and verify application

After installation, start only the first ECP endpoint and go through the registration process (see section 6. for more details). When ECP endpoint is successfully registered and synchronized, the remaining nodes may be started. Please follow section 5.4.5 to verify that the application has been installed correctly. It is recommended that each node be verified independently, one after the other. When finished, both nodes should be running.

5.6 Uninstall on Linux

5.6.1 Uninstall – RPM

To uninstall ECP endpoint, simply run the following command:

```
> rpm -e ecp-endpoint
```

5.6.2 Uninstall – installed via wizard

If the application was installed using the installer wizard, please follow one of these steps:

5.6.2.1 Standalone deployment

Stop the application and then navigate to the install directory. Change the directory to Uninstaller and execute the following command:

```
> java -jar uninstaller.jar -c
```

5.6.2.2 Deployment on existing application server

Stop the application server and undeploy war using the management console or remove it from the deployment folder.

5.7 ECP endpoint configuration

Before starting ECP endpoint, the application must be configured. The application is supplied with two configuration files.
If required, ECP Endpoint can be set up to automatically start after server start.

> chkconfig ecp-endpoint on

Config file	Description
/etc/ecp-endpoint/ecp.properties	This is the main ECP configuration file with all the properties such as database connection, paths to filesystem, etc.
/etc/default/ecp-endpoint/ecp-endpoint.defaults	Configuration of JVM options (memory settings)

5.7.1 Configuration of ecp.properties

The ecp.properties configuration file comes with many configuration parameters. All the parameters are described in the Administration Guide. Most of the configuration parameters use default values, but the following parameters must be configured for HA installation:

5.7.1.1 Parameters required for HA installation

Parameter	Description
spring.profiles.active	To enable HA mode value must be set to **ecp-ha** Please notice that **ecp-ha** profile can be also set to enable use of an external database when running component in a standalone mode. Profile **ecp-nonha** is intended for running component in standalone mode against embedded DB.
ecp.db.url	Connection URI to external database. You may find the pattern appropriate for your database in section 11. External Databases. Example: jdbc:mariadb://127.0.0.1:3306/ecp?characterEncoding=UTF-8&useSSL=false
ecp.db.username	Database user username. Example: ecp
ecp.db.password	Database user password. Example: password
spring.datasource.validation-query	This value should be set to a valid query for the selected database (see 11. External Databases) Example for the embedded DB: ‘select 1’ (without quotation marks)
ecp.cluster.manager.localInstanceID	Node identifier in the cluster. This value **must be unique**. Example: E1a or server’s hostname
internalBroker.host	Broker bind address. Use 0.0.0.0 to listen on all interfaces.
internalBroker.urls	Failover connection to internal broker. Example: failover:(amqp://<node1 IP>:5672,amqp://<node2 IP>:5672)
internalBroker.store.jdbc.url	Syntax same as ecp.db.url, used as a persistent storage for internal broker. Schema must be dedicated only for internal broker.
internalBroker.store.jdbc.user	Database user username. Example: activemq
internalBroker.store.jdbc.password	Database user password. Example: password
External broker parameters	The following parameters must be specified if direct communication can be used by this endpoint.

externalBroker.host	External broker IP address. Default is 0.0.0.0
externalBroker.store.jdbc.url	Syntax same as ecp.db.url, used as a persistence storage for external broker. Schema must be dedicated only for external broker
externalBroker.store.jdbc.user	Database user username. Example: activemqext
externalBroker.store.jdbc.password	Database user password. Example: password

6. ECP Endpoint Registration

6.1 Prerequisites

Before starting the endpoint registration please ensure that the following criteria are met:

- The component directory administrator has provided you with the component directory URL, component directory code, and registration certificate
- Your home component directory is running

6.2 Registration process

- Open a web browser and navigate to the URL of your endpoint (e.g. http://<endpoint-ip>:8080/ECP_MODULE).
- Insert the registration key store provided by your component directory administrator and wait for the file to upload. Click “Next” to proceed to the next step.
- Enter the component directory code and URL (e.g. https://<directory-ip>:8443/ECP_MODULE) provided by the component directory administrator and then click on the check connectivity. Please note that a successful connection check is required to proceed to the next step.
- Fill in your component code (allowed characters are A-Z, a-z, digits 0-9, at-sign @ and hyphen -) and your company information (please note that phone number must be in international format - e.g. 00<country code><number>) and click “Submit request”.
    - If the registration key store is provided specifically for your component, the component code is pre-filled and cannot be changed.
- At this point, you must wait until the component directory administrator approves your registration request. Once the request is approved, your endpoint automatically downloads the results and becomes operational.

7. ECP Component Directory Installation

The installation of the ECP component directory is very like the installation of ECP endpoint.

The following section describes the installation of the ECP component directory on the Linux and MS Windows platforms. Supported Linux distributions:

- CentOS 7
- RHEL 7

7.1.1 General prerequisites before installing ECP component directory

- ECP component directory installer (jar or rpm package, depends on deployment and OS type)
- Registration keystore (this keystore will be provided by the ECP network administrator)
- Component directory code (this should be defined or agreed to with the ECP network administrator)

7.1.2 Firewall configuration

Please ensure that your firewall allows communication described in 4.1.2 ECP network communication protocols

7.2 Installation on Windows

The application may be installed in either command line or GUI mode. Both modes are described in this chapter.

7.2.1 Variables

Several variables are used throughout this installation guide using the <variable_name> format.

Name	Description
<jre_install_path>	Location of the JRE 8 used by application
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name
<ecp-jar>	Path to the ECP component directory installation package, typically ecp-directory-<version>.jar

7.2.2 Prerequisites

Java Runtime Environment 8 64-bit with Java Cryptography Extension (JCE) is required.

Both Oracle's OpenJDK 8 and Oracle's Java SE 8 versions are supported. Oracle's OpenJDK Java is recommended for Organizations which do not have Oracle Java SE binaries available as a part of a commercial product or service from Oracle.

Check current Java version using the command:

java -version

Expected output:

java version "openjdk 1.8.0.181"

or

openjdk version "1.8.0_181"

- If the version does not start with "1.8" or the output is anything else, download and install Java Runtime Environment 8 for the corresponding OS from the Oracle website (for OpenJDK visit http://jdk.java.net/ or https://adoptopenjdk.net/). After installation, it is required to set JRE_HOME system variable. Open This PC, select Computer -

> Properties -> Advanced system settings -> Environment Variables -> System Variables -> New -> Name: JRE_HOME, value: path to jre (e.g. C:\Program Files\Java\jre1.8.0.181).

- (Oracle`s Java SE 8 only) If the version is 1.8.0.181 or later
  - Open the file <jre_install_path>\jre\lib\security\java.security
  - Find the line *#crypto.policy=unlimited*
  - Change the line to *crypto.policy=unlimited* and save the file
- (Oracle`s Java SE 8 only) If the version is not 1.8.0.181 or later
  - Download JCE from http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
  - Unzip the file into <jre_install_path>\jre\lib\security. All files should be overwritten.
- (Oracle`s OpenJDK 8 only) For Oracle's OpenJDK Java Cryptography Extension (JCE) is installed by default and no additional installation steps are required.

