Database Replication

On deployments using two YCE servers or more, two YCE databases can be setup using master-master replication. This setup offers redundancy with instant fail over capabilities. Although the installation always creates a master-master configuration, the administrator can assign a primary and a secondary database server to each of the front-ends, allowing any combination of master-slave or master-master relationships.

Pre conditions

- It is assumed you already have cloned the primary server or set up a new one.
  - If you have installed a new system (VM), make sure that the disk is sufficiently sized.
- You've followed the Installation guide, mainly for the network setup, using net_setup.

Configuration

At installation time, the capabilities and database relationships are defined on a per-server basis. The script `/opt/yce/system/`**`yce_setup.pl`** is used set this up in a step-by-step process. This script must be invoked without any options to interact with the user and create the setup profile. This XML document is used in turn to create the configuration files for the various YCE components (like http, mysql, psmon, tftp, etc).

Invoking the script with the `-r` option skips the interactive part and just creates the configuration files and restarts the various components.

sample session

In the sample session below, yce_setup.pl is used to add a second YCE system to an existing server. Both YCE systems support databases and font-end services, using their local databases as primary, an the other server databases as secondary.

A setup where the the database(s) are installed on dedicated YCE servers instead of shared front-end and database roles is fully supported and setup in the same fashion.

> Note 1: It is essential that each database is assigned its own unique id. The default is '1'.
>
> Note 2: NetYCE supports “MariaDB”, a database engine quite similar to Oracles' “MySql”. Since both are near-identical in setup - even the process is named “mysqld” - we refer commonly to MySql although the implementation is “MariaDB”.

```
$ yce_setup.pl
Starting 'yce_setup'
Read yce setup: '/opt/yce/etc/yce_setup.xml'
  Read network setup: '/opt/yce/etc/net_setup.xml'
YCE servers currently in setup:
```

```
  #  Hostname         IP-address           FQDN
* 0) genie            172.17.10.21         genie.netyce.org
  Select the server# to Remove, or 'A' to add, 'C' to continue: [C] a
  Add new server
    Hostname for new server? specter
    Domain name for new server? netyce.org
    IP-address for new server? 172.17.10.20
YCE servers currently in setup:
  #  Hostname         IP-address           FQDN
* 0) genie            172.17.10.21         genie.netyce.org
  1) specter          172.17.10.20         specter.netyce.org
  Select the server# to Remove, or 'A' to add, 'C' to continue: [C]
YCE server roles:
  #  Hostname         Front-end    SSL     HTTPD     URL      Database
Id
* 0) genie            yes          http    non-root  name     yes          1
  1) specter
  Select the server# to change, 'C' to continue: [0] 1
    'specter' is a Front-end server? [N] y
    'specter' uses SSL (y/n)? [N]
    'specter' https runs as root (y/n)? [Y] n
    'specter' is DNS resolvable (y/n)? [N] y
    'specter' is a Database server? [N] y
    Database-id for this database (1/2)? [1] 2
YCE server roles:
  #  Hostname         Front-end    SSL     HTTPD     URL      Database
Id
* 0) genie            yes          http    non-root  name     yes          1
  1) specter          yes          http    non-root  name     yes          2
  Select the server# to change, 'C' to continue: [1] c
YCE server database mapping:
  #  Hostname         Db-Id    Primary          Secondary
* 0) genie            1        genie            specter
  1) specter          2        specter          genie
  Select the server# to change, 'C' to continue: [0] c
Wiki setup:
    'genie' will use the NetYCE public Wiki server? [Y] y

#
# YCE Server overview:
# Name         Domain               IP-address      Database Front-end
Primary-db   Secondary-db
# ----         ------               ----------      -------- ---------- -----
-----   ------------
# genie        netyce.org           172.17.10.21     id=1    yes       genie
specter
# specter      netyce.org           172.17.10.20     id=2    yes
specter      genie
#
```

```
Create the YCE, httpd, and mysql configuration files for the desired systems
  each will be created in /opt/yce/etc
* 0)  'genie'
  1)  'specter'
  Select server# to create, 'A' for all, 'C' to complete: [0] c

Updating 'genie' tool-tree (C)
Updating 'genie' menu-tree (C)
Updating 'genie' encryption keys
Restarting 'yce_xch' (/usr/bin/sudo /opt/yce/system/init/yce_xch)
Restarting 'yce_skulker' (/usr/bin/sudo /opt/yce/system/init/yce_skulker)
Restarting 'yce_tftp' (/usr/bin/sudo /opt/yce/system/init/yce_tftp)
Restarting 'yce_sched' (/usr/bin/sudo /opt/yce/system/init/yce_sched)

Done
```

