Syscheck

Mysql replication for

redundancy and fail over

Henrik Andreasson

kinneh@users.sourceforge.net

2010-11-23

Version 1.5.17

Table of Contents

1 System preparation for database replication

1.1 Prerequisites

Master node must be installed and running.

Slave node must have syscheck installed, it will be configured during this instruction.

1.2 Configure master- and slave-hosts ipaddresses

Set THIS_NODE to NODE1(master) or NODE2(slave) and IP-addresses to node1 and node2
on node1 AND node2 run:

```
smartcard20-node1:/usr/local/syscheck # vi config/common.conf (was resources.sh )

#IP address or hostname to primary and secondary cluster nodes.

THIS_NODE=NODE1

# master node

HOSTNAME_NODE1=192.168.31.140

# slave node

HOSTNAME_NODE2=192.168.31.142
```

1.3 Configure SSH-keys

Follow the instruction in syscheck-setup-and-upgrade.pdf chapter “SSH Keys installation”

1.4 Configure related-enabled scripts

Enable these scripts to make it possible to run the copy-config command (921)

```
user@smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled> sudo ln -s ../related-available/906_ssh-copy-to-remote-machine.sh

user@smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled> sudo ln -s ../related-available/915_remote_command_via_ssh.sh .

user@smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled> sudo ln -s ../related-available/921_copy_htmf_conf.sh
```

1.5 Copy config from node1 to node2

Run the backup of key config and keystore files

Note: these files will be used in later steps during setup or upgrade

Optionally add or remove files to be copied:

```
username@smartcard20-node1:/usr/local/syscheck/> sudo vi config/921.conf
```

on node1 run:

```
root@smartcard20-node1:/usr/local/syscheck/related-enabled # ./921_copy_htmf_conf.sh  -s

Screenonly output:

Copying file: /usr/local/certificate-services/htmf/hardtokenmgmt.properties to:localhost dir:/tmp/backup_htmf_conf/
remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/htmf/autogenerated_hardtokenmgmt.properties to:localhost
dir:/tmp/backup_htmf_conf/ remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/htmf/src/resources/globalsettings/global.properties to:localhost
dir:/tmp/backup_htmf_conf/ remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/htmf/jarsigner.jks to:localhost dir:/tmp/backup_htmf_conf/ remotreuser:jboss sshkey:
/home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/ejbca/conf/ejbca.properties to:localhost dir:/tmp/backup_htmf_conf/ remotreuser:jboss
sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/ejbca/conf/database.properties to:localhost dir:/tmp/backup_htmf_conf/
remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/jboss/server/default/conf/keystore/keystore.jks to:localhost dir:/tmp/backup_htmf_conf/
remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/jboss/server/default/conf/keystore/truststore.jks to:localhost
dir:/tmp/backup_htmf_conf/ remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa

Screenonly output:

Copying file: /usr/local/certificate-services/jboss/server/default/deploy/jboss-web.deployer/server.xml to:localhost
dir:/tmp/backup_htmf_conf/ remotreuser:jboss sshkey: /home/jboss/.ssh/id_rsa
```

1.6 Configure database users and ipaddresses for nodes

Verify that the database name, username, and password to access the database is correct
on node1 AND node2 run:

```
smartcard20-node1:/usr/local/syscheck # vi config/common.conf (was resources.sh )

DB_NAME=ejbca

DB_USER=ejbca

DB_PASSWORD="foo123"
```

Enter new information about replication user into syscheck config/common.sh (you need to make up the username and password, the scripts later on will create the user based on that information)

```
# Database replication user and password

DBREP_USER=ejbcarep

DBREP_PASSWORD="foo123"
```

1.7 Create mysql-user access rules on node1

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./802-create-mysql-ejbca-user-db-user.sh

Will now insert these sql:

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.10' IDENTIFIED BY 'foo123';

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.11' IDENTIFIED BY 'foo123';

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.12' IDENTIFIED BY 'foo123';

select * from user where user like '%ejbca%'

Host User Password Select_priv Insert_priv Update_priv Delete_priv Create_priv Drop_priv Reload_priv Shutdown_priv
Process_priv File_priv Grant_priv References_priv Index_priv Alter_priv Show_db_priv Super_priv Create_tmp_table_priv
Lock_tables_priv Execute_priv Repl_slave_priv Repl_client_priv Create_view_priv Show_view_priv Create_routine_priv
Alter_routine_priv Create_user_priv ssl_type ssl_cipher x509_issuer x509_subject max_questions max_updates max_connections
max_user_connections

10.1.1.10 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

10.1.1.11 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

localhost ejbca *77805DB3940553564EF23E5EB2231A1BEB02EFC3 N N N N N N N N N N N N N N N N N N N N N N N N
```