The installer verifies that JCE is available at the beginning of the installation process. If this check fails, a warning message is displayed without aborting the installation.

Please make sure that MMC (Microsoft Management Console) or Services snap-ip is not running during the installation as it prevents successful registration of the ECP service.

As the Java is upgraded to the different version after ECP Directory installation, the Windows service has to be modified to the new JVM path. To do so, please, use the command and check the Java tab:

- <install_path>/tomcat/bin/tomcat9w.exe //ES//ecp-directory

7.2.3 Command line installation

- Open a command line with administrator rights:
  - Start > type "cmd" > right click > run as administrator
- Change directory to location of <ecp-jar>, e.g.
  - cd c:\ecp-installer
- Run the ecp-directory.jar file using
  - java -jar <ecp-jar> -console
- Follow the console installation wizard
- After successful installation please configure the ECP endpoint (see 4.7 ECP endpoint configuration)

7.2.4 GUI installation

1. Double-click the <ecp-jar> file. If Java was registered to execute jar files, an installer window should open. Skip to step 5. If an installer window does not open follow steps 2-4:

2. Open command line with administrator rights:

   2.1. Start > type "cmd" > right click > run as administrator

3. Change directory to location of ecp-directory.jar, e.g.

   3.1. cd c:\ecp-installer

4. Run the ecp-directory.jar file using

   4.1. java -jar <ecp-jar>

5. ECP Component directory installer window should open.

   5.1. Select target installation path

   5.2. Next

   5.3. Choose whether the ECP component directory server should be installed as a Windows service.

   5.4. Next

5.5. Choose whether the ECP component directory application should be automatically started after server start. The preferred option is to not start ECP component directory after server start.

5.6. Next > Next

5.7. If desired, an automatic installation script can be generated. It can be used in the future to install the application using the same options.

5.8. Done

5.9. Configure the ECP directory (see section 7.6)

7.2.5 Starting and stopping the ECP component directory

7.2.5.1 Command line

a) If installed as a service:

> Start service using
  - SC start ecp-directory
> Stop service using
  - SC stop ecp-directory

b) If not installed as a service:

> Start application using
  - <install_path>\tomcat\bin\startup.bat
> Stop application using
  - <install_path>\tomcat\bin\shutdown.bat

7.2.5.2 GUI

a) If installed as a service:

> Start > Run > “services.msc”
> Find service with a name like "ecp-directory" and description "ECP Directory"
> Start / stop service by buttons on the top left

b) If not installed as a service:

> Start / stop application by using
  - <install_path>\tomcat\bin\startup.bat
  - <install_path>\tomcat\bin\shutdown.bat

7.2.6 Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain configuration files, tomcat folder, and uninstaller. After the application is successfully started, it creates additional folders for data and log files. Default names and locations of these folders are <install_path>\data and <install_path>\logs.

7.2.6.1 Service status

The status of the service, if installed, can be checked either via command line:

> SC query ecp-directory

or in the Windows Services tool (see section 7.2.5.2)

7.2.6.2 Log files

Check the application log files for more information about its status. The default location is <install_path>\logs.

7.3 Uninstallation on Windows

Stop the application and run the command line as an administrator. Navigate to <installation directory>\Uninstaller and execute the following command:

```
> java -jar uninstaller.jar -c
```

Uninstaller will remove all application components and deregister the service if present.

7.4 Installation on Linux

7.4.1 Variables

Several variables are used throughout this installation guide using the <variable_name>. format

Name	Description
<jre-rpm>	Path to the JRE 8 installation package
<jre_install_path>	Location of the JRE 8 used by application
<ecp-rpm>	Path to the ECP component directory installation package
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name

7.4.2 Prerequisites

7.4.2.1 ECP component directory installer

Download the ECP component directory rpm file and copy it to the server.

7.4.2.2 JRE

Please setup JRE according to section 5.4.2 Prerequisites(the same as for ECP endpoint).

7.4.3 ECP component directory application installation

7.4.3.1 Install using RPM package

Install the ECP component directory. Change directory to the folder where the ECP RPM is located and invoke rpm install.

```
> rpm -i <ecp-rpm>
```

> If required, ECP component directory can be set up to automatically start after server start.

```
– chkconfig ecp-directory on
```

Installer automatically creates the following folder structure:

Folder	Description

/usr/share/ecp-directory	Application binaries folder. This is the default location of <install_path>
/var/lib/ecp-directory	Application data folder
/etc/ecp-directory	ECP component directory configuration files
/etc/default/ecp-directory/	Default configuration of ECP component directory. Used to set JVM configuration and location of configuration files. (file /etc/default/ecp-directory/ecp-directory.defaults)

7.4.4 Starting and stopping the ECP component directory

After the ECP component directory is configured (please see section 7.6.1), it can be started. The ECP component directory is registered as a Linux service. The following commands are available:

> To start ECP, use command “*service ecp-directory start*”
> To stop ECP, use command “*service ecp-directory stop*”
> To display the ECP logs, use command “*service ecp-directory log*”

7.4.5 Installation verification

- Check the status of the ecp-directory service:
  - *service ecp-directory status*
  - The service should be running.
- Display the application log (ecp.log) output:
  - *service ecp-directory log*
- Display all available log files:
  - *service ecp-directory log -ls*
- Display messages from a selected file:
  - *service ecp-directory log catalina.out*
  - A similar line should appear in the ecp-endpoint.log file: Started Component Directory in 17.785 seconds (JVM running for 25.084)
- Display the web interface of ECP
  - Open URL “http://<directory-ip>:8080/ECP_MODULE” in your web browser (replace the <directory-ip> with the correct IP address of your component directory server)
  - The page should display a registration page

7.4.6 Application accounts

The ECP component directory automatically creates a user account *ecp-directory* that belongs to the *ecp-directory* group.

7.5 Installation on Linux (HA)

The ECP component directory can be deployed in HA mode on two or more application servers. This setup consists of the following infrastructure components:

- Two or more application servers with ECP component directory.
- Shared MySQL database.

7.5.1 Setup MySQL database

Please see section 11. External MySQL Database.

7.5.2 Setup application servers

Please follow section 7.4 and install prerequisites and application on both application servers.

7.5.3 Configure application

Application must be configured according to section 7.6.1.1 for HA deployment.

7.5.4 Start and verify application

After installation, start only the first ECP directory and go through the registration process (see section 8. for more details). When the ECP component directory is successfully registered and synchronized, the remaining nodes may be started. For the initial start-up of other nodes it is important to let the node start and then restart it. This procedure is required for Tomcat to use correct SSL certificates. Please follow section 7.4.5 to verify that the application has been installed correctly. It is recommended that each node be verified independently, one after the other. When finished, both nodes should be running.

7.6 ECP Component Directory configuration

Before starting the ECP Component Directory, the application must be configured. The application is supplied with two configuration files.