xml setup file

The `yce_setup` script created the file `/opt/yce/etc/`**`yce_setup.xml`**. The information in this file is considered leading in the YCE configuration file creation process, which can be invoked separately by invoking `yce_setup.pl -r`.

The file created by the sample yce_setup session above resulted in:

```
<setup>
  <override>
    <configs crontab="update" httpd="update" mysql="update" network="update"
/>
    <daemons yce_ibd="disable" yce_nccmd="enable" />
  </override>
  <yce name="genie">
    <database database_id="1" type="mysql" />
    <host domainname="netyce.org" fqdn="genie.netyce.org" hostname="genie"
/>
    <httpd mode="non-root" proto="http" type="apache" urlbase="name" />
    <net ipv4="172.17.10.21" />
    <roles database="yes" frontend="yes" />
    <wiki domain="netyce.com" ip="" local="no" name="yce-wiki" proto="http"
/>
    <yce_db primary_db="genie" secondary_db="specter" />
  </yce>
  <yce name="specter">
    <database database_id="2" type="mysql" />
    <host domainname="netyce.org" fqdn="specter.netyce.org"
hostname="specter" />
    <httpd mode="non-root" proto="http" type="apache" urlbase="name" />
    <net ipv4="172.17.10.20" />
    <roles database="yes" frontend="yes" />
    <yce_db primary_db="specter" secondary_db="genie" />
  </yce>
```

```
</setup>
```

Install a replicating database (server)

Once both (database) systems are defined and activated (a running MySql or MariaDB server), the YCE database must be installed on these servers in such a way that the master-master replication can be initialized.

This is accomplished using the YCE front-end. The “**Admin - System**” menu offers two tools to manage the system and its databases. The “**System status**” tool provides an overview of the various YCE servers and their capabilities as provided by the yce_setup.xml document. It includes a section on the “Database status” regarding the replication and its synchronization.

The “**Db archives**” tool is used to manage the database archives, the (automatic) backups and restores. To install an archive with replication, first requires an archive. This archive can be a download sample NetYCE database, any of the available historical archives listed in the tool, or the current running database. The basic procedure is identical for all cases, only the 'current database' needs some preparation.

There are a few reasons why the 'current database' should be re-installed with replication in mind:
o First time setup of a replicating database
o Non-recoverable Synchronization failure.

After a system crash, networking problem or database corruption one or both YCE databases may report (System status) that the databases are out-of-sync and cannot be automatically resolved or the backlog is too large. In that case, the YCE application manager needs to determine which database should be considered leading. Normally it is quite evident that one database lags the other due to the synchronisation problem. Then proceed to the 'Leading' database server and create a manual archive of that database. Provide a proper description to indicate its origin and purpose (e.g. “Resynchronisation from Genie-server”).

> Note: See the paragraph on “**Repairing table data**” in case of extensive out-of-sync situations.

With the desired archive located or created, the procedure to install these archives and initiate the synchronization is as follows:

1. Ensure no active users are making modifications to the database(s).
2. Create a database backup of the primary server
3. Download the 'target' archive to the local workstation.
4. Upload the 'target' archive to the other YCE (database) server so the same archive is available on both servers.
5. On both systems, disable and stop the 'yce_skulker' process. In the System status page, click first `Set ignore flag` button for this process, then `Stop yce_skulker`. This prevents database change-over and replication-verifications db changes during the restores.
6. Restore the 'target' archive on both servers in succession. Preferably not simultaneously and within a few minutes of one another.
7. From the “System status” tool, click the `Start replication slave` button. First on one server, then the other.