```
N N 0 0 0 0

10.1.1.12 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

I-8021-PKI 20090311 15:57:53 smartcard20-node1: INFO - access rules inserted into mysql db ok
```

1.8 Create mysql-user access rules on node2

on node2 run:

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./802-create-mysql-ejbca-user-db-user.sh

Will now insert these sql:

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.10' IDENTIFIED BY 'foo123';

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.11' IDENTIFIED BY 'foo123';

GRANT ALL ON ejbca.* to 'ejbca'@'10.1.1.12' IDENTIFIED BY 'foo123';

select * from user where user like '%ejbca%'

Host User Password Select_priv Insert_priv Update_priv Delete_priv Create_priv Drop_priv Reload_priv Shutdown_priv
Process_priv File_priv Grant_priv References_priv Index_priv Alter_priv Show_db_priv Super_priv Create_tmp_table_priv
Lock_tables_priv Execute_priv Repl_slave_priv Repl_client_priv Create_view_priv Show_view_priv Create_routine_priv
Alter_routine_priv Create_user_priv ssl_type ssl_cipher x509_issuer x509_subject max_questions max_updates max_connections
max_user_connections

10.1.1.10 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

10.1.1.11 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

localhost ejbca *77805DB3940553564EF23E5EB2231A1BEB02EFC3 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

10.1.1.12 ejbca *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N N N N N N
N N 0 0 0 0

I-8021-PKI 20090311 15:57:53 smartcard20-node1: INFO - access rules inserted into mysql db ok
```

1.9 Create mysql-replication-user on node1

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./803-create-mysql-replication-user.sh

Host User Password Select_priv Insert_priv Update_priv Delete_priv Create_priv Drop_priv Reload_priv Shutdown_priv
```

```
Process_priv File_priv Grant_priv References_priv Index_priv Alter_priv Show_db_priv Super_priv Create_tmp_table_priv
Lock_tables_priv Execute_priv Repl_slave_priv Repl_client_priv Create_view_priv Show_view_priv Create_routine_priv
Alter_routine_priv Create_user_priv ssl_type ssl_cipher x509_issuer x509_subject max_questions max_updates max_connections
max_user_connections

10.1.1.10 ejbcarep *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N Y N N N
N N N 0 0 0 0

10.1.1.11 ejbcarep *1618E643E43E4921AC458E23C5E7728892CCF1A6 N N N N N N N N N N N N N N N N N N N Y N N N
N N N 0 0 0 0

I-8031-PKI 20090311 15:57:58 smartcard20-node1: INFO - Replication access rules inserted into mysql db ok
```

1.10 Create an empty db on node2

- If you got a db (from a previous install/testrun) do drop it then create a empty ejbca db!

- If yo dont got a db, ie this is a clean install, you can skip this step, and goto the creation of the empty db.

- If you are uncertain if there is a db, there is no harm in trying to do this step, it will tell you there was nothing to drop!

on node2 run:

```
username@smartcard20-node2:/usr/local/syscheck/database-replication> ./801-drop-existing-ejbca-db.sh

root's password:

are you really sure you want to drop and replace the ejbca db on this host?

enter 'im-really-sure' (without the '-') to continue or ctrl-c to abort

im really sure

I-9041-PKI 20090309 14:44:11 smartcard20-node1: INFO - Backed up db ok (file:/backup/mysql/ejbcabackup-2009-03-
09_14.44.10.sql.gz)

Dropping the database is potentially a very bad thing to do.

Any data stored in the database will be destroyed.

Do you really want to drop the 'ejbca' database [y/N] y

Database "ejbca" dropped

I-8011-PKI 20090309 14:44:13 smartcard20-node1: INFO - Dropped the db ok
```

And create an empty one:

on node2 run:

```
username@smartcard20-node2:/usr/local/syscheck/database-replication> sudo ./800-create-mysql-ejbca-db.sh

I-8001-PKI 20090309 14:44:18 smartcard20-node2: INFO - Created the db ok
```

1.11 Configure mysql server for replication on node1

Edit /etc/my.cnf and set the following options:

on node1 edit:

```
# Replication Master Server (default)

# binary logging is required for replication

log-bin=mysql-bin

# required unique id between 1 and 2^32 - 1

# defaults to 1 if master-host is not set

# but will not function as a master if omitted

server-id = 1
```

1.12 Configure mysql server for replication on node2

Edit /etc/my.cnf and set the following options:

on node2 edit:

```
# Replication Master Server (default)

# binary logging is required for replication

log-bin=mysql-bin

But do set server-id = 2

# required unique id between 1 and 2^32 - 1

# defaults to 1 if master-host is not set

# but will not function as a master if omitted

server-id = 2
```

1.13 Configure the virtual IP

Verify the VIP configuration in config/common.conf

Check the interface name of the host:s ipaddress

```
# ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0c:29:c2:1d:f9

          inet addr:192.168.31.146  Bcast:192.168.31.255  Mask:255.255.255.0

          inet6 addr: fe80::20c:29ff:fec2:1df9/64 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:39796 errors:0 dropped:0 overruns:0 frame:0