> If required, ECP Component Directory can be set up to automatically start after server start.

- chkconfig ecp-directory on

Config file	Description
/etc/ecp-directory/ecp-directory.properties	This is the main ECP configuration file with all the properties such as database connection etc.
/etc/default/ecp-directory/ecp-directory.defaults	Configuration of JVM options (memory settings)

7.6.1 Configuration of ecp-directory.properties

The ecp-directory.properties configuration file comes with many configuration parameters (all of them are described briefly in the ecp-directory.properties). Most of the configuration parameters use default values, but the following parameters must be configured for HA installation:

7.6.1.1 Parameters required for HA installation

Parameter	Description
spring.profiles.active	To enable HA mode value must be set to **ecp-ha** Please notice that **ecp-ha** profile can be also set to enable use of an external database when running component in standalone mode. Profile **ecp-nonha** is intended for running component in standalone mode against embedded DB.
ecp.db.url	Connection URI to external database. Example: jdbc:mariadb://127.0.0.1:3306/ecp-directory?characterEncoding=UTF-8&useSSL=false
ecp.db.username	Database user username. Example: ecp-directory
ecp.db.password	Database user password. Example: password
spring.datasource.validation-query	This value should be set to ‘select 1’ (without quotation marks)
ecp.cluster.manager.localInstanceID	Node identifier in the cluster. This value **must be unique**. Example: server’s hostname

Other contents of these files are described in the administration guide.

7.7 Uninstall on Linux

7.7.1 Uninstall – RPM

To uninstall ECP endpoint, simply run the following command:

```
> rpm -e ecp-directory
```

7.7.2 Uninstall – installed via wizard

If application was installed using installer wizard please follow one of these steps.

7.7.2.1 Standalone deployment

Stop the application and then navigate to the installation directory. Change the directory to Uninstaller and execute the following command:

```
> java -jar uninstaller.jar -c
```

7.7.2.2 Deployment on existing application server

Stop the application server and undeploy war using the management console or remove it from the deployment folder.

8. ECP Component Directory Registration

8.1 Prerequisites

Before starting the component directory registration, please ensure that following criteria are met:

> Component directory is running

8.2 Registration process

- Open a web browser and navigate to the URL of the ECP component directory (e.g. http://<directory-ip:8080/ECP_MODULE)
- Login into the application. The default credentials are admin@password and can be changed in the configuration.
- Insert the registration key store provided by the ECP network administrator and wait for the file to upload. Click "Next" to proceed to the next step.
- Fill in the component code (allowed characters are A-Z, a-z, digits 0-9, at-sign @ and hyphen -), URL and your company information (please note that phone number must be in international format - e.g. 00<country code><number>) and click "Submit request". The component code should be defined or agreed to with the ECP network administrator. The URL is https URL for your ECP component directory (e.g. https://<directory-ip>:8443/ECP_MODULE).
- After successful registration, component directory restart is required. Please follow the instructions for your operating system in the Starting and Stopping Component Directory section.

9. ECP Broker Installation

9.1 Installation on Linux

The broker application is installed using the installation package provided. Only the manual installation process is supported and described below.

9.1.1 Variables

Name	Description
<jre_install_path>	Location of the JRE 8 used by application
<install_path>	Path pointing to the target broker installation directory
<version>	Arbitrary application version name
<broker-installer-zip>	Path to the ECP broker installation package, typically broker-installer-<version>.zip

9.1.2 Prerequisites

9.1.2.1 ECP broker installation package

Download the ECP broker installation file <broker-installer-zip> and copy it to the server.

9.1.2.2 Registration keystore

The registration keystore will be provided by your ECP component directory administrator.

9.1.2.3 JRE

Please setup JRE according to section 5.4.2 Prerequisites (the same as for ECP endpoint and component directory).

9.1.2.4 Firewall

Please ensure that your firewall allows communication as described in 4.1.2 ECP network communication protocols

9.1.2.5 Application accounts

For security reasons, creating a special user account for the application is recommended. Application data and the application itself should be run under this account.

> groupadd -g 2000 ecp-broker

> useradd -u 2000 -g ecp-broker -c "Account to own and run ecp-broker" ecp-broker

9.1.2.6 ActiveMQ

Download and ActiveMQ version 5.15.8 and copy it to the server.

9.1.2.7 HW requirements

> The broker machine should be equipped with the recommended 4 GB of RAM. The amount of RAM required can be computed using the following formula: 2Gb + 0.20Gb x connected endpoint count.

9.1.3 Installation steps

1. Extract the downloaded ActiveMQ 5.15.8 package (apache-activemq-5.15.8-bin.tar.gz) to the <install_path> (e.g. /opt/ecp-broker) and rename the extracted folder to “activemq”.
2. Extract content of the <broker-installer-zip> file to the <install_path> (e.g. /opt/ecp-broker). Override any existing files inside the activemq folder.
3. Edit the <install_path>/activemq/conf/ecp-config.xml file and update the path to the broker.properties file.
   3.1. Find string “file:/opt/ecp/config/broker.properties” inside the file.

   3.2. Replace it with the path to the broker.properties file (e.g. file:/opt/ecp-broker/broker.properties)

4. Edit the <install_path>/activemq/bin/env file and insert parameter with the path to the component directory data
   4.1. Find string “ACTIVEMQ_OPTS="$ACTIVEMQ_OPTS_MEMORY” inside the file
   4.2. Add new parameter for component directory data path -Decp.directory.client.root=<*install_path*> (e.g. -Decp.directory.client.root=/opt/ecp/directory)
5. Edit the <install_path>/activemq/bin/env file and insert parameters for java heap dump creation in case of ECP broker failure.
   5.1. Find string “ACTIVEMQ_OPTS="$ACTIVEMQ_OPTS_MEMORY” inside the file
   5.2. Add new parameters for dump creation and path for storage (recommended path in example).
   -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=<path_for_dump>
   (e.g. ACTIVEMQ_OPTS="$ACTIVEMQ_OPTS_MEMORY -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath='/opt/ecp-broker/activemq/data/broker-dump.hprof').
6. Update <install_path>/broker.properties configuration settings. Follow instructions in section 9.4
7. Use broker registration tool to register the broker to the ECP component directory. Follow the instructions in section 10.
8. Start the broker – follow instructions in section 9.2.4

9.1.4 Starting and stopping ECP broker

After the ECP broker is configured and registered, it can be started. The following commands are available:

- Start the broker: “<install_path>/activemq/bin/activemq start”
- Stop the broker: “<install_path>/activemq/bin/activemq stop”
- Restart the broker: “<install_path>/activemq/bin/activemq restart”
- Show the broker status: “<install_path>/activemq/bin/activemq status”

9.1.5 Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain the configuration file and ActivmeMQ folder. After the application is successfully started, it creates an additional folder for data and log files. The default name and location of this folder is <install_path>/activemq/data.

9.1.5.1 Broker status

The status of the broker can be checked via command line:

- “<install_path>/bin/activemq status”

9.1.5.2 Log files

Check the application log files for more information about its status. The default location is <install_path>/activemq/data.

