# Implementation and Configuration of Oracle Connection manager

**Document history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Name** | **Comment** |
| 22.04.2020 | 01.00 | Thorbjørn Johnsøn | Created |
| 20-05-2020 | 01.00 | Flemming Grothe Jensen | Reviewed |
| 22-11-2021 | 01.10 | Thorbjørn Johnsøn | New servers for nonproduction |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Title | Oracle Connection Manager | 22 April 2020 | Date |
| Version | 1.0 | 1 () | Page |
| Author | Thorbjørn Johnsøn | Confidential |  |
|  |  | - | Reference |
| Subject | Implementation and Configuration of Oracle Connection Manager | Final |  |
| Department |  |  |  |
| Project | Oracle Platform |  |  |

# Introduction

The database connection requirements from unsecure zones like DMZ and VPN are increasing. The firewall are SQL\*Net aware and historically these situations has been handled using opening the listener port (typically 1521) between the database hosts and the client residing in the DMZ.

Today, due to cost savings effort, the database servers typically hosts a huge number of databases. An Exadata setup can handle more than 50 databases. Since the firewall only knows about a port we do have a less secure environment when we’re using the firewall to secure the databases. Every database on the database server will be accessible from the DMZ/VPN. Naturally the database session still has to be authenticated but one of the security layer has been breached for more databases than we would like it to be.,

What is required, is a database aware security component. Oracle connection manager can be used for this specific tasks. Rules in Connection Manager can be setup to ensure that only specific clients can connect to specific database hosts and accessing a specific database service.

# Implementation



The figure shows a possible setup for a production/preproduction setup in Nordea. In this setup we have two sites with RAC enabled on both sites and a Dataguard configured between the two sites. The database clusters contains 15 databases. It is illustrated that the two databases (the yellow and the blue) needs to be accessed from the DMZ. The other 13 databases do not.

The connection managers is installed on several virtual machines running Linux to avoid single point of failure (4 in this example).

## Oracle server implications

The database server is not aware of the Connection Manager and the Connection Manager cannot be controlled by Oracle Clusterware. The positive side of this is that we do not need to alter anything on the database server/cluster. The listener configuration is exactly the same as if the Connection Manager was not part of the setup.

If the connection manager does not accept the connection due to a missing rule the Oracle server simply does not get any requests. It is the Connection Manager that breaks the connection and it is only in the Connection Manager logs any error message can be found. It is a good idea to monitor that log to ensure that malicious attempt are caught.

## Oracle client implications

The clients connection information needs to be modified to enable the connection manager. The TNS entry for the client needs to have a route enabled. That route requires two descriptions. Description 1 for the connection manager and description 2 for the actual database connections. Naturally this can and will be involved to ensure switchover/failover on the database site still works.

A very simple connection could look like:

dmz\_dbct=

(description=

(source\_route=yes)

(address\_list=

(address=(protocol=tcp)

(host=10.53.13.76)

(port=1950))

(address=(protocol=tcp)

(host=ora-dbct-a)

(port=1521))

)

(connect\_data=(service\_name=dbct)))

In this case the connection manager is located on host 10.53.13.76. Connection Manager port is 1950. The address for the database host/cluster is forwarded to the connection manager and will be DNS resolved on the Connection Manager (located in the blue zone).

In the simple example we have introduced a single point of failure. That is adequate for Standard applications. HA/BCA applications needs a more sophisticated tns entry.

## Firewall details

The firewall needs to be opened between the client and the Connection Manager server(s). There is no need to open any ports between the client and the database cluster.

In the above example a firewall opening for port 1950 is needed between the client in the DMZ and the Connection manager (10.53.13.76). Aside from the name of the host and the name of the database service the database is not known to the client in the DMZ.

## Connection Manager instance

Any session connected remotely to a database needs to connect to a listener. Connection Manager needs to be started and needs to listen on the requested port (1950). Connection manager can then forward the connection to the database server. If the connection manager is not started the session cannot be forwarded to the database server and the client will not be able to connect to the database.

All configuration regarding the Connection Manager is maintained in a text file named cman.ora located in ${ORACLE\_HOME}/network/admin.

Connection Manager will automatically be started during the boot of the linux machine. This is implemented using systemctl.

## Connection Manager Rules

The connection manager contains rule(s) to ensure that only specific clients can connect to specific hosts (or clusters) on a specific database service. After the client connection has penetrated the firewall (port 1950 in our example) and the Connection Manager accepts the connection, the next line of defence is the connection manager rule.

