Statnett

Visiting adress
Nydalen allé 33, 0484 Oslo

Postal adress
PB 4904 Nydalen, 0423 Oslo

Enterprise No.
NO 962 986 633 MVA

T +47 23 90 30 00
F +47 23 90 30 01

W statnett.no
E firmapost@statnett.no

99.9% Uptime – 100K Messages per day

How to build and maintain such an endpoint

Version: 1.00

Date: 27.05.2021

1 Document History

Version	Date	Changes
1.00	16/04-2021	- The first draft

2 Introduction

An ECP/EDX-endpoint will for some require a 99.9% uptime in order to comply with the needs of the business. In this document we outline the Statnett strategy to achieve that and supply with resources on how to handle problems that eventually will occur. The document will also help the endpoint to handle a higher load (100K messages per day).

3 Build

3.1 Manual failover setup

We recommend building a manual failover to achieve the goal. Later in the document you'll find reasoning on why we do not choose automatic failover or high-availability setup. Here is what you need to do to setup a manual failover:

1) Complete default installation of ECP/EDX-endpoint, with default internal database.
2) After endpoint is running, install another endpoint on another host, but do not register the endpoint.
3) Setup a job to copy data-files from the original endpoint to the failover endpoint. Locate these 3 folders for both ECP and EDX, and copy them:
    a. Database (usually named db)
    b. Broker (usually named internalbroker or edx-activemq-data)
    c. Configuration (usually named conf, ecp-endpoint or edx-toolbox)
4) Other files (like JKS-files) will be created when starting the endpoint.
5) Make sure your Business Applications can retrieve send messages from both endpoints, whichever is active.
6) Only run one endpoint at the time, otherwise one endpoint might "steal" acknowledgements/messages from the other or in the worst case the endpoint will stop retrieving messages from the central broker.
7) Test the arrangement

There is a downside to this arrangement apart from the obvious one; that it is manually controlled: When you stop the failover-endpoint you might be unlucky and capture a message in-flight. Such a message will never be delivered unless you start the failover-endpoint again. For this reason, you would not want to use the failover option very often, but it could be useful whenever you have a major upgrade, major mal configuration of network or hardware failure.

3.2 Why we don't recommend automatic failover

1) We don't believe we have a fool proof way of knowing that only one endpoint will run at the time. We cannot easily detect if an endpoint is down, slow, halfway-connected (can send, but not receive – or vice versa) or has other connection problems. Thus, an automatic failover could cause two endpoints to run simultaneously – which cause problem the problems mentioned in previous chapter.
2) Automatic triggering of the failover could result in many such failovers because of some minor and frequent instability. If the instability is short lived (a few minutes), it could be better to simply wait it out. This depends upon the requirement of the traffic; the underlying assumption is that this network is not used for anything resembling real-time traffic but operates within minutes-boundaries.

3.3 Why we don’t recommend High Availability (HA) setup

ECP and EDX offers HA setup, so why do we not recommend that setup? In short, the answer is that we believe it's too costly for most and we're not convinced it will increase uptime. To be blunt, we think that the complexity of HA-setup will cause some downtime in itself and it will negate the gain you get from more hardware resources. With a high investment of time/resources/knowledge into such a setup, tuned over some time, it is probable that it would be a better solution.

Our reasoning goes as follows:

1) The benefit of HA is usually to cover breakdown in hardware. Network problems could sometimes be alleviated, but application problems cannot be expected to be helped by HA, quite the opposite (expect more complex database-setup).
2) Modern day hosting has built-in HA, so that failure in hardware should not affect the application in most cases. We expect less than 1 such incident per year. Network problems due to mal configuration (again, network hardware is expected to be redundant) is more likely, but a major incident where HA would help is expected to be rare.
3) Thus, the expected number of incidents where HA would be helpful is set to one per year as a maximum estimate. Such an incident can be handled with manual failover within 3 hours in a worst-case scenario, but in many cases much quicker. Measured across a year, this is well within 99.9% uptime.
4) Introduction of HA also introduces a more complex setup of external database (MySQL, Oracle, MSSQL) and more importantly, a less well-tested setup. It is very reasonable, and experience shows, that such setup by itself introduces some amount of downtime. Only by investing enough time/resources/knowledge into such a setup will you be able to counter the added complexity. ECP/EDX has a small user base with a wide variety of setup options – making it less likely that it will be stable in all cases.

3.4 Handling more traffic

For the time being (Spring 2021, ECP v4.7.2 and below + EDX v1.8.2 and below), these are the settings we use on ECP/EDX-endpoint – to make our system able to handle with high load (100K messages pr day):

3.4.1 ecp.properties

Property	**Explanation**
ecp.messagebox.retentionPeriod=86400000	Make sure you keep only one day (86400s) of messages in your message log (you can see the log in ECP GUI). If the number of messages go above 100K on ECP 4.7.2 and below, it will cause GUI to become rather unusable.

3.4.2 edx.properties

Property	**Explanation**
edx.toolbox.deleting.deleteOlderThan=24	Make sure you keep only one day (24h) of messages in your message log (you can see the log in EDX GUI). If the number of messages go above 100K on EDX 1.8.2 and below, it will cause the GUI to be rather unusable.
edx.toolbox.deleting.deleteJobDelay=60000	Start the job do delete messages a minute after it completed its last run.
edx.toolbox.deleting.dms.deleteOlderThan=1	Delete messages older than 1 hours in DMS. DMS contains a copy of the messages (unlike the message log mentioned above, which is simply a log entry in the database). If you receive an acknowledgement more than 1 hour after the message was sent, it will result in error message in the log.