          TX packets:58761 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:2834743 (2.7 MiB)  TX bytes:41442017 (39.5 MiB)

          Interrupt:19 Base address:0x2000
```

Verify the interface set in config/common.conf

```
IF_VIRTUAL="eth0:0"
```

test the activate / deactivate scripts

remove VIP from node1

```
smartcard20-node1:/usr/local/syscheck/related-enabled # ./912_deactivate_VIP.sh -s

I-9123-PKI 20100615 17:42:05 smartcard20-node1: INFO - While deactivating, the VIP was already NOT active on this host
```

Activate VIP on node1 again

```
smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled# ./911_activate_VIP.sh -s

I-9111-PKI 20100615 17:44:22 smartcard20-node1: INFO - Activate VIP run successfully
```

Check if the VIP is configured that the alias interface is showing up (eth0:0) and has the correct ip, if the VIP should NOT be activated, the interface shall not be shown in ifconfig.

```
# ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0c:29:c2:1d:f9

          inet addr:192.168.31.146  Bcast:192.168.31.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec2:1df9/64 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:39796 errors:0 dropped:0 overruns:0 frame:0

          TX packets:58761 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:2834743 (2.7 MiB)  TX bytes:41442017 (39.5 MiB)

          Interrupt:19 Base address:0x2000


eth0:0    Link encap:Ethernet  HWaddr 00:0c:29:c2:1d:f9

          inet addr:192.168.0.10  Bcast:192.168.0.255  Mask:255.255.255.0

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          Interrupt:19 Base address:0x2000
```

2 Production interrupting steps to sync db from node1 to node2

2.1 Stop jboss application server

NOTE: from this step the service is unavailable!

Stop jboss on node1 and node2

on node1 run:

```
smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled# /etc/init.d/jboss stop

Stopping JBoss: …... done
```

on node2 run:

```
smartcard20-node2:/usr/local/certificate-services/syscheck/related-enabled# /etc/init.d/jboss stop

Stopping JBoss: …...done
```

2.2 Restart mysql server to make replication options take effect

NOTE: this step is only needed if changes has been done to configuration of mysql-server (/etc/my.cnf)

Restart mysql on both nodes

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # /etc/init.d/mysql restart

Restarting service MySQL

Shutting down service MySQL done

Starting service MySQL done
```

on node2 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # /etc/init.d/mysql restart

Restarting service MySQL
```

```
Shutting down service MySQL done

Starting service MySQL done
```

2.3 Lock the database

Now we lock the tables from writes, read statements will still work.

NOTE: This console will be occupied with this command until the steps below are done so you need screen or two terminals. This is because the *LOCK TABLES* command will only last while the session is active, so if you exits the mysql-console the lock is automaticly unlocked, dont do that !

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./811-master-node-flush-tables-with-read-lock.sh

Connecting to localhost, run this on master only!

Enter manually:

FLUSH TABLES WITH READ LOCK;

keep the console open until the last step is done

then enter:

UNLOCK TABLES

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 26

Server version: 5.0.67 SUSE MySQL RPM

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> FLUSH TABLES WITH READ LOCK;

Query OK, 0 rows affected (0.00 sec)

mysql>
```

keep this console running, until instructed to exit or run the unlock tables command.

2.4 Make a database backup

on node1 run:

```
smartcard20-node1:/usr/local/syscheck-1.4.3b3-cluster/related-enabled # ./904_make_mysql_db_backup.sh -s

I-9041-PKI 20090309 13:48:14 smartcard20-node1: INFO - Backed up db ok (file:/backup/mysql/ejbcabackup-2009-03-09_13.48.13.sql.gz)
```

Write down the name of this database backup in your protocol as the last db backup before this change. If anything would go wrong this is the backup to revert to!!!

2.5 Transfer the database-backup to node2

Transfer it to node2:

on node1 run:

```
jboss@smartcard20-node1:~> scp /backup/mysql/ejbcabackup-2009-03-09_13.44.55.sql.gz smartcard20-node2:

ejbcabackup-2009-03-09_13.44.55.sql.gz 100% 1459KB 1.4MB/s 00:00
```

OR use ./906_ssh-copy-to-remote-machine.sh
on node1 run:

```
smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled# ./906_ssh-copy-to-remote-machine.sh --help

smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled# ./906_ssh-copy-to-remote-machine.sh -s /backup/mysql//default/ejbcabackup-2010-06-15_16.03.18.sql.gz smartcard20-node2 /tmp jboss /home/jboss/.ssh/id_rsa

I-9061-PKI 20100615 16:05:30 smartcard20.demo: INFO - file transfered ok
```