9.2 Installation on Windows

The broker application is installed using the installation package provided. Only the manual installation process is supported and described below.

9.2.1 Variables

Name	Description

<jre_install_path>	Location of the JRE 8 used by application
<install_path>	Path pointing to the target broker installation directory
<version>	Arbitrary version name
<broker-installer-zip>	Path to the ECP broker installation package, typically broker-installer-<version>.zip

9.2.2 Prerequisites

> Download and unpack ActiveMQ 5.15.8 binary distribution to <install_path> directory (e.g. C:\ecp-broker) and rename extracted folder “apache-activemq-5.15.8” to “activemq”.

> Install the Java Runtime Environment 8 64-bit with Java Cryptography Extension (JCE).

- Check current Java version using the command:

  java -version

  Expected output:

  java version "1.8.0.181"
- If the version does not start with “1.8” or the output is anything else, download and install Java Runtime Environment 8 for the corresponding OS from the Oracle website.
- If the version is 1.8.0.181 or later
  - Open the file <jre_install_path>\jre\lib\security\java.security
  - Find a line *#crypto.policy=unlimited*
  - Change the line to *crypto.policy=unlimited* and save the file
- If the version is not 1.8.0.181 or later
  - Download JCE from http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
  - Unzip the file into <jre_install_path>\jre\lib\security. All files should be overwritten.

9.2.2.1 HW requirements

> The broker machine should be equipped with the recommended 4 GB of RAM. The amount of RAM required can be computed using the following formula: 2Gb + 0.20Gb x connected endpoint count.

9.2.3 Installation steps

1. Extract contents of <broker-installer-zip> file to <install_path>.
2. Copy contents of <install_path>\activemq to <active_mq_path>
3. Edit <active_mq_path>\conf\ecp-config.xml file. Locate existing broker.properties path setting and update it to conform with <install_path>\broker.properites. (e.g. from file:/opt/ecp/config/broker.properties to file:/ECP/Broker/broker.properties)
4. Update <install_path>\broker.properties configuration settings. Follow the instructions in section 9.4
5. Use the broker registration tool to register broker with the ECP component directory. Follow the instructions in section 10.
6. Start the broker – follow the instructions in section 9.2.4

9.2.4 Starting and stopping ECP broker

Before starting the broker make sure that you have successfully performed the whole registration procedure described in chapter 10. ECP Broker Registration.

> Command line

<install_path>\activemq\bin\win64\activemq.bat start -D"ecp.directory.client.root=<install_path>"

(e.g. C:\ecp-broker\activemq\bin\win64\activemq.bat start -D"ecp.directory.client.root=c:/ecp-broker")

> To start the broker as a service:
  - Edit file <install_path>\activemq\bin\win64\wrapper.conf - add additional Java parameter defined by the following line

    wrapper.java.additional.13=-Decp.directory.client.root="<install_path>"

    (e.g. wrapper.java.additional.13=-Decp.directory.client.root="c:/ecp-broker")
  - Execute *<install_path>\activemq\bin\win64\InstallService.bat*

Note that maximum Java heap size can be also set at wrapper.conf file parameter wrapper.java.maxmemory according to recommended RAM settings – see 9.2.2.1

9.2.5 Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain the configuration file and ActivmeMQ folder. After the application is successfully started, it creates an additional folder for data and log files. The default name and location of this folder is <install_path>\ activemq\data.

9.2.5.1 Service status

The status of the service, if installed, can be checked either via command line:

> *SC query ActiveMQ*

or in the Windows services tool (see section 5.2.4.1)

9.2.5.2 Log files

Check the application log files for more information about its status. The default location is <install_path>\activemq\data.

9.3 High available installation

The ECP broker can be deployed in HA mode on two or more application servers. This setup consists of the following infrastructure components:

> Two or more application servers with ECP broker.

> Shared MySQL database.

9.3.1 Setup MySQL database

Please see section 11. External MySQL Database.

9.3.2 Setup application servers

Please follow the installation procedure for your OS. Only one server is needed for initial setup.

9.3.3 Configure application

The application must be configured as described in section 9.4 HA deployment.

In addition to broker.properties changes, also activemq.xml in your <install_path>/activemq/conf folder must be modified.

Find element persistenceAdapter in your activemq.xml

> Comment out or remove the current element and uncomment the same element below (the one with comment Persistence adapter for HA deployment). In other words, comment out kahaDB persistenceAdpater and uncomment jdbcPersistenceAdapter.

Find the element starting with <bean id="broker-ds".

> Uncomment this block. There is no need for further changes as all its values are loaded from broker.properties.

9.3.4 Start and verify application, install remaining nodes

After installation, start one node with the ECP broker and go through the registration process (see section 10. for more details). When the ECP broker is successfully registered and synchronized, the remaining nodes may be set up.

To set up the remaining nodes, stop the broker, copy the current installation directory, and distribute it to other servers. All configuration should remain untouched except the ecp.cluster.manager.localInstanceID. This value must be unique across all nodes in the cluster.

9.4 ECP broker configuration

Before starting the ECP broker, the application must be configured and registered by component directory. The application comes with a single configuration file.

> broker.properties This is the main ECP configuration file with all properties like component information, paths to filesystem, etc. File location is <install_path>/broker.properties

9.4.1 Configuration of ecp-broker.properites

The configuration file broker.properties comes with many configuration parameters (all of them are briefly described in the broker.properties file and also in this section). Some of the configuration parameters use the default value, but the following parameters must be configured for each installation:

Registration process properties

Parameter	Description
ecp.broker.contactPerson	Broker registration data
ecp.broker.contactEmail	Broker registration data
ecp.broker.contactPhone	Broker registration data
ecp.broker.organization	Broker registration data
ecp.broker.urls	Broker URL with pattern <protocol>://<host>:<port> Broker protocol is typically amqps Multiple values are separated by comma (for HA deployment). e.g. amqps://192.168.1.1:5671
ecp.broker.networks	Comma separated list of physical network names to which the URLs (ecp.broker.urls) belong. The number of networks must be equal to the number of URLs and the order must match, i.e. URL and network forms a pair. e.g. Internet
ecp.broker.code	ECP broker component code
ecp.broker.filter.components	Define which components can communicate through broker. Multiple values of component codes separated by comma. Wildcard character * can be used to allow communication for all component. Default value is *
ecp.broker.filter.types	Define which message types are accepted by the broker. Multiple values are separated by comma Wildcard character * can be used to accept all message types. Default value is *

Broker component directory properties

Parameter	Description
ecp.directory.client.synchronization.homeComponentDirectoryPrimaryCode	Component code of home component directory
ecp.directory.client.synchronization.homeComponentDirectoryPrimaryUrl	URL of home component directory. CD URL with the following pattern: <protocol>://<host>:<port> CD protocol is typically https.
ecp.directory.client.synchronization.directorySynchronizationInterval	Component directory synchronization interval CRON expression, default value is 0 */5 * * * *
ecp.directory.client.root	Customizable location of component directory data stored on broker. Location must exist at broker filesystem. e.g.: /opt/ecp_broker/cd

Broker keystores properites

Parameter	Description
ecp.keystore.location	Location of keystore with registration certificates. This keystore is provided by the component directory administrator and was listed in the prerequisites.
ecp.keystore.password	Registration keystore password (by default this is “password”)
ecp.authKeystore.location	Location of the keystore with authentication certificates (e.g. /opt/ecp-broker/authKeystore.jks). This location must point to an unexisting file as it will be used by ECP to save new keystore after successful registration.
ecp.authKeystore.password	Authentication keystore password (e.g. password).

