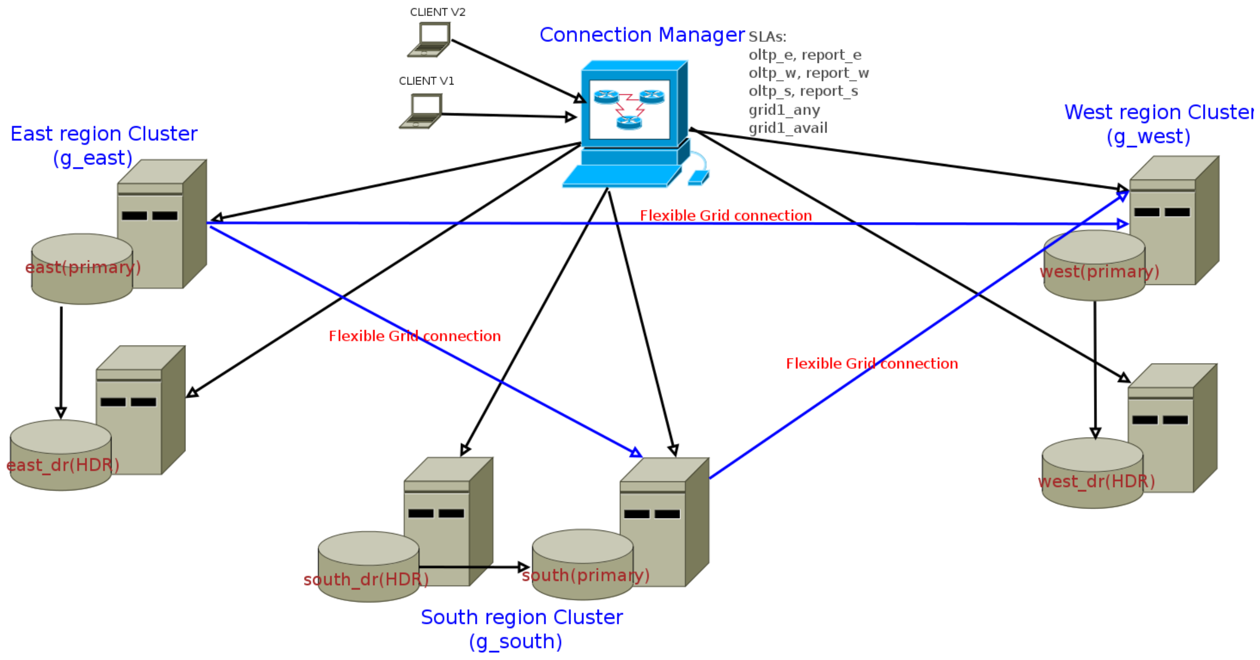
Rolling Schema Upgrade Hands on Lab Instructions

Instructor: Nagaraju Inturi, HCL, nagaraju.inturi@hcl.com

Story:



Lab include 7 node docker containers. Setup flexible grid between three Informix servers in East, West and South regions. Flexible grid allows seamless DDL and DML replication, and supports database write activity at each server. Flexible grid also supports rolling schema upgrade with minimal downtime to client applications.

Each of these three servers configured with HDR secondary node for disaster recovery.

Unified Connection Manager is configured to monitor East, West and South region clusters, and Flexible grid.

Client applications connect to flexible grid cluster through Connection manager grid_oltp1/grid_oltp2 SLA definitions.

Each cluster has “oltp” SLA to redirect clients –to work with latest data-- to current primary, and “report” SLA to redirect clients – who may be ok working with little bit old data, mainly report activity-- to HDR server.

Grid SLA definition is used to load balance clients across East, West and South region servers. Two grid SLA definitions(grid_oltp1, grid_oltp2) are configured to support rolling schema upgrade procedure.

High level steps for the Lab exercise:

1) Create and replicate stores demo database schema through Flexible GRID
2) Auto registering new tables with Enterprise Replication
3) Load data into stores demo database tables
4) Make sure data gets replicated to all three flexible grid servers
5) Verify data across all grid servers using ‘cdr check replset’ command,
6) Test connection redirection logic through connection manager
7) Start client application **clientV1.sh** by connecting to grid_oltp1 SLA
8) Upgrade schema at East region servers
    a. Update grid_oltp1 SLA definition to remove g_east (group name for east server) server.
    b. Force single user mode for server g_east to kill existing sessions
    c. Change server mode back to On-Line mode
    d. Check and make sure new connections go to g_west and g_south servers
    e. Disconnect ER network connection between g_east and g_west, and also between g_east and g_south
    f. Upgrade schema
    g. Update grid_oltp2 SLA definition to redirect updated client application(clientV2.sh) to g_east server.
    h. Start **clientV2.sh** application and make sure that clientV2.sh connections go to g_east server.
9) Upgrade schema at West region servers
    a. Update grid_oltp1 SLA definition to remove g_west server.
    b. Force single user mode for g_west server to kill existing sessions

c. Change server mode back to On-Line mode

d. Check and make sure **clientV1.sh** connections only go to g_south server

e. Disconnect ER network connection between g_west and g_south servers.

f. Reconnect network connection between g_east and g_west servers and check and make sure schema changes replicated from g_east to g_west server.

g. Update grid_oltp2 SLA definition to redirect updated client application(**clientV2.sh**) to both g_east and g_west servers.

10) Upgrade schema at South region servers

    a. Shutdown clientV1.sh application.

    b. Update grid_oltp1 SLA definition to remove g_south server.

    c. Force single user mode for g_south server to kill existing sessions

    d. Change server mode back to On-Line mode

    e. Reconnect network connection between g_east and g_south servers

    f. Check and make sure schema changes replicated from g_east server to g_south server

    g. Reconnect network connection between g_west and g_south servers.

    h. Update grid_oltp2 SLA definition to redirect updated client application(**clientV2.sh**) to all regions: g_east, g_west and g_south servers.

11) Verify data across all servers

Docker container names:

east : Flexible Grid server for east region

east_dr : DR server for east region

west: Flexible Grid server for west region

west_dr: DR server for west region

south: Flexible Grid server for south region

south_dr : DR server for south region.

cm1: Container for unified connection manager, and client applications

Home directory: cd ~/rolling_schema_upgrade

Scripts:

login.sh <container name>: Login to docker container. Logs in as root user. Informix environment was already set.

```
$ ./login.sh east

--shell

[root@east ibm]# id

uid=0(root) gid=0(root) groups=0(root)

[root@east ibm]# su informix

[informix@east ibm]$ id

uid=200(informix) gid=102(informix) groups=102(informix)

[informix@east ibm]$
```

status.sh <container name>: Shows server status

```
$ ./status.sh east

IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 00:42:49 -- 16446

8 Kbytes
```

Lab exercise starts here:

Verify east, west and south standard server status:

Note that these three are independent standard servers. SQLHOSTS file and trusted hosts file pre-configured to setup Enterprise replication and DR servers.