2.6 Restore mysql database on node2

NOTE: Upto and including version 1.5.15 of syscheck 920_restore_mysql_db_from_backup.sh don't handle the case that there is no database very good, so be sure to create a empty one first as a work around until this is fixed

on node2 run:

```
root@smartcard20-node2:/usr/local/syscheck/related-enabled # ./920_restore_mysql_db_from_backup.sh -s /tmp/ejbcabackup-2010-06-15_16.03.18.sql.gz

now we'll backup the current database before we restore the one you specified

I-9041-PKI 20090309 14:51:51 smartcard20-node2: INFO - Backed up db ok (file:/backup/mysql/ejbcabackup-2009-03-09_14.51.50.sql.gz)

restoring the db from /home/jboss/ejbcabackup-2009-03-09_13.44.55.sql.gz

I-9202-PKI 20090309 14:51:55 smartcard20-node2: INFO - Restored the db from file (/home/jboss/ejbcabackup-2009-03-09_13.44.55.sql.gz)
```

2.7 Make node1 master

Now make node1 take the role of mysql master!!

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./804-make-mysql-server-act-as-master.sh

Are you sure you want to make this mysql server act as mysql master?

enter 'im-really-sure' (without the '-') to continiue or ctrl-c to abort

im really sure

I-8041-PKI 20090312 16:34:20 smartcard20-node1: INFO - Mysql server made to act as a master
```

Run the show master status command and note the log: “File” and “position”, you will need them in the next step.

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh

File Position Binlog_Do_DB Binlog_Ignore_DB

mysql-bin.000001 98

I-8101-PKI 20090311 15:26:32 smartcard20-node1: INFO - Master status shown

*************************** 1. row ***************************

Id: 3

User: root

Host: localhost

db: mysql

Command: Query

Time: 0

State: NULL

Info: SHOW PROCESSLIST

I-8101-PKI 20090311 15:26:32 smartcard20-node1: INFO - Master status shown

I-8101-PKI 20090311 15:26:32 smartcard20-node1: INFO - Master status shown
```

2.8 Make node 2 slave

Now it's time to start the slave

on node2 run:

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./805-make-mysql-server-act-as-slave.sh

Are you sure you want to make this mysql server act as mysql slave?

Press enter to continiue, ctrl-c to abort



now you need to run 810-show-mysql-master-status.sh on the master node

For a first time setup (master has never had a slave) default file=" and pos=4 is the values to use

then enter File and Position

Enter Log File default:[]>

mysql-bin.000001

Enter Log Pos default:[4]>

98

I-8051-PKI 20090312 16:48:09 smartcard20-node2: INFO - Mysql server made to act as a slave
```

2.9 Unlock master database tables

In the console used to lock the tables in previous step enter the following commands:
on node1 run:

```
mysql> UNLOCK TABLES;

Query OK, 0 rows affected (0.00 sec)

mysql>
```

2.10 Verify replication

on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh
```

```
File Position Binlog_Do_DB Binlog_Ignore_DB

mysql-bin.000001 4439

I-8101-PKI 20090311 16:02:33 smartcard20-node1: INFO - Master status shown

*************************** 1. row ***************************

Id: 73

User: ejbcarep

Host: 10.15.251.247:6265

db: NULL

Command: Binlog Dump

Time: 1015

State: Has sent all binlog to slave; waiting for binlog to be updated

Info: NULL
```

Here one or more connections can be described, but those are from other clients than the replication client so they are not important at this step.

```
*************************** 2. row ***************************

Id: 88

User: ejbca

Host: localhost:24797

db: ejbca

Command: Sleep

Time: 306

State:

Info: NULL



Info: SHOW PROCESSLIST

I-8101-PKI 20090311 16:02:33 smartcard20-node1: INFO - Master status shown

I-8101-PKI 20090311 16:02:33 smartcard20-node1: INFO - Master status shown
```

Check the slave

on node2 run:

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./809-show-mysql-slave-status.sh

Slave_IO_State Master_Host Master_User Master_Port Connect_Retry Master_Log_File Read_Master_Log_Pos
Relay_Log_File Relay_Log_Pos Relay_Master_Log_File Slave_IO_Running Slave_SQL_Running Replicate_Do_DB
Replicate_Ignore_DB Replicate_Do_Table Replicate_Ignore_Table Replicate_Wild_Do_Table Replicate_Wild_Ignore_Table
Last_Errno Last_Error Skip_Counter Exec_Master_Log_Pos Relay_Log_Space Until_Condition Until_Log_File Until_Log_Pos

Master_SSL_Allowed Master_SSL_CA_File Master_SSL_CA_Path Master_SSL_Cert Master_SSL_Cipher Master_SSL_Key
Seconds_Behind_Master

Waiting for master to send event 10.15.251.246 ejbcarep 3306 60 mysql-bin.000001 4439 smartcard20-node2-relay-bin.000003
2057 mysql-bin.000001 Yes Yes 0 0

4439 2057 None 0 No 0

I-8091-PKI 20090311 16:02:25 smartcard20-node2: INFO - Slave status shown
```

2.11 Write to the test table and verify both servers answers the same number

NOTE: this command will use localhost as master so never run this on the slave-host then the replication of the test table will stop!

on node1(master-node) run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./807-test-table-create-table.sh

creating the test table:

smartcard20-node1:/usr/local/syscheck/database-replication # ./808-test-table-update-and-check-master-and-slave.sh

cleaning and inserting new val: 1236784263

values from 10.15.251.246

value

1236784263

values from 10.15.251.247

value

1236784263
```