9.4.2 Parameters required for HA installation

Parameter	Description
broker.store.jdbc.url	Connection URI to external database. Example: jdbc:mariadb://127.0.0.1:3306/ecp-broker?characterEncoding=UTF-8&useSSL=false
broker.store.jdbc.user	Database user username. Example: activemq
broker.store.jdbc.password	Database user password. Example: password
ecp.cluster.manager.localInstanceID	Node identifier in the cluster. This value **must be unique**. Example: server’s hostname

Other contents of these files are described in the administration guide.

10. ECP Broker Registration

10.1 Prerequisites

Before starting the registration of the broker please ensure that following criteria are met:

- Broker is configured (see section 9.4)
- The component directory administrator has provided you with the component directory URL, component directory code, and registration certificate
- Your home component directory is running

10.2 Registration process

The broker installation package comes with a broker registration tool (located in <install_path>). This tool is used for the broker registration and broker certificates update. The registration process is divided into three steps, all the steps must be executed successfully as specified to finish the registration process:

1. *Create a broker registration request* – creates a new broker registration request and sends it to the home component directory.
2. *Check the registration request state* – checks the registration request state at the component directory and prints it to the console, updates broker certificates when request state is approved. Note that registration request state must be approved to continue with next step.
3. *Broker configuration push* – pushes the broker configuration to the component directory.

To successfully finish the broker registration process, broker registration request must be approved by the administrator of the component directory. Please note that the outcome of the "*Check registration request state*" step can be approved, pending, or rejected. You can see the outcome in the console log output. The "*Check registration request state*" step can be executed multiple times.

The final step "*Broker configuration push*" can be executed successfully only if the outcome of the previous step "*Check of registration request state"* execution is approved (in other words the broker registration request is approved by the component directory administrator and broker certificates are updated).

To execute the commands, open the command prompt and change the current directory to <install_path> where the broker-registration-tool.jar is located.

1. Send the broker registration requests by running the command:

   java -D"spring.config.location=<install_path>/broker.properties" -jar broker-registration-tool.jar -m INITIAL_REGISTRATION

2. Check the status of the registration request by running the command:

   java -D"spring.config.location=<install_path>/broker.properties" -jar broker-registration-tool.jar -m CHECK_REGISTRATION_STATUS

   To continue with next step the output from console should be:

   "Request with ID {requestID} is approved. Storing signed AUTHENTICATION certificate."

   When console output is like: "*Request with ID {requestID} is still pending...*" or "*Request with ID {requestID} was REJECTED by component directory administrator.*" please contact your component directory administrator for broker registration process assistance.

3. Push the broker configuration to the component directory by running the command:

   java -D"spring.config.location=<install_path>/broker.properties" -jar broker-registration-tool.jar -m PUSH_CONFIGURATION

10.2.1 Broker registration tool proxy support

Broker registration tool can be used also for ECP networks accessible through forward proxy server. To enable proxy start tool with additional JVM command line parameters:

- socksProxyHost – define proxy IP address or host name
- socksProxyPort – define proxy port

For example, the broker registration request can be sent through proxy server by running the following command:

```
java -D"spring.config.location=<install_path>/broker.properties" -D"socksProxyHost=<HostIP>" -D"socksProxyPort=20080" -jar broker-registration-tool.jar -m INITIAL_REGISTRATION
```

11. External Databases

To install ECP on an external database, follow a sub-chapter appropriate for your database system. The chapters list some ECP parameters related to the database that are specific for a given database system. For the complete list of parameters that must be configured, refer to the HA chapter for your component type:

- Endpoint: 5.7.1.1 Parameters required for HA installation
- Component directory: 7.6.1.1 Parameters required for HA installation
- Broker: 9.4.2 Parameters required for HA installation

Remember to configure your ECP instance for HA mode (profile) when using the external database even if you have the ECP installed on just one server.

11.1 MySQL database

11.1.1 Install the database

The supported version of MySQL database is MySQL 5.7.15. Please follow the official installation guides to install MySQL 5.7 for your environment.

Please be aware that there is a change in version 5.7 regarding root access to the server in a Linux environment. MySQL 5.7 creates a password automatically and writes it into the error log which is by default located at: n1 /var/log/mysql/error.log

Example command to see temporary password:

```
tail -n1 /var/log/mysql/error.log
```

Example output:

```
2016-09-20T11:32:17.262112Z 1 [Note] A temporary password is generated for root@localhost: t;/VkleMJ9eM
```

11.1.2 Configure the database

ECP requires the following configuration in the mysql configuration file (usually located in /etc/my.cnf). It is recommended, but not mandatory, to change properties like innodb-buffer-pool-size and innodb-log-file-size according to the environment.

Properties required by ECP:

Parameter	Value
lower-case-table-names	1
max_allowed_packet	100M

11.1.3 Create application schema and configure user access

It is necessary to be able to connect from the ECP application servers under the application account to the MySQL database. The following command can be used to setup MySQL accounts. Please change the password included in this example:

```
mysql -u root -ppassword <<EOF
CREATE USER IF NOT EXISTS 'ecp'@'%' IDENTIFIED BY 'ecp-password';
CREATE DATABASE IF NOT EXISTS ecp
  DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON ecp.* TO 'ecp'@'%' IDENTIFIED BY 'ecp-password' with GRANT OPTION;
```

GRANT ALL ON ecp. TO 'ecp'@'localhost' IDENTIFIED BY 'ecp-password' with GRANT OPTION;*

EOF

Note: This command creates only an ECP user (used by application). When deploying the application in HA setup, user accounts and databases for brokers should be created as well. List of recommended database names for different ECP component types:

- **Component Directory**
  - ecp
- **Endpoint**
  - ecp
  - int-broker – ActiveMQ database
  - ext-broker – ActiveMQ optional database should be created only when DIRECT communication pattern is enabled
- **Broker**
  - broker - ActiveMQ database

11.1.4 ECP configuration changes

The following table shows MySQL-specific values for parameters that must be changed in the ECP configuration file.

Parameter	Value
spring.datasource.validation-query	select 1
ecp.db.url	jdbc:mariadb://<host>:<port>/<schema>?autoReconnect=true&characterEncoding=UTF-8
spring.datasource.driver-class-name	org.mariadb.jdbc.Driver

11.2 MS SQL database

11.2.1 Install the database

The supported version of MS SQL Database is SQL Server 2016. Other versions of the database (SQL server version 2012 and above) may work correctly as well, but the support is not guaranteed and ECP is not tested against versions other than the supported one. Please follow the official installation guides to install SQL Server for your environment. Note that SQL Server Authentication mode must be set to mixed - SQL Server Authentication is required by ECP.