1) *$ ./status.sh east*

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 02:43:26 -- 16446

8 Kbytes
```

2) *$ ./status.sh west*

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 02:43:21 -- 16446

8 Kbytes
```

3) *$ ./status.sh south*

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 02:43:14 -- 16446

8 Kbytes
```

Login to east container and review sqlhosts file content.

4) $ ./login.sh east

```
5)  [root@east ibm]# su informix

6) [informix@east ibm]$ onstat -

IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line -- Up 02:47:10 -- 16446

8 Kbytes
```

7) [informix@east ibm]$ cat $INFORMIXSQLHOSTS

```
g_east group - - i=1

east onsoctcp 172.20.0.10 60000 g=g_east

east_dr onsoctcp 172.20.0.11 60000 g=g_east

g_west group - - i=2

west onsoctcp 172.20.0.12 60000 g=g_west

west_dr onsoctcp 172.20.0.13 60000 g=g_west

g_south group - - i=3

south onsoctcp 172.20.0.14 60000 g=g_south

south_dr onsoctcp 172.20.0.15 60000 g=g_south
```

Setup Enterprise replication between east, west, and south region servers:

Note: Make sure you are still in east container logged in as user Informix:

```
8) [informix@east ibm]$ id

        uid=200(informix) gid=102(informix) groups=102(informix)
```

Define ER for east server:

```
9) [informix@east ibm]$ cdr define serv -c east -I g_east
```

Check ER status:

```
10) [informix@east ibm]$ cdr list server

        SERVER           ID STATE    STATUS     QUEUE  CONNECTION CHANGED

        -----------------------------------------------------------------------

        g_east            1 Active   Local         0
```

Now define ER for west server sync with east server:

Note: Even though we logged into "east" container, notice that we are connecting to west server using –connect (-c) option.

```
11) [informix@east ibm]$ cdr define serv -c west -I -S g_east g_west
```

Verify ER status:

```
12) [informix@east ibm]$ cdr list server

        SERVER           ID STATE    STATUS     QUEUE  CONNECTION CHANGED

        -----------------------------------------------------------------------

        g_east            1 Active   Local         0

        g_west            2 Active   Connected      0 Apr 21 23:09:50
```

Now define ER for south region server sync with east server:

```
13) [informix@east ibm]$ cdr define serv -c south -I -S g_east g_south
```

Verify ER status:

```
14) [informix@east ibm]$ cdr list server

        SERVER              ID STATE    STATUS     QUEUE  CONNECTION CHANGED

        -----------------------------------------------------------------------

        g_east              1 Active   Local        0

        g_south              3 Active   Connected       0 Apr 21 23:11:55

        g_west              2 Active   Connected       0 Apr 21 23:09:50
```

Configure DR servers:

Ok, now we have Enterprise Replication setup between east, west and south region servers. Now lets setup HDR servers for each of these three servers.

Now logout from east container. You will have to run 'exit' command twice to exit from east container.

```
15) [informix@east ibm]$ exit

exit

16) [root@east ibm]# exit

Exit

$
```

Now lets login to east_dr container to setup HDR server using ifxclone command.

```
17) $ ./login.sh east_dr

--shell

[root@east_dr ibm]# id

        uid=0(root) gid=0(root) groups=0(root)
```

Note that this time, we do run command as root.

Review /opt/ibm/clone.sh script content:

18) [root@east_dr ibm]# cat clone.sh

```
 su informix -c "ifxclone -S east -I 172.20.0.10 -P 60000 -t east_dr -i 172.20.0
.11 -p 60000 -L -T -d HDR "
sleep  30
```

```
     Run 'onstat –m' and wait for 'DR: HDR secondary server operational'
message
```

Note that we are running ifxclone command as Informix user, and east region server is our source server. And we are using HDR as our final disposition (-d option).

Ifxclone first starts new server as RSS then it converts it to HDR. After east_dr server starts as HDR, we will have to delete RSS server entry from primary server. ‘ha rss delete’ sysadmin task command does that.

Now execute the clone.sh command as root user to clone HDR server.

19) [root@east_dr ibm]# sh -x clone.sh

```
+ su informix -c 'ifxclone -S east -I 172.20.0.10 -P 60000 -t east_dr -i 172.20.
0.11 -p 60000 -L -T -d HDR '
Restoring clone server east_dr from source server east.
Look at online log for status of clone server...
+ sleep 10
```

Verify east_dr server and HDR status:

20) [root@east_dr ibm]# onstat -g dri

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- Read-Only (Sec) -- Up 00:01:35
-- 164468 Kbytes

Data Replication at 0x45a4a028:
  Type        State       Paired server       Last DR CKPT (id/pg)    Suppo
```

```
          rts Proxy Writes

           HDR Secondary  on          east                  5 / 5        N


           DRINTERVAL   0

           DRTIMEOUT    30

           DRAUTO       0

           DRLOSTFOUND  /opt/ibm/informix/etc/dr.lostfound

           DRIDXAUTO    0

           ENCRYPT_HDR  0

           Backlog      0

           Last Send    2016/04/21 23:22:24

           Last Receive 2016/04/21 23:22:24

           Last Ping    2016/04/21 23:22:20

           Last log page applied(log id,page): 0,0
```

Exit from “east_dr” container.

```
21) [root@east_dr ibm]# exit

exit

$
```

Verify “east” server status:

```
22) $ ./status.sh east

          IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -- Up 03:05:27 -

          - 172660 Kbytes
```

Now let’s setup west_dr HDR server as DR server to west region server.

Login to west_dr container:

23) $ ./login.sh west_dr

--shell

24) [root@west_dr ibm]# id

```
uid=0(root) gid=0(root) groups=0(root)
```

Review clone.sh script:

25) [root@west_dr ibm]# cat clone.sh

```
 su informix -c "ifxclone -S west -I 172.20.0.12 -P 60000 -t west_dr -i 172.20.0
.13 -p 60000 -L -T -d HDR "
sleep  10
```

Run 'onstat –m' and wait for 'DR: HDR secondary server operational' message

Execute clone.sh script as root user:

26) [root@west_dr ibm]# sh -x clone.sh

```
+ su informix -c 'ifxclone -S west -I 172.20.0.12 -P 60000 -t west_dr -i 172.20.
0.13 -p 60000 -L -T -d HDR '
Restoring clone server west_dr from source server west.
Look at online log for status of clone server...
+ sleep 10
```

Now verify HDR status:

27) [root@west_dr ibm]# onstat -g dri

```
IBM Informix Dynamic Server Version 12.10.FC6 -- Read-Only (Sec) -- Up 00:01:54
```

```
-- 172660 Kbytes

Data Replication at 0x45a4a028:
  Type         State      Paired server      Last DR CKPT (id/pg)   Suppo
rts Proxy Writes
  HDR Secondary  on         west                    5 / 8        N

  DRINTERVAL   0
  DRTIMEOUT    30
  DRAUTO       0
  DRLOSTFOUND  /opt/ibm/informix/etc/dr.lostfound
  DRIDXAUTO    0
  ENCRYPT_HDR  0
  Backlog      0
  Last Send    2016/04/21 23:28:23
  Last Receive 2016/04/21 23:28:23
  Last Ping    2016/04/21 23:27:58
  Last log page applied(log id,page): 0,0
```

Exit from west_dr container.