2.12 Change datasource config in jboss

Set jboss datasource on node1

on node1 run:

```
smartcard20-node1:/usr/local/certificate-services/syscheck/database-replication# ./806-change-active-mysql-server-in-jboss-datasource.sh node1

ejbca-ds.xml in jboss switched host to 10.1.1.10

remember to restart jboss when you want the change to take effect
```

Set jboss datasource on node2

on node2 run:

```
smartcard20-node2:/usr/local/certificate-services/syscheck/database-replication# ./806-change-active-mysql-server-in-jboss-datasource.sh node1

ejbca-ds.xml in jboss switched host to 10.1.1.10

remember to restart jboss when you want the change to take effect
```

2.13 Start Jboss application server

Start jboss on node1

on node1 run:

```
smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled# /etc/init.d/jboss start

Starting JBoss using Java from /usr/local/certificate-services/java: Waiting for jboss server to start:......done
```

Start jboss on node2

on node2 run:

```
smartcard20-node2:/usr/local/certificate-services/syscheck/related-enabled# /etc/init.d/jboss start

Starting JBoss using Java from /usr/local/certificate-services/java: Waiting for jboss server to start:......done
```

2.14 Activate CA:s

Either activate CA:s in EJBCA / The Admin Console or with Syscheck

With syscheck you need to config PIN-codes into config/common.conf

on node1 run:

```
smartcard20-node1:/usr/local/certificate-services/syscheck/related-enabled#   ./909_activate_CAs.sh -s

Screenonly output:
```

```
Activating CA :  eIDCA (./bin/ejbca.sh ca activateca eIDCA 1111)

Using JBoss JNDI provider...

I-9091-PKI 20100615 17:27:04 smartcard20.demo: INFO - Activate CA:s run successfully

[...]
```

NOTE: from this step on the service is again available!

3 Fail over and fail back

3.1 Fail over, make the slave master

Master has problems, thus we need to make the slave accept updates i.e. make it mysql master

3.1.1 Simulate master problems

shut down mysql

```
smartcard20-node1:/usr/local/syscheck/database-replication # /etc/init.d/mysql stop
```

3.1.2 Shut down jboss on both nodes

```
smartcard20-node1:/usr/local/syscheck/database-replication # /etc/init.d/jboss stop

smartcard20-node2:/usr/local/syscheck/database-replication # /etc/init.d/jboss stop
```

3.1.3 Diagnose slave

Slave now shows it can't connect to master:

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./809-show-mysql-slave-status.sh

Slave_IO_State Master_Host Master_User Master_Port Connect_Retry Master_Log_File Read_Master_Log_Pos Relay_Log_File
Relay_Log_Pos Relay_Master_Log_File Slave_IO_Running Slave_SQL_Running Replicate_Do_DB Replicate_Ignore_DB
Replicate_Do_Table Replicate_Ignore_Table Replicate_Wild_Do_Table Replicate_Wild_Ignore_Table Last_Errno Last_Error
Skip_Counter Exec_Master_Log_Pos Relay_Log_Space Until_Condition Until_Log_File Until_Log_Pos

Master_SSL_Allowed Master_SSL_CA_File Master_SSL_CA_Path Master_SSL_Cert Master_SSL_Cipher Master_SSL_Key
Seconds_Behind_Master

Reconnecting after a failed master event read 10.15.251.246 ejbcarep 3306 60 mysql-bin.000001 822 smartcard20-node2-relay-
bin.000002 235 mysql-bin.000001 No Yes 0

0 822 235 None 0 No NULL

I-8091-PKI 20090312 16:33:15 smartcard20-node2: INFO - Slave status shown
```

3.1.4 Promote node2 to master

Now make this node take the role of mysql master

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./804-make-mysql-server-act-as-master.sh

Are you sure you want to make this mysql server act as mysql master?

enter 'im-really-sure' (without the '-') to continiue or ctrl-c to abort

im really sure

I-8041-PKI 20090312 16:34:20 smartcard20-node2: INFO - Mysql server made to act as a master
```

Check the master status

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh

File Position Binlog_Do_DB Binlog_Ignore_DB

mysql-bin.000001 98

I-8101-PKI 20090312 16:34:35 smartcard20-node2: INFO - Master status shown

*************************** 1. row ***************************

Id: 64

User: ejbca

Host: smartcard20-node2.demo:23152

db: ejbca

Command: Sleep

Time: 313

State:

Info: NULL
```

Here one or more connections can be described, but those are from other clients than the replication client so they are not important at this step.