11.2.2 Create application databases

Microsoft SQL Server Management Studio is recommended for creation of ecp databases and users with logins.

The following T-SQL script snippets can be also used for ecp database creation.

When HA setup for Endpoint or Broker is enabled, ActiveMQ databases must be also created. Please repeat the steps below to create one or more ActiveMQ databases.

List of recommended database names for different ECP component types:

- **Component Directory**
  - ecp
- **Endpoint**

  - ecp
  - int-broker – ActiveMQ database
  - ext-broker – ActiveMQ optional database should be created only when DIRECT communication pattern is enabled

- **Broker**
  - broker - ActiveMQ database

Please note that ActiveMQ DDL script for creating the ActiveMQ database(s) must be executed manually. Auto creation does not work for MS SQL server due to an ActiveMQ bug. DDL script must be executed successfully after the creation of Active MQ database and before first start of ECP Endpoint or Broker. The ActiveMQ DDL script is in ECP component installation folder (/scripts/broker-ddl-mssql.sql)

11.2.2.1 Create database

Please check and modify database name and filenames before execution of the script.

```
USE [master]
GO
CREATE DATABASE [ecp]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'ecp', FILENAME = N'<db_path>\ecp.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'ecp_log', FILENAME = N'<db_path>\ecp_log.ldf' , SIZE = 73728KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
GO
```

11.2.2.2 Create login

Please check and modify login name, password and default database name before execution of the script.

```
USE [master]
GO
CREATE LOGIN [ecp] WITH PASSWORD=N'ecp-password', DEFAULT_DATABASE=[ecp], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
```

11.2.2.3 Create database user

Please check and modify user and database name before execution of the script.

```
USE [ecp]
GO
CREATE USER [ecp] FOR LOGIN [ecp] WITH DEFAULT_SCHEMA=[dbo]
GO
```

11.2.2.4 Assign user to the following roles:

Please check and modify user and database name before execution of the script.

```
 USE [ecp]
```

```
GO
ALTER ROLE [db_datareader] ADD MEMBER [ecp]
GO
USE [ecp]
GO
ALTER ROLE [db_datawriter] ADD MEMBER [ecp]
GO
USE [ecp]
GO
ALTER ROLE [db_ddladmin] ADD MEMBER [ecp]
GO
USE [ecp]
GO
ALTER ROLE [db_owner] ADD MEMBER [ecp]
GO
```

11.2.3 ECP configuration changes

The following table shows MSSQL-specific values for parameters that must be changed in the ECP configuration file.

Parameter	Value
spring.datasource.validation-query	select 1
ecp.db.url	jdbc:sqlserver://<host>:<port>;databaseName=<database>
spring.datasource.driver-class-name	com.microsoft.sqlserver.jdbc.SQLServerDriver

11.3 Oracle database

11.3.1 Install the database

The supported and tested version of Oracle Database is 12c. Other database versions may also work, but the support is not guaranteed and ECP is not tested against versions other than the supported one. Please follow the official installation guides to install Oracle Database for your environment.

The charset of the database instance must be set to AL32UTF8, otherwise the installation of database does not work.

11.3.2 Create application databases

Oracle SQLDeveloper tool is recommended for creation of ECP databases and users with logins.

The following PL/SQL script snippets can be also used for ECP user and database creation.

When HA setup for Endpoint or Broker is enabled, ActiveMQ databases must be also created. Please repeat the steps below to create one or more ActiveMQ databases.

List of recommended database names for different ECP component types:

- **Component Directory**
  - ecp

> **Endpoint**
  - ecp
  - int-broker – ActiveMQ database
  - ext-broker – ActiveMQ optional database should be created only when DIRECT communication pattern is enabled

> **Broker**
  - broker - ActiveMQ database

11.3.2.1 Create application user and database

CREATE USER ecp IDENTIFIED BY password ;

11.3.2.2 Assign user to the following roles and privileges

GRANT "CONNECT" TO ecp ;

GRANT ALTER ANY INDEX TO ecp ;

GRANT CREATE ANY INDEX TO ecp ;

GRANT ALTER ANY TABLE TO ecp ;

GRANT ALTER ANY SEQUENCE TO ecp ;

GRANT CREATE TABLE TO ecp ;

GRANT SELECT ANY SEQUENCE TO ecp ;

GRANT MANAGE TABLESPACE TO ecp ;

GRANT CREATE TABLESPACE TO ecp ;

GRANT CREATE SEQUENCE TO ecp ;

GRANT UPDATE ANY TABLE TO ecp ;

GRANT ALTER TABLESPACE TO ecp ;

GRANT READ ANY TABLE TO ecp ;

GRANT INSERT ANY TABLE TO ecp ;

GRANT ALTER ANY TYPE TO ecp ;

GRANT CREATE ANY TABLE TO ecp ;

GRANT ALTER DATABASE TO ecp ;

GRANT DROP TABLESPACE TO ecp ;

The following command sets unlimited tablespace. If this is not desired behaviour, please contact your DBA to set up quotas for you.

GRANT UNLIMITED TABLESPACE TO ecp ;

11.3.3 ECP configuration changes

The following table shows Oracle-specific values for parameters that must be changed in the ECP configuration file.

Parameter	Value

spring.datasource.validation-query	select 1 from dual
ecp.db.url	jdbc:oracle:thin:@<host>:<port>:<sid>
spring.datasource.driver-class-name	oracle.jdbc.OracleDriver

11.4 PostgreSQL database

11.4.1 Install the database

The supported and tested version of PostrgeSQL Database is 9.6. Other database versions may also work, but the support is not guaranteed and ECP is not tested against versions other than the supported one. Please follow the official installation guides to install PostgreSQL Database for your environment.

11.4.2 Create application databases

PostgreSQL pgAdmin is the recommended tool for creation of ECP databases and users with logins.

The following SQL script snippets can also be used for ECP user and database creation.

When HA setup for Endpoint or Broker is enabled, ActiveMQ databases must be also created. Please repeat the steps below to create one or more ActiveMQ databases.

List of recommended database names for different ECP component types:

- **Component Directory**
  - ecp
- **Endpoint**
  - ecp
  - int-broker – ActiveMQ database
  - ext-broker – ActiveMQ optional database should be created only when DIRECT communication pattern is enabled
- **Broker**
  - broker - ActiveMQ database

11.4.2.1 Create application user and database

```
CREATE ROLE ecp LOGIN
  ENCRYPTED PASSWORD 'md54f9fc70b2169fc81a1eb20d71ac314b2' -- password
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATION;


CREATE DATABASE ecp
  WITH OWNER = ecp
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       CONNECTION LIMIT = -1;
```

11.4.3 ECP configuration changes

The following table shows PostgreSQL-specific values for parameters that must be changed in the ECP configuration file.