28) [root@west_dr ibm]# exit

Exit

Let’s check west server status:

29) $ ./status.sh west

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -- Up 03:11:05 -
- 172660 Kbytes
```

Now let’s setup south_dr HDR server for south region server:

Login to south_dr container:

30) $ ./login.sh south_dr

```
 [root@south_dr ibm]#
```

Review clone.sh script content:

31) [root@south_dr ibm]# cat clone.sh

```
 su informix -c "ifxclone -S south -I 172.20.0.14 -P 60000 -t south_dr -i 172.20
.0.15 -p 60000 -L -T -d HDR "
sleep  10
```

Run 'onstat –m' and wait for 'DR: HDR secondary server operational' message

Execute clone.sh to clone south_dr HDR server from south region server.

32) [root@south_dr ibm]# sh -x clone.sh

```
+ su informix -c 'ifxclone -S south -I 172.20.0.14 -P 60000 -t south_dr -i 172.2
0.0.15 -p 60000 -L -T -d HDR '
Restoring clone server south_dr from source server south.
Look at online log for status of clone server...
+ sleep 10
```

Verify HDR state:

33) [root@south_dr ibm]# onstat -g dri

```
IBM Informix Dynamic Server Version 12.10.FC6 -- Read-Only (Sec) -- Up 00:02:05
```

```
-- 172660 Kbytes


Data Replication at 0x45a4a028:
  Type           State       Paired server       Last DR CKPT (id/pg)    Suppo
rts Proxy Writes
  HDR Secondary  on          south                      5 / 5         N


  DRINTERVAL   0
  DRTIMEOUT    30
  DRAUTO       0
  DRLOSTFOUND  /opt/ibm/informix/etc/dr.lostfound
  DRIDXAUTO    0
  ENCRYPT_HDR  0
  Backlog      0
  Last Send    2016/04/21 23:34:23
  Last Receive 2016/04/21 23:34:23
  Last Ping    2016/04/21 23:34:18
  Last log page applied(log id,page): 0,0
```

Exit from south_dr container.

34) [root@south_dr ibm]# exit

Exit

Now verify south region server status and make sure onstat show it as primary server.

35) $ ./status.sh south

```
    IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -- Up 03:18:06 -

    - 172660 Kbytes
```

Now we did setup Enterprise replication between east, west and south region servers, and also added HDR server for each of these three servers for disaster recovery purpose.

Flexible grid configuration:

Now let's define grid definition for our flexible grid configuration.

Login to east container:

```
36) $ ./login.sh east

--shell

37) [root@east ibm]# id

    uid=0(root) gid=0(root) groups=0(root)
```

Now change owner to Informix user:

```
38) [root@east ibm]# su informix

[informix@east ibm]$ id

    uid=200(informix) gid=102(informix) groups=102(informix)
```

Verify ER status:

```
39) [informix@east ibm]$ cdr list server

    SERVER          ID STATE   STATUS    QUEUE  CONNECTION CHANGED

    -----------------------------------------------------------------------

    g_east           1 Active   Local         0

    g_south          3 Active   Connected     0 Apr 21 23:11:55

    g_west           2 Active   Connected     0 Apr 21 23:09:50
```

Now define ‘grid1’ grid definition with all three ER servers using –a option:

```
40) [informix@east ibm]$ cdr define grid grid1 -a
```

Now enable ‘grid1’ definition to run grid commands at all three ER servers (using –n option), and grant permissions to informix user for to run grid commands.

```
41) [informix@east ibm]$ cdr enable grid -g grid1 -n g_east -n g_west -n g_south -u informix
```

Check control and send queue to make sure grid command propagated to all three servers.

```
42) [informix@east ibm]$ cdr check queue  -q cntrlq -w 60 -a

        Checking cntrlq queue status for server g_south ...

        Checking cntrlq queue status for server g_west ...

        cntrlq queue status for g_south as of Thu Apr 21 23:45:51 2016:     COMPLETE

        cntrlq queue status for g_west as of Thu Apr 21 23:45:51 2016:      COMPLETE

        Checking cntrlq queue status for server g_east ...

        cntrlq queue status for g_east as of Thu Apr 21 23:45:51 2016:      COMPLETE

43) [informix@east ibm]$ cdr check queue  -q sendq -w 60 -a

        Checking sendq queue status for server g_south ...

        sendq queue status for g_south as of Thu Apr 21 23:45:55 2016:     COMPLETE

        Checking sendq queue status for server g_west ...

        sendq queue status for g_west as of Thu Apr 21 23:45:55 2016:      COMPLETE

        Checking sendq queue status for server g_east ...

        sendq queue status for g_east as of Thu Apr 21 23:45:55 2016:      COMPLETE
```

Now verify grid definition:

```
44) [informix@east ibm]$ cdr list grid grid1 -v
```

```
Grid                    Node                    User
------------------------ ------------------------ ------------------------
grid1                   g_east*                 informix
                        g_south*                informix
                        g_west*                 informix
Details for grid grid1
```

Replicate DDLs and DMLs using Flexible Grid:

Now, using dbaccess, lets connect to grid, and create stores database. Note that once we connected to the grid, all DDL operations gets replicated to all grid servers.

Once we create stores database, lets create sysdbopen() procedure for Informix user within stores database, and set grid environment by default. This allows us to setup grid environment for Informix user by default without needing to set grid environment each time Informix user connects to stores database.

SQL commands for these operations are in sysdbopen.sql file. Let's review SQL statements within this file.

```
45) [informix@east ibm]$ cat sysdbopen.sql

        database sysmaster;

        execute procedure ifx_grid_connect('grid1', 'dbaccessdemo', 1);

        create database stores with log;

        CREATE PROCEDURE informix.sysdbopen()

        execute procedure ifx_grid_connect('grid1', 'dbaccessdemo', 3);

        END PROCEDURE;
```

Note: Third argument '3' to ifx_grid_connect enables server to automatically create new replication definition after creating new table.

Now let's run sysdbopen.sql as use Informix:

```
46) [informix@east ibm]$ id

        uid=200(informix) gid=102(informix) groups=102(informix)

47) [informix@east ibm]$ dbaccess - sysdbopen.sql


        Database selected.