```
*************************** 2. row ***************************

Id: 65

User: ejbca

Host: smartcard20-node2.demo:23153

db: ejbca

Command: Sleep

Time: 313

State:

Info: NULL

I-8101-PKI 20090312 16:34:35 smartcard20-node2: INFO - Master status shown

I-8101-PKI 20090312 16:34:35 smartcard20-node2: INFO - Master status shown
```

Since there is no slave we wont see that process in the list!

3.1.5 Move the virtual IP to node2

Failover VIP , remove VIP from node1

```
smartcard20-node1:/usr/local/syscheck/related-enabled # ./912_deactivate_VIP.sh -s

I-9123-PKI 20100615 17:42:05 smartcard20.demo: INFO - While deactivating, the VIP was already NOT active on this host
```

Activate VIP on node2

```
smartcard20:/usr/local/certificate-services/syscheck/related-enabled# ./911_activate_VIP.sh -s

I-9111-PKI 20100615 17:44:22 smartcard20.demo: INFO - Activate VIP run successfully
```

Failover jboss datasource configuration on node2

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./806-change-active-mysql-server-in-jboss-datasource.sh node2

ejbca-ds.xml in jboss switched host to 10.1.1.11

remember to restart jboss when you want the change to take effect
```

todo: start jboss and verify funcionallity

3.1.6 start jboss on node2

```
smartcard20-node2:/usr/local/syscheck/related-enabled # /etc/init.d/jboss start
```

3.1.7 Verify functionallity

activate CA:s
issue a CRL for each CA, write down the CRL number before(eg: 1) and after issuance(eg.:2), the CRL-number should be increased by one. after failback this number shall remain at the higer one eg. 2.
optionally issue a test certificate to verify node 2 is working.

3.2 Fail-back, make the old master master again

3.2.1 Lock tables

NOTE: This console will be occupied with this command until the steps below are done so you need screen or two terminals. This is because the *LOCK TABLES* command will only last while the session is active, so if you exits the mysql-console the lock is automaticly unlocked, dont do that !

On node2 run

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./811-master-node-flush-tables-with-read-lock.sh

Connecting to localhost, run this on master only!

Enter manually:

FLUSH TABLES WITH READ LOCK;

keep the console open until the last step is done

then enter:

UNLOCK TABLES

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 26

Server version: 5.0.67 SUSE MySQL RPM

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> FLUSH TABLES WITH READ LOCK;

Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

keep this console running, until instructed to exit or run the unlock tables command.

3.2.2 Stop jboss application server

Shutdown jboss on node1

```
smartcard20-node1:/usr/local/syscheck/related-enabled # /etc/init.d/jboss stop
```

Shutdown jboss on node2

```
smartcard20-node2:/usr/local/syscheck/related-enabled # /etc/init.d/jboss stop
```

3.2.3 Make a backup of the data at node2

```
smartcard20-node2:/usr/local/syscheck/related-enabled # ./904_make_mysql_db_backup.sh -s

I-9041-PKI 20090312 16:44:14 smartcard20-node2: INFO - Backed up db ok (file:/backup/mysql/ejbcabackup-2009-03-12_16.44.14.sql.gz)
```

3.2.4 transfer backup to node1

```
smartcard20-node2:/usr/local/syscheck/related-enabled # ./906_ssh-copy-to-remote-machine.sh -s /backup/mysql/ejbcabackup-2009-03-12_16.44.14.sql.gz smartcard20-node1 /tmp jboss /home/jboss/.ssh/id_rsa

I-9061-PKI 20090312 16:44:39 smartcard20-node2: INFO - file transfered ok
```

3.2.5 start MySQL-server

At node1 start the database server (if it was off during recovering the server)

```
username@smartcard20-node1:/usr/local/syscheck-1_5_0/related-enabled> sudo /etc/init.d/mysql start
```

3.2.6 At node1 restore the database

```
username@smartcard20-node1:/usr/local/syscheck-1_5_0/related-enabled> sudo ./920_restore_mysql_db_from_backup.sh -s /tmp/ ejbcabackup-2009-03-12_16.44.14.sql.gz

enter 'im-really-sure' (without the '-') to continiue or ctrl-c to abort
```

```
im really sure

now we'll backup the current database before we restore the one you specified

I-9041-PKI 20090327 11:09:46 sles20sp2-2: INFO - Backed up db ok (file:/backup/mysql/ejbcabackup-2009-03-27_11.09.46.sql.gz)

restoring the db from /backup/mysql/ejbcabackup-2009-03-27_11.09.08.sql.gz

I-9202-PKI 20090327 11:09:47 sles20sp2-2: INFO - Restored the db from file (/backup/mysql/ejbcabackup-2009-03-27_11.09.08.sql.gz)
```

3.2.7 make node1 master again

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./804-make-mysql-server-act-as-master.sh

Are you sure you want to make this mysql server act as mysql master?

enter 'im-really-sure' (without the '-') to continiue or ctrl-c to abort

im really sure

I-8041-PKI 20090312 16:47:00 smartcard20-node1: INFO - Mysql server made to act as a master
```

3.2.8 Check mysql master status

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh

File Position Binlog_Do_DB Binlog_Ignore_DB

mysql-bin.000001 98

I-8101-PKI 20090312 16:47:08 smartcard20-node1: INFO - Master status shown

*************************** 1. row ***************************

Id: 5

User: root

Host: localhost

db: mysql

Command: Query

Time: 0

State: NULL
```

```
Info: SHOW PROCESSLIST

I-8101-PKI 20090312 16:47:08 smartcard20-node1: INFO - Master status shown

I-8101-PKI 20090312 16:47:08 smartcard20-node1: INFO - Master status shown
```

3.2.9 Make the node2 slave again,

enter YOUR values from the last commnd!