4 Maintain

4.1 Problems and how to fix them

Problem	**Options – each option should be a complete sequence to try to solve the issue. The options are ordered from harmless to nuclear.**
ECP Configuration Reload – process is stuck	1. Restart ECP-endpoint. Test message flow.
ECP Certificate renewal fails ECP Cannot connect to CD/Broker due to certificate problems	2. Restart ECP-endpoint. Test message flow. 3. Go to ECP-GUI-Settings. Push configuration. Restart ECP-endpoint. Wait 1-5 minutes and test message flow. 4. Control that your firewall is not interfering (deep packet inspection) with the SSL-traffic between ECP-endpoint and/or CD/Broker. Check by running something similar to this command (openssl s_client -connect ecp4.statnett.no:5671 -showcerts). The firewall must not touch/change anything in SSL-traffic. 5. Go to ECP-GUI-Settings. Renew Manually. Wait 1-5 minutes and test message flow. 6. Re-register endpoint: Check if you have a not-expired registration keystore. If not, contact Statnett (ecp@statnett.no) to retrieve one. The registration request will require Statnett do a manual operation – so this step can take some time. a. In ECP 4.7.x and above: Go to ECP-GUI-Settings. Initiate registration. Complete sequence as described in installation guide. b. In ECP 4.6.x and below: Delete db-folder of ECP. Restart ECP-endpoint. Complete sequence as described in installation guide.
ECP connects to a wrong broker ECP tries to send messages to the wrong broker	1. Restart ECP-endpoint 2. Manually remove some rows from some tables – the following show how to do this in Derby-database, but the same SQL applies for any database of course. a. Connect to the database (see chapter "How to connect to a Derby-database) b. Run the following SQLs: i. select * from ecp.message_path_sender; ii. select * from ecp.message_upload_route; iii. select * from ecp.message_path; c. find the set of IDs that are connected to the wrong broker (usually with type MESSAGE_PATH_TYPE of "ACKNOWLEDGMENT"). Assume you found ID 101 and 102 as common ID in all tables. If there are some variations, adjust the SQL to delete those IDs. Make sure not to delete any ID that is connected to your "local" broker. Now run the appropriate SQLs: i. delete from ecp.message_path_sender where message_path in (101,102); ii. delete from ecp.message_upload_route where id in (101,102); iii. delete from ecp.message_path where id in (101,102); 3. Re-register by following 5b above

ECP or EDX stuck/frozen; messages are not received/retrieved	It can often be hard to know where the problem is located. Even if EDX is stuck, it could be that the problem-queue is on ECP. Therefore, make sure to investigate both servers. The advice below will be to delete/purge message – it is expected that proper B2B communication have business acknowledgements, to that even loss of messages will be handled. 1. Restart ECP/EDX. It is best if EDX is started 30 sec before ECP, if they're both taken down. 2. Purge a specific queue which is assumed to be full: a. Install Hawtio (see chapter "How to install Hawtio"). Wait 30 sec. b. Connect to your ECP or EDX GUI, but change the URL path to "/hawtio/". c. The top-menu should show "ActiveMQ" – click on that choice. d. All queues will be listed, find queues where queue size > 0 e. Consider purging the queue (click on queue – choose "Delete"-submenu – Purge) or delete specific messages. 3. Purge all messages from ECP/EDX-endpoint: a. For EDX: Delete the edx-activemq-data-folder and restart. b. For ECP: Delete the internalbroker-folder and restart. 4. Reset endpoint: a. For EDX: Delete both db and edx-activemq-data folders and restart EDX. Be aware that EDX will not work properly until it has received a new ServiceCatalogue-copy. This happens automatically after restart, but it requires that a message is transmitted to EDX from ECP (you can see this message in ECP inbox, but not in EDX messages). If you already have other messages coming in from ECP to EDX, the SC-copy will not be processed before the other messages are processed. BUT: The EDX may not be able to process any messages without the SC-copy and will get stuck! To avoid this, you must purge messages from ECP as well. In a high-traffic-volume endpoint this may be difficult. b. For ECP: Follow option 5b in the "ECP Certificate renewal fails"-problem.

4.1.1 How to connect to a Derby-database:

1. Download Derby client: https://db.apache.org/derby/derby_downloads.html (we have been running 10.14.2)
2. 2. Unzip into a folder, ex /opt/derby
3. Change folder to your ECP, ex /var/lib/ecp-endpoint (your database you should now be found on "db"-folder in the folder you are located)
4. Stop ECP-endpoint - you cannot edit the DB from more than one user at the time (or make a copy of the DB-folder and work on that)
5. 5. Run /opt/derby/db-derby-10.14.2.0-bin/bin/ij
    a. ij version 10.14
    b. ij> connect 'jdbc:derby:db';

   c. The "db" marked in red above is the name of the folder, so this works if you want to copy the database to another folder and work on it while the ECP endpoint is running. While inside the ij-tool you can do all SQL and some other commands:
   d. ij> help;
   e. ij> show tables;
   f. ij> exit;

6. After exit and you've edited the database (by INSERT/UPDATE/DELETE) you should check that file permission/ownership is the same as before – and if not, change back so that the ECP/EDX process can read/change the database.

4.1.2 How to install Hawtio

1. Add/change the property "spring.jmx.enabled=true" to ecp.properties and edx.properties. This will make it possible for Hawtio to see the queues. Restart ECP or EDX if change was necessary.
2. Download hawtio: https://repo1.maven.org/maven2/io/hawt/hawtio-web/1.5.11/hawtio-web-1.5.11.war
3. Rename the file to "hawtio.war" and place the file in the webapps-folder of ECP and/or EDX – it will automatically install.
4. After you've done using Hawtio you should remove the war-file – Hawtio is a liability security-wise. You should also consider reversing the jmx-settings introduced in 1.