        Routine executed.

        Database closed.

        Database created.

        Routine created.

        Database closed.
```

Now let's make sure control and send queues are empty

```
48)  [informix@east ibm]$ cdr check queue  -q cntrlq -w 60 -a

        Checking cntrlq queue status for server g_south ...

        Checking cntrlq queue status for server g_west ...

        cntrlq queue status for g_south as of Thu Apr 21 23:45:51 2016:     COMPLETE

        cntrlq queue status for g_west as of Thu Apr 21 23:45:51 2016:      COMPLETE

        Checking cntrlq queue status for server g_east ...

        cntrlq queue status for g_east as of Thu Apr 21 23:45:51 2016:      COMPLETE

49) [informix@east ibm]$ cdr check queue  -q sendq -w 60 -a

        Checking sendq queue status for server g_south ...

        sendq queue status for g_south as of Thu Apr 21 23:45:55 2016:     COMPLETE

        Checking sendq queue status for server g_west ...

        sendq queue status for g_west as of Thu Apr 21 23:45:55 2016:      COMPLETE

        Checking sendq queue status for server g_east ...
```

```
sendq queue status for g_east as of Thu Apr 21 23:45:55 2016:      COMPLETE
```

Now let’s review grid command propagation status:

50) [informix@east ibm]$ cdr list grid grid1 -v

```
Grid               Node               User

------------------------ ------------------------ ------------------------

grid1              g_east*            informix

                 g_south*           informix

                 g_west*            informix

Details for grid grid1


Node:g_east Stmtid:1 User:informix Database:stores 2016-04-21 23:54:24

Tag:dbaccessdemo

create database stores with log

ACK g_east 2016-04-21 23:54:24

ACK g_south 2016-04-21 23:54:25

ACK g_west 2016-04-21 23:54:26


Node:g_east Stmtid:2 User:informix Database:stores 2016-04-21 23:54:24

Tag:dbaccessdemo

CREATE PROCEDURE informix.sysdbopen()

execute procedure ifx_grid_connect('grid1', 'dbaccessdemo', 3);

END PROCEDURE;

ACK g_east 2016-04-21 23:54:24

ACK g_south 2016-04-21 23:54:25

ACK g_west 2016-04-21 23:54:26
```

Now let’s create new tables in stores database.

Review stores.sql schema :

51) [informix@east ibm]$ cat stores.sql

Now let’s execute stores.sql schema file. Note that since we do have sysdbopen procedure created, all these DDLs get replicated to all flexible grid servers.

52) [informix@east ibm]$ id

```
uid=200(informix) gid=102(informix) groups=102(informix)
```

53) [informix@east ibm]$ dbaccess stores stores.sql

```
Database selected.

Lockmode set.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Index created.

Table created.

Database closed.
```

Check control and send queue to make sure grid command propagated to all three servers.

54) [informix@east ibm]$ cdr check queue -q cntrlq -w 60 -a

```
Checking cntrlq queue status for server g_south ...
```

```
        Checking cntrlq queue status for server g_west ...

        cntrlq queue status for g_south as of Thu Apr 21 23:45:51 2016:     COMPLETE

        cntrlq queue status for g_west as of Thu Apr 21 23:45:51 2016:      COMPLETE

        Checking cntrlq queue status for server g_east ...

        cntrlq queue status for g_east as of Thu Apr 21 23:45:51 2016:      COMPLETE
```

```
55) [informix@east ibm]$ cdr check queue  -q sendq -w 60 -a

        Checking sendq queue status for server g_south ...

        sendq queue status for g_south as of Thu Apr 21 23:45:55 2016:     COMPLETE

        Checking sendq queue status for server g_west ...

        sendq queue status for g_west as of Thu Apr 21 23:45:55 2016:      COMPLETE

        Checking sendq queue status for server g_east ...

        sendq queue status for g_east as of Thu Apr 21 23:45:55 2016:      COMPLETE
```

Now verify grid command replication status:

```
56) [informix@east ibm]$ cdr list grid grid1 -v

        Grid                    Node                    User

        ------------------------ ------------------------ ------------------------

        grid1                   g_east*                 informix

                                g_south*                informix

                                g_west*                 informix

        Details for grid grid1


        Node:g_east Stmtid:1 User:informix Database:stores 2016-04-21 23:54:24

        Tag:dbaccessdemo

        create database stores with log

        ACK g_east 2016-04-21 23:54:24

        ACK g_south 2016-04-21 23:54:25
        ACK g_west 2016-04-21 23:54:26

        ---- ----

        ---- ----

        Node:g_east Stmtid:21 User:informix Database:syscdr 2016-04-21 23:59:10

        Tag:dbaccessdemo

        Define Repl G65539_1_21_catalog for stores:informix.catalog

        ACK g_east 2016-04-21 23:59:10

        ACK g_south 2016-04-21 23:59:10

        ACK g_west 2016-04-21 23:59:10
```

Now, let's load data into these tables and make sure data gets replicated to all flexible grid servers:

Review load.sh script content:

57) [informix@east ibm]$ cat load.sh

Now execute load.sh as user Informix:

58) [informix@east ibm]$ id

```
uid=200(informix) gid=102(informix) groups=102(informix)
```

59) [informix@east ibm]$ sh -x load.sh

```
+ export DEMODIR=/opt/ibm/informix/demo/dbaccess/demo_ids

+ DEMODIR=/opt/ibm/informix/demo/dbaccess/demo_ids

+ dbaccess stores -

Database selected.

Lockmode set.

28 row(s) loaded.

23 row(s) loaded.
```

```
9 row(s) loaded.

74 row(s) loaded.

67 row(s) loaded.

52 row(s) loaded.

5 row(s) loaded.

7 row(s) loaded.

Statistics updated.

Permission granted.

Database closed.
```

Check control and send queue to make sure grid command propagated to all three servers.

```
60) [informix@east ibm]$  cdr check queue  -q cntrlq -w 60 -a

        Checking cntrlq queue status for server g_south ...

        Checking cntrlq queue status for server g_west ...

        cntrlq queue status for g_south as of Thu Apr 21 23:45:51 2016:     COMPLETE

        cntrlq queue status for g_west as of Thu Apr 21 23:45:51 2016:      COMPLETE

        Checking cntrlq queue status for server g_east ...

        cntrlq queue status for g_east as of Thu Apr 21 23:45:51 2016:      COMPLETE

61) [informix@east ibm]$  cdr check queue  -q sendq -w 60 -a

        Checking sendq queue status for server g_south ...

        sendq queue status for g_south as of Thu Apr 21 23:45:55 2016:     COMPLETE

        Checking sendq queue status for server g_west ...

        sendq queue status for g_west as of Thu Apr 21 23:45:55 2016:      COMPLETE

        Checking sendq queue status for server g_east ...

        sendq queue status for g_east as of Thu Apr 21 23:45:55 2016:      COMPLETE
```

Using ‘cdr check’ command, now verify data across all three servers.