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./805-make-mysql-server-act-as-slave.sh

Are you sure you want to make this mysql server act as mysql slave?

enter 'im-really-sure' (without the '-') to continiue or ctrl-c to abort

im really sure

now you need to run 810-show-mysql-master-status.sh on the master node

For a first time setup (master has never had a slave) default file='' and pos=4 is the values to use

then enter File and Position

Enter Log File default:[]>

mysql-bin.000001

Enter Log Pos default:[4]>

98

I-8051-PKI 20090312 16:48:09 smartcard20-node2: INFO - Mysql server made to act as a slave
```

3.2.10 Unlock tables

go back to the console used to lock the tables and enter:

```
mysql> UNLOCK TABLES;

Query OK, 0 rows affected (0.00 sec)

mysql>
```

3.2.11 Failback VIP

deactivate on node2

```
sc20fmv-node2:/usr/local/certificate-services/syscheck/related-enabled # ./912_deactivate_VIP.sh -s

I-9121-PKI 20100610 13:21:00 sc20fmv-node2: INFO - Deactivate VIP run successfully
```

Failback VIP - activate on node1

```
sc20fmv-node1:/usr/local/certificate-services/syscheck/related-enabled # ./911_activate_VIP.sh -s

I-9111-PKI 20100610 13:21:06 sc20fmv-node1: INFO - Activate VIP run successfully
```

3.2.12 Failback jboss datasource configuration

Change to node1 to use as datasource.

Note: to make sure the datasource configuration is correct run this command even though the datasource configuration on node1 could be correct.

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./806-change-active-mysql-server-in-jboss-datasource.sh node1

ejbca-ds.xml in jboss switched host to 192.168.31.140

remember to restart jboss when you want the change to take effect
```

Change to node2 to use as datasource

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./806-change-active-mysql-server-in-jboss-datasource.sh node1

ejbca-ds.xml in jboss switched host to 192.168.31.140

remember to restart jboss when you want the change to take effect
```

3.2.13 Verify replication with the simple test tool

All three values MUST be the same
on node1 run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./808-test-table-update-and-check-master-and-slave.sh

cleaning and inserting new val: 1276170492

values from 192.168.31.140

value

1276170492
```

```
values from 192.168.31.142

value

1276170492
```

3.2.14 start jboss application server

start jboss on node1

```
smartcard20-node1:/usr/local/syscheck/related-available # /etc/init.d/jboss start

Starting JBoss application server: Waiting for jboss server to start:

.................................Jboss server is up and running.
```

start jboss on node2

```
smartcard20-node2:/usr/local/syscheck/related-available # /etc/init.d/jboss start

Starting JBoss application server: Waiting for jboss server to start:

.................................Jboss server is up and running.
```

3.2.15 Verify replication

Verify replication status – The position shall be updated from the initial “98”

on node 1 run

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh

File    Position        Binlog_Do_DB    Binlog_Ignore_DB

mysql-bin.000001        19528

I-8101-PKI 20080504 18:09:18 sc20fmv-node1: INFO - Master status shown

*************************** 1. row ***************************

   Id: 12

  User: ejbcarep

  Host: 192.168.31.142:35185

   db: NULL
```

```
Command: Binlog Dump
   Time: 2018
  State: Has sent all binlog to slave; waiting for binlog to be updated
   Info: NULL
*************************** 2. row ***************************
<...>
I-8101-PKI 20090312 16:50:51 smartcard20-node1: INFO - Master status shown
I-8101-PKI 20090312 16:50:51 smartcard20-node1: INFO - Master status shown
```

show slave status, the slave must have the same file and position (or larger value on replica)

on node 2 run

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./809-show-mysql-slave-status.sh
*************************** 1. row ***************************
             Slave_IO_State: Waiting for master to send event
                Master_Host: 192.168.31.140
                Master_User: ejbcarep
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: mysql-bin.000001
        Read_Master_Log_Pos: 19528
             Relay_Log_File: sc20fmv-node2-relay-bin.000002
              Relay_Log_Pos: 19665
      Relay_Master_Log_File: mysql-bin.000001
           Slave_IO_Running: Yes
          Slave_SQL_Running: Yes
            Replicate_Do_DB:
        Replicate_Ignore_DB:
         Replicate_Do_Table:
     Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
                 Last_Errno: 0
                 Last_Error:
               Skip_Counter: 0
        Exec_Master_Log_Pos: 19528
            Relay_Log_Space: 19665
            Until_Condition: None
             Until_Log_File:
              Until_Log_Pos: 0
         Master_SSL_Allowed: No
         Master_SSL_CA_File:
         Master_SSL_CA_Path:
            Master_SSL_Cert:
          Master_SSL_Cipher:
             Master_SSL_Key:
      Seconds_Behind_Master: 0
I-8091-PKI 20100610 13:50:09 sc20fmv-node2: INFO - Slave status shown
```