8. Verify if the synchronization status is “OK” on both servers. If it is not, use the `Skip SQL replication error` button if available, notice the size of the backlog to see if repeated use is sensible.

Notes

- It is normal to be forced to logout and re-login after a database restore because the current session cookie cannot be verified in the restored database.
- The skulker process is automatically re-enabled after restoring a database.
- Make sure to initially copy local modified files if these are being used, e.g. csv_api.ini, tool_setup.xml, sched_rules.conf. If files are updated at a later point in time using the GUI, they will be automatically synced to other systems.

Please consult the Database archives page to get familiar with the “Db archives” tool.
The “System status” tool is described in the System status page.

Repairing table data

If it is suspected that there are (extensive) differences between two out-of-sync databases that need to be resolved before one database can be defined the 'leading' and be restored, the cli tool `/opt/yce/system/ck_dbsync.pl` can be used.

It allows for table-by-table, row-by-row compare and updates between two YCE databases that have been setup for replication. The tool can therefore also be used to determine if the synchronization of one or all tables is truly “in-sync”.

```
$ ck_dbsync.pl -h
Compare data between synced YCE databases
usage: /opt/yce/system/ck_dbsync.pl options
  options -x -s
          -x -c
          -x -k [-r]      [-t table_name]
          -x -u [-r] [-T] [-t table_name]
          -x -L [-r]
          -x -F [-r]
          -x -R
          -x -Q
   -x        execute: mandatory option. Just to prevent accidental use
   -s        summary: return row count and size of each table
   -c        check tables: defragment and/or repair. Only on local database
   -k        keys only. Insert missing records from PRI into SEC
   -u        update full. Insert and update all SEC records from PRI
   -T        update using timestamps. If PRI timestamp is more recent than
SEC (use with -u)
   -t table  table-name only. Use 'db.table_name' or 'table_name' format
   -r        reverse: switch PRI (default local) and SEC (remote) databases
   -L        lookup: reset Job, Task and other ID's (after finishing table
updates!)
   -F        force: reset the master and slave configuration for PRI <- SEC
   -R        repair-all: perform a sequence to full-repair using cycles
```

```
            of the various options for both databases (-F, -uT, -L, -F).
   -Q       repair-quick: perform a sequence to key-repair using cycles
            of the various options for both databases (-F, -k, -L, -F).
   -h       help: this help message

   The '-x' option is mandatory
```

On larger tables the processing is time consuming (e.g. 20 minutes) and cpu intensive (e.g. 95% cpu for mysqld). So it is best not performed during production hours.

It is highly recommended to create a manual backup archive of the databases (yes, both) before using the 'update' option. If mistakes are made as to the direction of the update (which db is considered leading), it can only be undone by restoring it.

When using the update option (`-u`), please be aware that all SQL update and insert statements will be executed directly on the local or remote database. When replication is in operation these operations will also be synced - usually resulting in a synchronisation error due to a key violation. (If a missing record is added remotely, the replication will also try to install it locally - where it already existed).

In general, when using this tool to update tables, the resulting database should afterwards be re-installed for synchronization using the above procedure. Alternatively the ck_dbsync tool can be used to reset the replication log files and administration using its `-F` option.

This tool also logs all actions in the file `/var/opt/yce/logs/ck_dbsync.log` and auto-rotates each run. The SQL statements are logged in `/var/opt/yce/logs/ck_dbsync_db.log`.

> Please consult the page NetYCE Database Re-synchronization procedure for a detailed procedure how to use this tool.

Change Replication Schema

The replication schema is based on a setup where the database servers are both a Master and a Slave. SQL UPDATE and INSERT statements executed on one database are forwarded to the other (the master function) where the SQL statement in turn is executed as well (the slave function). If both databases do this properly, the databases stay in sync.

The technical implementation involves 'binlog' files on the master and a slave that maintains its position in the binlog it is reading from. Should the connection between the databases be lost for a while, the slave stopped or the database temporarily be down (like doing backups), nothing gets lost since the backlog can be processed at any time.