```
62) [informix@east ibm]$  cdr check replset -s grid1 -m g_east -a
```

```
Apr 22 2016 00:05:15 ------   Table scan for G65539_1_14_state start  --------

Node                 Rows     Extra   Missing  Mismatch Processed

---------------- --------- --------- --------- --------- ---------

g_east               52        0        0        0        0

g_south               52        0        0        0        0

g_west               52        0        0        0        0

--- ----

--- ----




Apr 22 2016 00:05:15 ------   Table scan for G65539_1_12_items start  --------

Node                 Rows     Extra   Missing  Mismatch Processed

---------------- --------- --------- --------- --------- ---------

g_east               67        0        0        0        0

g_south               67        0        0        0        0

g_west               67        0        0        0        0


Apr 22 2016 00:05:15 ------   Table scan for G65539_1_12_items end   ---------
Apr 22 2016 00:05:15 ------   Table scan for G65539_1_21_catalog start  --------

Node              Rows     Extra   Missing  Mismatch Processed
---------------- --------- --------- --------- --------- ---------
g_east               0        0        0        0        0
g_south               0        0        0        0        0
g_west               0        0        0        0        0

Apr 22 2016 00:05:15 ------   Table scan for G65539_1_21_catalog end   ---------
```

Now exit from east container:

Note that you will have to run exit command twice to exit from Informix and root shell.

```
62) [informix@east ibm]$ exit
exit
63) [root@east ibm]# exit
Exit
```

Connection manager configuration:

Now let us setup connection manager to monitor all three clusters and flexible grid.

Login to cm1 container:

```
64) $ ./login.sh cm1
--shell
65) [root@cm1 ibm]# id
```

```
uid=0(root) gid=0(root) groups=0(root)
```

Review /opt/ibm/informix/etc/cmsm_demo.cfg configuration file:

66) [root@cm1 ibm]# cat /opt/ibm/informix/etc/cmsm_demo.cfg

```
##############################################################################

GRID grid1

{

  INFORMIXSERVER       g_east,g_west,g_south

  SLA grid_oltp1       DBSERVERS=ANY POLICY=LATENCY

  SLA grid_oltp2       DBSERVERS=ANY POLICY=LATENCY

}
```

Review connection manager sqlhosts file. This file has connectivity details for each server, and for SLA definitions for client connections:

67) [root@cm1 ibm]# cat $INFORMIXSQLHOSTS

```
g_east group - - i=1

east onsoctcp 172.20.0.10 60000 g=g_east

east_dr onsoctcp 172.20.0.11 60000 g=g_east

g_west group - - i=2

west onsoctcp 172.20.0.12 60000 g=g_west

west_dr onsoctcp 172.20.0.13 60000 g=g_west

g_south group - - i=3

south onsoctcp 172.20.0.14 60000 g=g_south

south_dr onsoctcp 172.20.0.15 60000 g=g_south
```

```
oltp_w onsoctcp 172.20.0.16 50000

report_w onsoctcp 172.20.0.16 50001

oltp_e onsoctcp 172.20.0.16 50002

report_e onsoctcp 172.20.0.16 50003

oltp_s onsoctcp 172.20.0.16 50004

report_s onsoctcp 172.20.0.16 50005

grid_oltp1 onsoctcp 172.20.0.16 50006

grid_oltp2 onsoctcp 172.20.0.16 50007
```

Now, let’s review start_cm.sh script. It has command to start connection manager.

68) [root@cm1 ibm]# cat start_cm.sh

```
su informix -c "/opt/ibm/informix/bin/oncmsm -c /opt/ibm/informix/etc/cmsm_demo.

cfg"
```

Let’s start connection manager:

69) [root@cm1 ibm]# sh -x start_cm.sh

```
+ su informix -c '/opt/ibm/informix/bin/oncmsm -c /opt/ibm/informix/etc/cmsm_dem

o.cfg'

Connection Manager started successfully

Please check IBM Informix Connection Manager log file: /opt/ibm/informix/tmp/cm1

.log
```

Verify connection redirection using grid related connection manager SLA definitions:

Now connect to each of the SLA definition, and verify client redirection logic.

First, let's change user to Informix:

```
70) [root@cm1 ibm]# su informix

71) [informix@cm1 ibm]$ id

        uid=200(informix) gid=102(informix) groups=102(informix)
```

"sla_connect" executable allows us to connect to each of the SLA definition and prints out which server it did connect to. SLA definition needs to be specified using –s option.

```
72)  [informix@cm1 ibm]$ sla_connect -s grid_oltp1

        ===== sla_connect =====

        Client's SLA  : grid_oltp1

        Client Redirected to: [east]  🡐== this server could be east,west or south

        ==========================

73) [informix@cm1 ibm]$ sla_connect -s grid_oltp2

        ===== sla_connect =====

        Client's SLA  : grid_oltp2

        Client Redirected to: [east] 🡐== this server could be east,west or south


        ==========================
```

Start clientV1.sh application from the connection manager container

As user Informix, Run clientV1.sh with connection to grid_oltp1

```
74A) [root@east ibm]# su informix

[informix@east ibm]$ id

        uid=200(informix) gid=102(informix) groups=102(informix)


74B) [informix@cm1 ibm]# ./clientV1.sh grid_oltp1
```

This script run forever till you hit Control-c.

Leave this window running.

Upgrade schema for east region servers:

From new terminal window, login to cm1 host:

75) $ ./login.sh cm1

--shell

76) [root@cm1 ibm]# id

uid=0(root) gid=0(root) groups=0(root)

Update grid_oltp1 SLA definition to remove g_east server and reload connection manager configuration file:

77) [root@cm1 ibm]# upd_sla.sh grid_oltp1 g_west,g_south

Note: This prevents clinetV1.sh from connecting to g_east server.

Exit from cm1 container:

78) [informix@cm1 ibm]$ exit

exit

Force single user mode for server g_east to kill existing sessions

Login to east container:

79) $ ./login.sh east

First, let's change user to Informix:

80) [root@east ibm]# su Informix

81) [informix@cm1 ibm]$ id

uid=200(informix) gid=102(informix) groups=102(informix)

82) [informix@east ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line -- Up 02:47:10 -- 16446

8 Kbytes
```

Force single user mode

83) [informix@cm1 ibm]$ onmode -jy

84) [informix@east ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- Single-User (Prim) -- Up
2 days 21:27:02 -- 180852 Kbytes
```

Change server mode to On-Line

85) [informix@cm1 ibm]$ onmode -m

86) [informix@east ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line(Prim) -- Up 02:47:10 -- 16446

8 Kbytes
```