4 Verification and Troubleshooting

4.1 Check the log file:

```
smartcard20-node2:/usr/local/syscheck/database-replication # less /var/log/mysqld.log
```

```
090311 15:45:37 mysqld ended

090311 15:45:37 mysqld started
090311 15:45:38 InnoDB: Started; log sequence number 0 43665
090311 15:45:38 [Warning] Neither --relay-log nor --relay-log-index were used; so replication may break when this MySQL serve
r acts as a slave and has his hostname changed!! Please use '--relay-log=smartcard20-node2-relay-bin' to avoid this problem.
090311 15:45:38 [Note] /usr/sbin/mysqld: ready for connections.
Version: '5.0.26' socket: '/var/lib/mysql/mysql.sock' port: 3306 SUSE MySQL RPM
090311 15:45:38 [Note] Slave SQL thread initialized, starting replication in log 'mysql-bin.000001' at position 2617, relay l
og './smartcard20-node2-relay-bin.000001' position: 98
090311 15:45:38 [Note] Slave I/O thread: connected to master 'ejbcarep@10.15.251.246:3306', replication started in log 'mysq
l-bin.000001' at position 2617
```

4.2 Write to the test table and verify both servers answers the same number

NOTE Must be on master node (normally node1) run:

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./807-test-table-create-table.sh
creating the test table:
smartcard20-node1:/usr/local/syscheck/database-replication # ./808-test-table-update-and-check-master-and-slave.sh
cleaning and inserting new val: 1236784263
values from 10.15.251.246
value
1236784263
values from 10.15.251.247
value
```

```
1236784263
```

4.3 Use the show master and show slave scripts

Run this script on the master!

on master-host run: (since this can change back an forth, you need to know which host is master)

```
smartcard20-node1:/usr/local/syscheck/database-replication # ./810-show-mysql-master-status.sh

File Position Binlog_Do_DB Binlog_Ignore_DB

mysql-bin.000001 6106

I-8101-PKI 20090311 16:22:44 smartcard20-node1: INFO - Master status shown

*************************** 1. row ***************************

Id: 73

User: ejbcarep

Host: 10.15.251.247:6265

db: NULL

Command: Binlog Dump

Time: 2226

State: Has sent all binlog to slave; waiting for binlog to be updated

Info: NULL
```

[usally several more connection]

```
*************************** 7. row ***************************

Id: 148

User: root

Host: localhost

db: mysql
```

```
Command: Query

Time: 0

State: NULL

Info: SHOW PROCESSLIST

I-8101-PKI 20090311 16:22:44 smartcard20-node1: INFO - Master status shown

I-8101-PKI 20090311 16:22:44 smartcard20-node1: INFO - Master status shown
```

It’s important the master show status says:

State: Has sent all binlog to slave; waiting for binlog to be updated

And then compare the logfilename and log_pos with slave show status

Run this script on the slave!

on slave-host run:

```
smartcard20-node2:/usr/local/syscheck/database-replication # ./809-show-mysql-slave-status.sh

Slave_IO_State Master_Host Master_User Master_Port Connect_Retry Master_Log_File Read_Master_Log_Pos
Relay_Log_File Relay_Log_Pos Relay_Master_Log_File Slave_IO_Running Slave_SQL_Running Replicate_Do_DB
Replicate_Ignore_DB Replicate_Do_Table Replicate_Ignore_Table Replicate_Wild_Do_Table Replicate_Wild_Ignore_Table
Last_Errno Last_Error Skip_Counter Exec_Master_Log_Pos Relay_Log_Space Until_Condition Until_Log_File Until_Log_Pos

Master_SSL_Allowed Master_SSL_CA_File Master_SSL_CA_Path Master_SSL_Cert Master_SSL_Cipher Master_SSL_Key
Seconds_Behind_Master

Waiting for master to send event 10.15.251.246 ejbcarep 3306 60 mysql-bin.000001 6106 smartcard20-node2-relay-bin.000003
3724 mysql-bin.000001 Yes Yes 0 0

6106 3724 None 0 No 0

I-8091-PKI 20090311 16:23:24 smartcard20-node2: INFO - Slave status shown
```

Here it’s important the slave states it waits for master to send updates, has the right ip to the master and the same Logfilename and log_pos is the same as master show status shows.