Parameter	Value
spring.jpa.properties.hibernate.dialect	eu.entsoe.ecp.common.db.postgres.PostgreSQLDialectCustom
ecp.db.url	jdbc:postgresql://<host>:<port>/<database>
spring.datasource.driver-class-name	org.postgresql.Driver

12. Docker Support

ECP components can be run within Docker containers – ECP components are shipped also as Docker images. We also provide also Docker Compose file for setup of basic test ECP network.

Using ECP in Docker assumes basic Docker skills. To get familiar with Docker please visit following links:

> https://docs.docker.com/engine/docker-overview/
> https://docs.docker.com/

12.1 Prerequisites

> Docker for Windows / Linux installed. Please see https://docs.docker.com/engine/installation/ for instructions.
> Access to ECP repositories on Docker Hub (https://hub.docker.com/) – can be granted on request to your existing Docker Hub account

12.2 ECP Docker images

ECP4 is delivered with Docker Images. The ECP Docker Images are available for all ECP4 components (component directory, endpoint and broker). Images are available at Docker Hub (https://hub.docker.com/) under the following names:

> entsoe/ecp-directory
> entsoe/ecp-endpoint
> entsoe/ecp-broker

Images are tagged with their version (e.g. 4.1.2.542) and you can always get the latest revision of selected minor version by pulling tag containing only major and minor versions (e.g. 4.1).

Before you start your container with ECP software, check the following sections and consider the use of port and volume mapping. It is recommended to store your data outside the Docker container either by mounting a data directory on your Docker host to the volume inside the container (see the **Chyba! Nenalezen zdroj odkazů.** section) or by using a n external database.

12.2.1 Exposed ports

> Endpoint
- 8080 – provides access to web GUI
- 5671 – optional AMQPS connector used for direct communication between endpoints
- 5672 – AMQP connector used for AMQP API for business applications

> Component directory
- 8080 – provides access to web GUI
- 8443 – provides access to the component directory API (e.g. for synchronization)

> Broker
- 8161 – provides access to web management console of ActiveMQ
- 5671 – amqps transport protocol of ActiveMQ

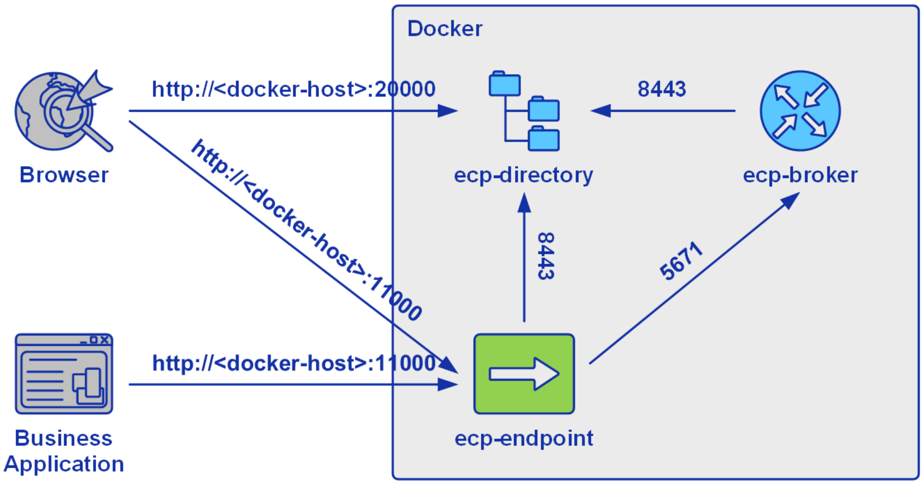
12.2.2 Environment variables

> Endpoint and component directory
- JAVA_OPTS – this variable can be used to modify JVM memory allocation for endpoint or directory. Also, additional JVM parameters can be specified.

> Broker
- ACTIVEMQ_OPTS_MEMORY – this variable should be used to modify JVM memory allocation of broker

12.3 ECP Docker Compose test environment

Together with ECP Docker Images, the Docker Compose test environment is also provided.



The test environment can be easily run with one CLI command and contains all available ECP component types (component directory, endpoint, broker), one container instance per component type.

12.3.1 How to start / stop Docker Compose environment

Open command line and change to folder where test environment file docker-compose.yml is located. Next execute command:

> docker-compose up

Wait until all ECP containers are running. For testing purposes, all ECP components are already registered under the started component directory, therefore manual component registration is not required. The endpoint has also a default message path configured so you can immediately start sending messages.

To check if all components are running execute Docker CLI command:

> docker ps

You can also open browser and visit URLs:

- http://127.0.0.1:20000/ECP_MODULE for access to component directory (login admin@password)
- http://127.0.0.1:11000/ECP_MODULE for access to endpoint
- http://127.0.0.1:12010 for access to ActiveMQ management console (login admin@admin)

If you are not running the Docker on your machine, substitute the IP address 127.0.0.1 for the address of the Docker host machine.

Once the environment is started, you can use it for your testing purposes, e.g. to evaluate the platform or to test the integration with your business application. By default, the endpoint has only Web Service interface enabled, but you can configure the endpoint however you need using the standard procedure (see the Administration Guide).

To stop ECP Docker test environment, either press “Ctrl+C” keyboard shortcut within the terminal running the “docker-compose up” command in foreground, or issue command “docker-compose stop” under the folder containing the docker-compose.yml file.

12.3.2 ECP Docker Compose environment configuration

12.3.2.1 Logs

Endpoint and component directory components log files are located in the following folders:

- directory/logs
- endpoint/logs

12.3.2.2 ECP component configuration

Component configuration can be changed manually by editing its property file. The property file is located at nested config folder under the component root folder:

- directory/config/ecp-directory.properties
- endpoint/config/ecp.properties
- broker/config/broker.properties

For instructions, how to configure specific component please see the Administration Guide.

12.3.2.3 ECP containers configuration

Delivered testing Docker compose configuration (docker-compose.yml) can be edited manually to experiment with the ECP setup. You can change the default port mapping for containers, specify additional JVM parameters for ECP components (e.g. limit the memory settings) or you can also add an additional endpoint to the network.