Note: From clientV1.sh application window, check and make sure new connections only go to g_west and g_south servers

From the same east container window, disconnect ER network connection between g_east and g_west, and also between g_east and g_south

87) $ cdr disconnect server -c g_east g_west

88) $ cdr disconnect server -c g_east g_south

Verify connection status:

89) $ cdr list server -c g_east

```
SERVER                 ID STATE    STATUS     QUEUE  CONNECTION CHANGED
-----------------------------------------------------------------------
g_east                  1 Active   Local          0
g_south                 3 Active   Disconnect     0 Apr  7 21:42:04
```

```
g_west                 2 Active   Disconnect      0 Apr  7 21:41:56
```

Review schema changes in file `schema_upgrade.sql`

90) $ `cat schema_upgrade.sql`

```
-- Set grid context for the client,
-- required to replicate DDLs and update replicate definitions
--execute procedure ifx_grid_connect('grid1', 'schema_upgrade', 1);

--Add email and twitter id columns to customer table
alter table customer add email char(100);
alter table customer add twitter char(100);

-- Increase customer first and last name length from 15 to 128 characters.
alter table customer modify fname char(128);
alter table customer modify lname char(128);

-- Create new coupons table for promotion offers.
create table coupons (coupon_code int primary key,
                      coupon_desc char(512),
                      discount int,
                      start_date date,
                      end_date date) with crcols lock mode row;
```

Upgrade schema:

91) $ dbaccess stores `schema_upgrade.sql`

```
Database selected.


Table altered.


Table altered.


Table altered.


Table altered.


Table created.


Database closed.
```

Note: The above command auto registers replicate definition on new coupons table and modify replicate definition on customer table to include new columns email and twitter, also change column length for fname and lname.

Load data into new coupons table and update customer table and populate data for new columns.

Review load2.sh script:

92) [informix@east ibm]$ cat load2.sh

```
dbaccess stores - <<!
set lock mode to wait;
--Load data into coupons table
load from coupons.unl insert into coupons;

-- Update Frank and Chris email and twitter ids.
update customer set (email, twitter) = ("Frank_Lessor@gmail.com", "@Frank_Lessor") where
customer_num=128;
update customer set (email, twitter) = ("Chris_Putnum@gmail.com", "@Chris_Putnum") where
customer_num=124;

-- Add new customer records
insert into customer (customer_num, fname, lname, company, address1, address2, city, state,
zipcode, phone, email, twitter) values(130, "Sam", "Hill", "IBM", "11200 Lakeview ave", "",
"Lenexa", "KS", 66219, "913-222-444", "sam_hill@us.ibm.com", "@sam_hill");
insert into customer (customer_num, fname, lname, company, address1, address2, city, state,
zipcode, phone, email, twitter) values(131, "Alex", "Smith", "Chiefs", "1 Arrowhead Dr",
"", "Kansas city", "MO", 64129, "816-232-474", "alex_smith@us.ibm.com", "@alex_smith");
```

Execute load2.sh script:

93) [informix@east ibm]$ sh -x load2.sh

```
 + dbaccess stores -

Database selected.


Lockmode set.


3 row(s) loaded.


1 row(s) updated.


1 row(s) updated.


1 row(s) inserted.


1 row(s) inserted.



Database closed.
```

Note that schema changes, and newly changed data is staged in sendq as we already disconnected network connection between g_east server and g_west and g_south servers.

Verify and make sure replicate definition is created on new coupons table

94) $ cdr list repl |grep coupons

```
REPLICATE:        G65541_1_32_coupons
PARTICIPANT:      stores:informix.coupons
```

Now logout from east container. You will have to run ‘exit’ command twice to exit from east container.

```
95) [informix@east ibm]$ exit

exit

96) [root@east ibm]# exit

Exit

$
```

Update grid_oltp2 SLA definition to redirect updated client application(clientV2.sh) to g_east server.

Login to cm1 container.

```
97) $ ./login.sh cm1

--shell

98) [root@cm1 ibm]# id

        uid=0(root) gid=0(root) groups=0(root)
```

Update grid_oltp2 SLA definition to only redirect upgraded client(clientV2.sh) connections to g_east server, and reload connection manager configuration file:

```
99) [root@cm1 ibm]# upd_sla.sh grid_oltp2 g_east
```

Start clientV2.sh server and make sure that clientV2.sh connections to g_east server.

Start clientV2.sh application

First, let’s change user to Informix:

```
100) [root@cm1 ibm]# su informix

[informix@cm1 ibm]$ id
```

```
uid=200(informix) gid=102(informix) groups=102(informix)
```

Run clientV2.sh with connection to grid_oltp2

101) [informix@cm1 ibm]# ./clientV2.sh grid_oltp2

This script run forever till you hit Control-c.

Leave this window running.

Verify the output from clientV2.sh and make sure that new client is only connecting to east server.

```
set lock mode to wait;
Lockmode set.


select dbservername from sysmaster:sysdual;


(expression)  east

1 row(s) retrieved.


update customer set email=email where customer_num=120;
1 row(s) updated.


update coupons set discount = discount + 5 where coupon_code=3;
0 row(s) updated.


update coupons set discount = discount - 5 where coupon_code=3;
0 row(s) updated.




Database closed.

Loop count 37, enter 'Control-C' to stop script
```

Upgrade schema for west region servers:

From new terminal window, login to cm1 host:

```
102) $ ./login.sh cm1
```

--shell

```
103) [root@cm1 ibm]# id

        uid=0(root) gid=0(root) groups=0(root)
```

Update grid_oltp1 SLA definition to remove g_west server and reload connection manager configuration file:

```
104) [root@cm1 ibm]# upd_sla.sh grid_oltp1 g_south
```

Note: This prevents clinetV1.sh from connecting to g_west server.

Exit from cm1 container:

```
105) [informix@cm1 ibm]$ exit

exit
```

Force single user mode for server g_west to kill existing sessions

Login to west container:

```
106) $ ./login.sh west
```

First, let's change user to Informix:

```
107) [root@west ibm]# su informix

108) [informix@west ibm]$ onstat -

        IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line (Prim) -- Up 02:47:10 -- 16446

        8 Kbytes
```

Force single user mode

```
109) [informix@west ibm]$ onmode -jy

110) [informix@west ibm]$ onstat -
```

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- Single-User (Prim) -- Up
2 days 21:27:02 -- 180852 Kbytes
```

Change server mode to On-Line

111) [informix@west ibm]$ onmode -m

112) [informix@west ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line(Prim) -- Up 02:47:10 -- 16446

8 Kbytes
```

Note: From clientV1.sh application window, check and make sure new connections only go to **g_south** server.