The rule will validate the connection using the client address, database host/cluster name and database service name. All this information must be available to the client in the DMZ to request to connect to the database. Two of these parameters are part of the connect string. The client address is not part of the tns entry and that makes it impossible for an intruder to copy the connect string to another client. Connects from another computer will fail due to the lack of a rule. An exception of this is the possibility to use wildcards in the rule (src=\*).

A rule will look like:

(rule=

(src=xxx.xxx.xxx.xxx)

(dst=ora-dbct-a)(srv=dbct)(act=accept)

(action\_list=(aut=off)(moct=0)(mct=0)(mit=0)(conn\_stats=on))

)

After the rule has been added it is necessary to reload the configuration. This can be done on the Connection Manager using the cmctl tool.

# Installing the Connection Manager

Software is preinstalled on all Connection Managers. Basically it is a client custom setup including the Oracle Connection Manager utility.

On all connection manager the Oracle home (client\_1) is located in the catalog:

/u01/app/oracle/product/12.2.0.1/client\_1

The configuration file for the connection manager is located in the catalog:

$ORACLE\_HOME/network/admin/cman.ora

The connection manager configuration is named cman\_is-<hostname>

Every connection manager will autostart the connection manager during reboot using a systemctl service.

# Grant access to a database from DMZ/VPN

The process is relative simple.

* Customer request access to a database service

Customer needs to provide:

Client IP (if possible)  
Environment (Sandbox, Integration, Preprod, Production)  
Database service

* Oracle team configures a rule in the Connection Manager(s) in the requested environment. This rule should be present in all connection managers in the requested environments.
* Oracle team informs the customer what firewall opening(s) are needed between the client IP and the connection managers in the particular environment.
* Oracle team sends a TNS entry to the customer.
* Customer requires the firewall opening and modify the TNS entry on the client in the DMZ/VPN.

# Automatically restart of Connection Manager

The Connection Managers are installed on Linux servers maintained by the Linux Operation Team. The Linux servers are typically patched each month and that will give an outage. If the TNS entry is created correctly this will not be an issue for the database service (unless all servers are maintained in the same period).

During the Linux patch the server is typically booted. Connection Manager will automatically startup after a boot usin a systemctl enabled service.

# Server list

This chapter describes the Configuration Managers installed in the different environments.

## Integration/Sandbox Environments

If the request for access is concerning a database in the Integration/Sandbox the following applies:

**Connection Managers:**

Hostname: ap-isoracm1t.oneadr.net

IP address: 10.96.243.113

Port: 1950

Connection Manager: cman-isoracm1t

Hostname: ap-isoracm2t.oneadr.net

IP address: 10.96.136.20

Port: 1950

Connection Manager: cman-isoracm2t

**Old Test Configurations (to be maintained but no new configurations):**

Hostname: is-oracm1t.oneadr.net

IP address: 10.53.13.75

Port: 1950

Connection Manager: cman\_is-oracm1t

Hostname: is-oracm2t.oneadr.net

IP address: 10.53.13.76

Port: 1950

Connection Manager: cman\_is-oracm2t

## Preproduction Environments

If the request for access is concerning a database in the Preproduction environment the following applies:  
  
**Connection Managers:**

Hostname: ap-isoracm1d.oneadr.net

IP address: 10.96.127.105

Port: 1950

Connection Manager: cman\_is-oracm1d

Hostname: ap-isoracm2d.oneadr.net

IP address: 10.96.127.104

Port: 1950

Connection Manager: cman\_is-oracm2d

Hostname: ap-isoracm3d.oneadr.net

IP address: 10.96.127.106

Port: 1950

Connection Manager: cman\_is-oracm3d

Hostname: ap-isoracm4d.oneadr.net

IP address: 10.96.127.68

Port: 1950

Connection Manager: cman\_is-oracm4d

**Old Preprod Configurations (to be maintained but no new configurations):**

Hostname: is-oracm1d.oneadr.net

IP address: 10.53.13.73

Port: 1950

Connection Manager: cman\_is-oracm1d

Hostname: is-oracm2d.oneadr.net

IP address: 10.53.17.112

Port: 1950

Connection Manager: cman\_is-oracm2d

Hostname: is-oracm3d.oneadr.net

IP address: 10.53.17.113

Port: 1950

Connection Manager: cman\_is-oracm3d

Hostname: is-oracm4d.oneadr.net

IP address: 10.53.17.114

Port: 1950

Connection Manager: cman\_is-oracm4d

## Production Environments

If the request for access is concerning a database in the Production environment the following applies:  
  