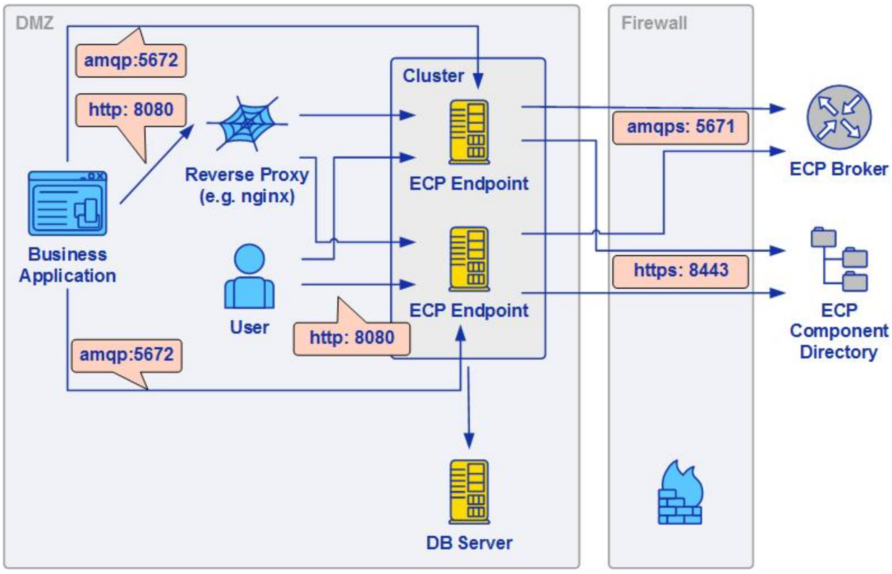
Please see the Docker Compose documentation online (**https://docs.docker.com/compose/**) for more information on the configuration file options.

13. Component Deployment - Security Recommendations

This section provides information about secure deployment of ECP components. The following recommendations represent the recommended level of security, which can be further enhanced (e.g. by additional components like proxies if necessary).

13.1 Endpoint deployment

ECP endpoint secure deployment is formed of high availability (HA) deployment and managed access rules. Access to endpoint cluster is not restricted inside demilitarized zone (DMZ). Outside DMZ inbound access to endpoint is completely denied by the firewall rules. Outbound communication from endpoint to broker or component directory is allowed at firewall for AMQPS and HTTPS protocol ports e.g.: 5671, 8443. HTTP access to the endpoint cluster nodes can be routed through a reverse proxy inside DMZ to provide load balancing and transparent failover. This applies for business applications using web service API and for the users accessing ECP via GUI.

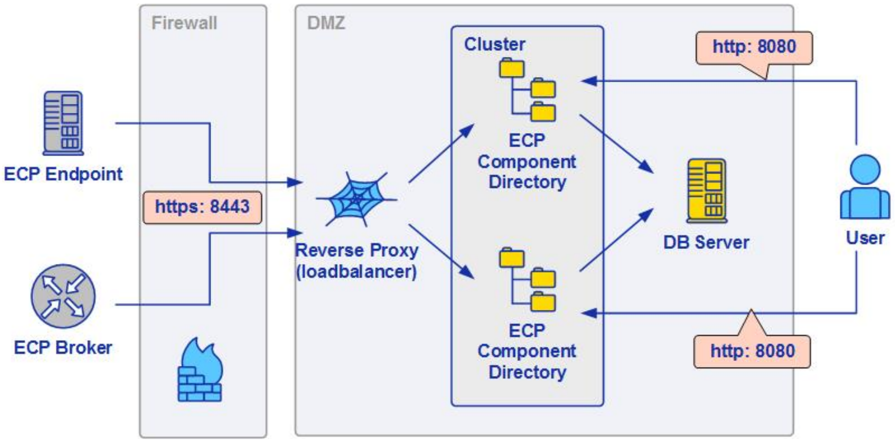


13.2 Component directory deployment

Component directory not only opens outbound connections, but it must also be able to accept inbound connections from other ECP components. Inbound connections are allowed only for HTTPS protocol (default ECP port setting is 8443). HTTPS protocol is used by component directory synchronization. Just like endpoint, component directory is recommended to be deployed at HA setup – access to cluster nodes is load balanced (reverse proxy).

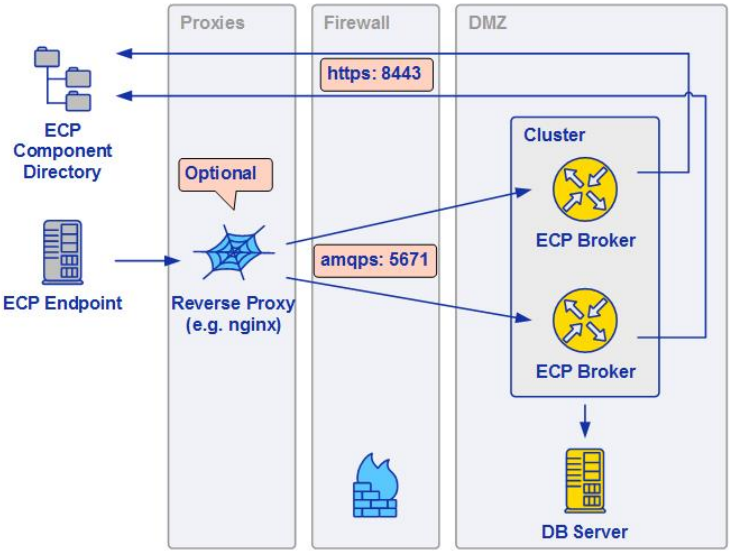
HTTP Reverse Proxy cannot be used due to Authentication Certificate that needs to be validated on the Component Directory. HTTP Reverse Proxy normally terminates the SSL connection and removes the client Authentication Certificate.

The deployment was tested against NGINX proxy.



13.3 Broker deployment

Outbound connections from the broker are allowed only for component directory synchronization on HTTPS protocol (port 8443). Inbound connections from endpoints are allowed for AMQPS protocol (default ECP port setting is 5671). Just like endpoint and component directory, broker is recommended to be deployed at HA setup. Access to the broker cluster nodes can be provided through TCP load balancer / reverse proxy or it can be addressed directly from endpoint to the specific broker node through configured multiple failover URLs.



14. FAQ

This section contains the most frequently asked questions and answers to these questions in a single place.

14.1 Support of a forward proxy

Question: Is it possible to use http proxy for ECP?

Answer: ECP uses the AMQP protocol to exchange data and it is not possible to use http proxy. It is possible to use SOCKS5 proxy, which works on a lower session layer of OSI network model and is agnostic to the application protocol.

ECP was tested with Dante proxy server (https://www.inet.no/dante/). See the Administration Guide for details on the configuration of proxy in ECP.

14.2 Support of a reverse proxy

Question: Is it possible to use a reverse proxy for inbound connections?

Answer: Yes, it is possible with some constraints put by the transport protocols, specifically AMQP. The reverse proxy directing traffic to a Broker or Endpoint must support a TCP forwarding to handle AMQP connections. The Component Directory uses only HTTP communication protocol so it is possible to use any HTTP proxy or the TCP forwarding as well.

ECP was tested with NGINX (https://www.nginx.com) to intercept the inbound connections using its TCP and UDP load balancing features (https://www.nginx.com/resources/admin-guide/tcp-load-balancing/). Below is a basic sample configuration to proxy connections to an ECP Broker.

```
stream {
   server {
      listen     5671;
      proxy_pass stream_brokers;
   }
   upstream stream_brokers {
      server 172.22.0.11:5671;
   }
}
```

14.3 Support of other databases for ECP endpoint

Question: Is it possible to use other than the supported databases?

Answer: For non-HA mode, it is possible to use an embedded database, which does not have to be installed. For HA mode, it is only possible to use one of the supported external databases listed in chapter 11. External Databases.