From the same west container window, disconnect ER network connection between g_west and g_souths

113) $ cdr disconnect server -c g_west g_south

Verify connection status:

114) $ cdr list server -c g_west

```
SERVER                 ID STATE    STATUS     QUEUE  CONNECTION CHANGED
-----------------------------------------------------------------------
g_east                  1 Active   Disconnect  93804 Apr  7 21:41:56
g_south                 3 Active   Disconnect    591 Apr  7 21:59:56
g_west                  2 Active   Local           0
```

Note: g_east server should be already in disconnected state.

Now, reconnect connect between g_east and g_west server to replicate schema changes from east and west server.

115) $ cdr connect server -c g_west g_east

Verify connection status:

```
116) [informix@west ibm]$ cdr list serv
     SERVER                 ID STATE    STATUS     QUEUE  CONNECTION CHANGED
```

```
      ----------------------------------------------------------------------
      g_east                  1 Active   Connected  125425 Apr  7 22:12:38
      g_south                 3 Active   Disconnect  33788 Apr  7 21:59:56
      g_west                  2 Active   Local           0
```

Review replicated schema changes:

First check grid command status:

```
117) [informix@west ibm]$ cdr list grid -v
      Grid                    Node                    User
      ------------------------ ------------------------ ------------------------
      grid1                   g_east*                 informix
                              g_south*                informix
                              g_west*                 informix
      Details for grid grid1
      …. ….

      …. ….

      Node:g_east Stmtid:31 User:informix Database:stores 2017-04-07 21:43:11
      Tag:dbaccessdemo
      create table coupons (coupon_code int primary key,
                            coupon_desc char(512),
                            discount int,
                            start_date date,
                            end_date date) with crcols lock mode row
      ACK g_east 2017-04-07 21:43:11
      ACK g_west 2017-04-07 22:12:39
      PENDING g_south

      Node:g_east Stmtid:32 User:informix Database:syscdr 2017-04-07 21:43:11
      Tag:dbaccessdemo
      Define Repl G65541_1_32_coupons for stores:informix.coupons
      ACK g_east 2017-04-07 21:43:11
      ACK g_west 2017-04-07 22:12:39
      PENDING g_south
```

Verify and make sure replicate definition is created on new coupons table

```
118) $ cdr list repl |grep coupons

      REPLICATE:        G65541_1_32_coupons
      PARTICIPANT:      stores:informix.coupons
```

Verify new schema using dbschema:

```
119) $ dbschema -t coupons -d stores
```

```
create table "informix".coupons
  (
    coupon_code integer,
    coupon_desc char(512),
    discount integer,
    start_date date,
    end_date date,
    primary key (coupon_code)
  );

revoke all on "informix".coupons from "public" as "informix";


create unique index "informix".erkey_42 on "informix".coupons
    (ifx_erkey_1,ifx_erkey_2,ifx_erkey_3) using btree ;
```

120) $ `dbschema -t customer -d stores`

```
{ TABLE "informix".customer row size = 578 number of columns = 12 index size = 34 }

create table "informix".customer
  (
    customer_num serial not null ,
    fname char(128),
    lname char(128),
    company char(20),
    address1 char(20),
    address2 char(20),
    city char(15),
    state char(2),
    zipcode char(5),
    phone char(18),
    email char(100),
    twitter char(100),
    primary key (customer_num)
  );

revoke all on "informix".customer from "public" as "informix";


create unique index "informix".erkey_4 on "informix".customer
    (ifx_erkey_1,ifx_erkey_2,ifx_erkey_3) using btree ;
create index "informix".zip_ix on "informix".customer (zipcode)
    using btree ;
```

Now logout from east container. You will have to run ‘exit’ command twice to exit from east container.

121) [informix@east ibm]$ exit

exit

122) [root@east ibm]# exit

Exit

$

Update grid_oltp2 SLA definition to redirect updated client application(clientV2.sh) to both g_east and g_west servers.

Login to cm1 container.

```
123) $ ./login.sh cm1
```

--shell

```
124) [root@cm1 ibm]# id
        uid=0(root) gid=0(root) groups=0(root)
```

Update grid_oltp2 SLA definition to only redirect upgraded client(clientV2.sh) connections to g_east server, and reload connection manager configuration file:

```
125) [root@cm1 ibm]# upd_sla.sh grid_oltp2 g_east,g_west
```

Verify the output from clientV2.sh and make sure that new client is now connecting to both east and west servers.

Upgrade schema for South region servers

Stop clientV1.sh application using Control-c command.

From the previous terminal, exit from the cm1 container:

```
126) [root@cm1 ibm]# id
     uid=0(root) gid=0(root) groups=0(root)

127) [root@cm1 ibm]# exit
     exit
```

Force single user mode for server g_south to kill existing sessions

Login to east container:

128) $ ./login.sh south

First, let's change user to Informix:

129) [root@ south ibm]# su informix

130) [informix@ south ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line (Prim) -- Up 02:47:10 -- 16446

8 Kbytes
```

Force single user mode

131) [informix@ south ibm]$ onmode -jy

132) [informix@ south ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- Single-User (Prim) -- Up
2 days 21:27:02 -- 180852 Kbytes
```

Change server mode to On-Line

133) [informix@south ibm]$ onmode -m

134) [informix@ south ibm]$ onstat -

```
IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line(Prim) -- Up 02:47:10 -- 16446

8 Kbytes
```

Verify Enterprise Replication network status:

```
135) [informix@south ibm]$ cdr list server –c g_south
```

```
SERVER                    ID STATE    STATUS     QUEUE  CONNECTION CHANGED
-----------------------------------------------------------------------------
g_east                     1 Active   Disconnect 350200 Apr  7 21:42:04
g_south                    3 Active   Local          0
g_west                     2 Active   Disconnect 254223 Apr  7 21:59:56
```

Note: Both east and west servers should show "Disconnect" status.

From the same west container window, disconnect ER network connection between g_west and g_souths

Redundant command, not needed ->136) $ cdr disconnect server -c g_west g_south

Verify connection status:

Now, reconnect network connect between g_south and g_east servers to replicate schema changes from east to south server.