**Connection Managers:**

Hostname: is-oracm1p.oneadr.net

IP address: 10.104.127.58

Port: 1630

Connection Manager: cman\_is-oracm1p

Hostname: is-oracm2p.oneadr.net

IP address: 10.104.127.57

Port: 1630

Connection Manager: cman\_is-oracm2p

Hostname: is-oracm3p.oneadr.net

IP address: 10.104.127.59

Port: 1630

Connection Manager: cman\_is-oracm3p

Hostname: is-oracm4p.oneadr.net

IP address: 10.104.127.56

Port: 1630

Connection Manager: cman\_is-oracm4p

# Appendix A: Cman.ora

cman\_is-oracm1t =

(configuration=

(address=(protocol=tcp)(host=is-oracm1t.oneadr.net)(port=1950))

(parameter\_list =

(aso\_authentication\_filter=off)

(connection\_statistics=yes)

(log\_level=user)

(max\_connections=256)

(idle\_timeout=600)

(inbound\_connect\_timeout=0)

(session\_timeout=0)

(outbound\_connect\_timeout=0)

(max\_gateway\_processes=16)

(min\_gateway\_processes=2)

(remote\_admin=on)

(trace\_filelen=1000)

(trace\_level=off)

(max\_cmctl\_sessions=4)

(event\_group=init\_and\_term,memory\_ops)

)

(rule\_list=

(rule=

(src=is-oracm1t.oneadr.net)(dst=127.0.0.1)(srv=cmon)(act=accept)

(action\_list=(aut=off)(moct=0)(mct=0)(mit=0)(conn\_stats=on))

)

#

# Arc

#

(rule=

(src=\*)(dst=ora-arc64201s-a.qaoneadr.local)(srv=arc64201s)(act=accept)

(action\_list=(aut=off)(moct=0)(mct=0)(mit=0)(conn\_stats=on))

)

(rule=

(src=\*)(dst=ora-arc64201s-b.qaoneadr.local)(srv=arc64201s)(act=accept)

(action\_list=(aut=off)(moct=0)(mct=0)(mit=0)(conn\_stats=on))

)

)

)

**Appendix B: TNS Example:**

(DESCRIPTION\_LIST =

(LOAD\_BALANCE=OFF)(FAILOVER=ON)

(CONNECT\_TIMEOUT=5000)(TRANSPORT\_CONNECT\_TIMEOUT=3000)(RETRY\_COUNT=3)

(DESCRIPTION=(SOURCE\_ROUTE=YES)

(ADDRESS=(PROTOCOL=TCP)(HOST=10.53.13.75)(PORT=1950))

(ADDRESS=(PROTOCOL=TCP)

(HOST=ora-arc64201t-a.qaoneadr.local)(PORT=1521))

(CONNECT\_DATA=(SERVER=DEDICATED)

(SERVICE\_NAME=arc64201t)

(FAILOVER\_MODE = (TYPE = SELECT)(METHOD = BASIC))))

(DESCRIPTION=(SOURCE\_ROUTE=YES)

(ADDRESS=(PROTOCOL=TCP)(HOST=10.53.13.76)(PORT=1950))

(ADDRESS=(PROTOCOL=TCP)

(HOST=ora-arc64201t-b.qaoneadr.local)(PORT=1521))

(CONNECT\_DATA=(SERVER=DEDICATED)(SERVICE\_NAME=arc64201t)

(FAILOVER\_MODE = (TYPE = SELECT)(METHOD = BASIC))))

)

**Appendix C: cman.service**

[Unit]

Description=Cman Service

Wants=network.target

After=network.target

[Service]

User=oracle

Group=oinstall

LimitNOFILE=10240

MemoryLimit=8G

RestartSec=30s

StartLimitInterval=1800s

StartLimitBurst=20

ExecStart=/u01/app/oracle/scripts/cman\_service.sh -c cman\_is-oracm1t -a start -o /u01/app/oracle/product/12.2.0.1/client\_1

ExecReload=/u01/app/oracle/scripts/cman\_service.sh -c cman\_is-oracm1t -a reload -o /u01/app/oracle/product/12.2.0.1/client\_1

ExecStop=/u01/app/oracle/scripts/cman\_service.sh -c cman\_is-oracm1t -a stop -o /u01/app/oracle/product/12.2.0.1/client\_1

KillMode=control-group

Restart=on-failure

Type=forking

[Install]

WantedBy=multi-user.target

Alias=cman.service