To create and process the binlogs, each database server has a configuration what databases and tables to replicate. This configuration is maintained in the mysql configuration file, **`/etc/my.cnf`**.

The relevant section is shown below:

```
#  Replication Master Server
#  binary logging is required for replication
log-bin = mysql-bin
```

```
binlog-ignore-db = alerts

#  use no checksums when replicating against pre-5.6 versions
# binlog-checksum = none

# Replication Slave
replicate-wild-do-table = YCE.%
replicate-wild-do-table = NMS.%
replicate-wild-do-table = NCCM.%
replicate-wild-do-table = CMDB.%
replicate-ignore-db = mysql
replicate-ignore-db = alerts
```

If (permanent) changes need to be made to the replication it should be done by editing this configuration file. The entries like `replicate-wild-do-table = YCE.%` define that all tables of the database 'YCE' should be replicated. Adding or removing such a line will add or remove the replication setup.

It is essential that the replication setup of both servers are identical! Changes made on one server should be copied on the other.

The change becomes effective when the mysql server is restarted.

```
yce@yceseven /opt/yce/etc
$ go restart mysql
restarting Daemon 'mysqld'
mysqld: 28544 28738
  kill: /usr/bin/sudo /sbin/service mysql stop
  wait stop 'mysqld':
  spawn: /usr/bin/sudo /sbin/service mysql start
  wait start 'mysqld': 10202 10396
done
```

Since the `/etc/my.cnf` file is auto-generated by the `yce_setup.pl` script, care should be taken that the manual changes are not undone at the next NetYCE update. This is accomplished by informing the setup script it may not overwrite the mysql configuration.

This requires a small alteration of the configuration file `/opt/yce/etc/yce_setup.xml`. The example below shows the top section:

```
<setup>
  <override>
    <configs crontab="update" httpd="update" mojo="update" mysql="update"
network="update" />
    <daemons vsftpd="enable" yce_ibd="disable" yce_nccmd="enable" />
  </override>
```

The modification consists of changing `mysql=“update”` into **`mysql=“keep”`**:

```
<setup>
  <override>
```

```
    <configs crontab="update" httpd="update" mojo="update" mysql="keep"
network="update" />
    <daemons vsftpd="enable" yce_ibd="disable" yce_nccmd="enable" />
  </override>
```

Temp Change Replication Schema

The above directions serve to make permanent changes to the replication schema. It is also possible to briefly change it. These directions follow below. Keep in mind though, the change is lost when the mysql process is stopped. This regularly happens daily for a maintenance backup between 23:00 and 24:00.

> The required mysql superuser password will not be shared here. Please contact a technical NetYCE representative to obtain it.

Execute the steps below using the NetYCE functional account, `yce`. These steps illustrate to remove the NCCM and CMDB databases from the replication schema.

```
yce@yceseven /opt/yce/bin
$ mysql -u netYCE -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12302
Server version: 10.0.32-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

```
MariaDB [(none)]> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
               Slave_IO_State: Connecting to master
                  Master_Host: 72.17.10.21
                  Master_User: replication
                  Master_Port: 3306
                Connect_Retry: 60
:::
          Replicate_Ignore_DB: mysql,alerts
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table: YCE.%,NMS.%,NCCM.%,CMDB.%
  Replicate_Wild_Ignore_Table:
:::
```

```
1 row in set (0.00 sec)
```

```
MariaDB [(none)]> STOP SLAVE;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
MariaDB [(none)]> SET GLOBAL Replicate_Wild_Do_Table = "YCE.%,NMS.%";
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
               Slave_IO_State:
                  Master_Host: 72.17.10.21
                  Master_User: replication
                  Master_Port: 3306
                Connect_Retry: 60
:::
          Replicate_Ignore_DB: mysql,alerts
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table: YCE.%,NMS.%
  Replicate_Wild_Ignore_Table:
:::
1 row in set (0.00 sec)
```

```
MariaDB [(none)]> START SLAVE;
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> \q
Bye
yce@yceseven /opt/yce/bin
$
```

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:reference:database:database_replication

Last update: **2024/07/03 12:31**