137) $ cdr connect server -c g_south g_east

Verify connection status:

```
138) [informix@south ibm]$ cdr list serv
     SERVER                 ID STATE    STATUS     QUEUE  CONNECTION CHANGED
     -----------------------------------------------------------------------
     g_east                  1 Active   Connected  248175 Apr  7 22:43:04
     g_south                 3 Active   Local           0
     g_west                  2 Active   Disconnect 254873 Apr  7 21:59:56
```

Review replicated schema changes:

First check grid command status:

```
139) [informix@west ibm]$ cdr list grid -v
     Grid                     Node                     User
     ------------------------ ------------------------ ------------------------
     grid1                    g_east*                  informix
                              g_south*                 informix
                              g_west*                  informix
     Details for grid grid1
     …. ….

     …. ….

     Node:g_east Stmtid:31 User:informix Database:stores 2017-04-07 21:43:11
     Tag:dbaccessdemo
     create table coupons (coupon_code int primary key,
                           coupon_desc char(512),
                           discount int,
                           start_date date,
                           end_date date) with crcols lock mode row
     ACK g_east 2017-04-07 21:43:11
     ACK g_south 2017-04-07 22:43:05
     PENDING g_west

     Node:g_east Stmtid:32 User:informix Database:syscdr 2017-04-07 21:43:11
     Tag:dbaccessdemo
     Define Repl G65541_1_32_coupons for stores:informix.coupons
     ACK g_east 2017-04-07 21:43:11
     ACK g_south 2017-04-07 22:43:05
     PENDING g_west
```

Verify and make sure replicate definition is created on new coupons table

140) $ cdr list repl |grep coupons

```
REPLICATE:        G65541_1_32_coupons
PARTICIPANT:      stores:informix.coupons
```

Verify new schema using dbschema:

141) $ dbschema -t coupons -d stores

```
create table "informix".coupons
  (
    coupon_code integer,
    coupon_desc char(512),
    discount integer,
    start_date date,
    end_date date,
    primary key (coupon_code)
  );

revoke all on "informix".coupons from "public" as "informix";


create unique index "informix".erkey_42 on "informix".coupons
    (ifx_erkey_1,ifx_erkey_2,ifx_erkey_3) using btree ;
```

142) $ dbschema -t customer -d stores

```
{ TABLE "informix".customer row size = 578 number of columns = 12 index size = 34 }

create table "informix".customer
  (
    customer_num serial not null ,
    fname char(128),
    lname char(128),
    company char(20),
    address1 char(20),
    address2 char(20),
    city char(15),
    state char(2),
    zipcode char(5),
    phone char(18),
    email char(100),
    twitter char(100),
    primary key (customer_num)
  );
```

```
revoke all on "informix".customer from "public" as "informix";


create unique index "informix".erkey_4 on "informix".customer
    (ifx_erkey_1,ifx_erkey_2,ifx_erkey_3) using btree ;
create index "informix".zip_ix on "informix".customer (zipcode)
    using btree ;
```

Now, reconnect network connect between g_south and g_west servers to replicate data from west to south server.

143) $ cdr connect server -c g_south g_west

Verify connection status:

```
144) [informix@south ibm]$ cdr list server -c g_south

     SERVER                   ID STATE    STATUS     QUEUE  CONNECTION CHANGED
     -----------------------------------------------------------------------
     g_east                    1 Active   Connected       0 Apr  7 22:43:04
     g_south                   3 Active   Local           0
     g_west                    2 Active   Connected  226111 Apr  7 22:46:05
```

Check control and send queue to make sure data is being replicated.

145) [informix@south ibm]$ cdr check queue -q cntrlq -w 60 -a

```
     Checking cntrlq queue status for server g_south ...
     cntrlq queue status for g_south as of Fri Apr  7 22:47:55 2017:      COMPLETE
     Checking cntrlq queue status for server g_west ...
     cntrlq queue status for g_west as of Fri Apr  7 22:47:55 2017:       COMPLETE
     Checking cntrlq queue status for server g_east ...
     cntrlq queue status for g_east as of Fri Apr  7 22:47:55 2017:       COMPLETE
```

```
146) [informix@south ibm]$ cdr check queue  -q sendq -w 60 -a
     Checking sendq queue status for server g_south ...
     sendq queue status for g_south as of Fri Apr  7 22:49:01 2017:      COMPLETE
     Checking sendq queue status for server g_west ...
     sendq queue status for g_west as of Fri Apr  7 22:49:01 2017:       COMPLETE
     Checking sendq queue status for server g_east ...
     sendq queue status for g_east as of Fri Apr  7 22:49:01 2017:       COMPLETE
```

Now logout from south container. You will have to run ‘exit’ command twice to exit from east container.

147) [informix@south ibm]$ exit

exit

148) [root@south ibm]# exit

Exit

$

Update grid_oltp2 SLA definition to redirect updated client application(clientV2.sh) to all threeservers: g_east , g_west and g_south servers.

Login to cm1 container.

149) $ ./login.sh cm1

--shell

150) [root@cm1 ibm]# id

uid=0(root) gid=0(root) groups=0(root)

Update grid_oltp2 SLA definition to only redirect upgraded client(clientV2.sh) connections to g_east server, and reload connection manager configuration file:

151) [root@cm1 ibm]# upd_sla.sh grid_oltp2 g_east,g_west,south

Verify the output from clientV2.sh and make sure that new client is now connecting to both east and west servers.

Verify data across all servers

152) Stop clientV2.sh application using Control-c command and logout from cm1 container:

```
Loop count 3618, enter 'Control-C' to stop script
^C

153) [informix@cm1 ibm]$ exit
exit
```

```
154) [root@cm1 ibm]# exit
exit
```

Login to east container:

155) $ ./login.sh east

First, let’s change user to Informix:

156) [root@east ibm]# su Informix

157) [informix@cm1 ibm]$ id

uid=200(informix) gid=102(informix) groups=102(informix)

```
158) [informix@east ibm]$ onstat -

    IBM Informix Dynamic Server Version 12.10.FC8W1 -- On-Line (Prim) -- Up 2
    days 22:43:33 -- 189044 Kbytes
```

Verify data across all three servers:

```
159) [informix@east ibm]$ cdr check replset -s grid1 -m g_east -a


Apr 07 2017 22:56:42 ------   Table scan for G65541_1_14_state start  --------

Node                  Rows     Extra   Missing  Mismatch Processed
---------------- --------- --------- --------- --------- ---------
g_east                  52         0         0         0         0
g_south                 52         0         0         0         0
g_west                  52         0         0         0         0

Apr 07 2017 22:56:42 ------   Table scan for G65541_1_14_state end   ---------

… ….


… ….

Apr 07 2017 22:56:44 ------   Table scan for G65541_1_12_items start  --------

Node                  Rows     Extra   Missing  Mismatch Processed
---------------- --------- --------- --------- --------- ---------
g_east                  67         0         0         0         0
g_south                 67         0         0         0         0
g_west                  67         0         0         0         0

Apr 07 2017 22:56:44 ------   Table scan for G65541_1_12_items end   ---------


Apr 07 2017 22:56:44 ------   Table scan for G65541_1_21_catalog start  --------

Node                  Rows     Extra   Missing  Mismatch Processed
---------------- --------- --------- --------- --------- ---------
g_east                   0         0         0         0         0
g_south                  0         0         0         0         0
g_west                   0         0         0         0         0
```

```
Apr 07 2017 22:56:44 ------   Table scan for G65541_1_21_catalog end   ---------
```

Congratulations for successfully completing rolling schema upgrade lab!!