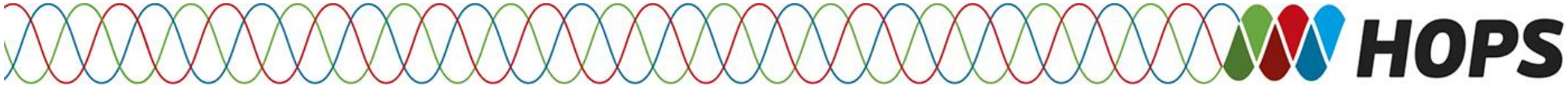
Oracle Unbreakable Cluster v. 12c

out of the box cluster – for free



Active/Passive configuration for single instance database using SCAN

- Cluster
  - Active/Passive configuration |Failover
  - Oracle Single Client Access Name (SCAN) (without a floating VIP)

- Aliases:
  - **Using Oracle Clusterware to Protect A Single Instance Oracle Database**
  - Oracle Database Failover Cluster with Grid Infrastructure
  - Protect single instance database with Grid Infrastructure

About company

- CTSO –
  Croatian Transmission System Operator Ltd.

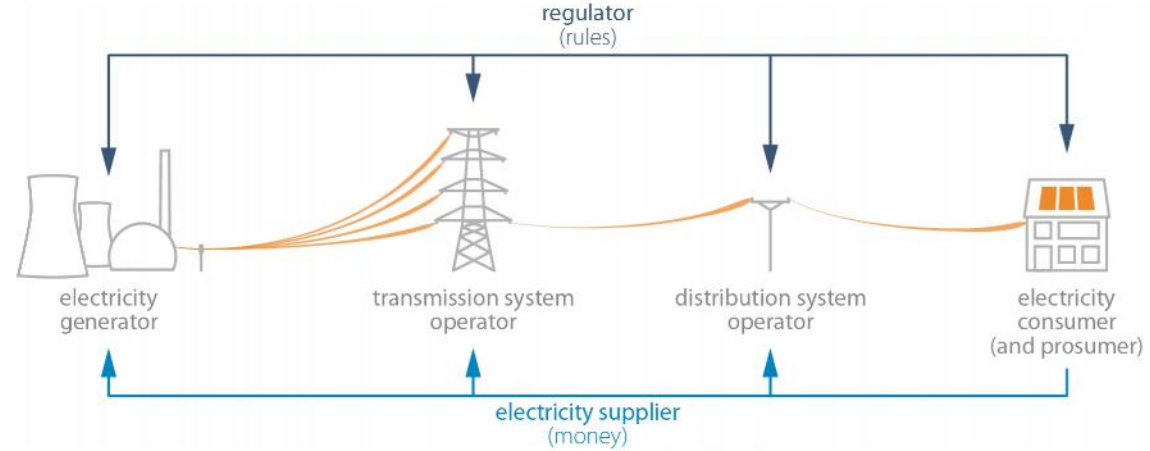
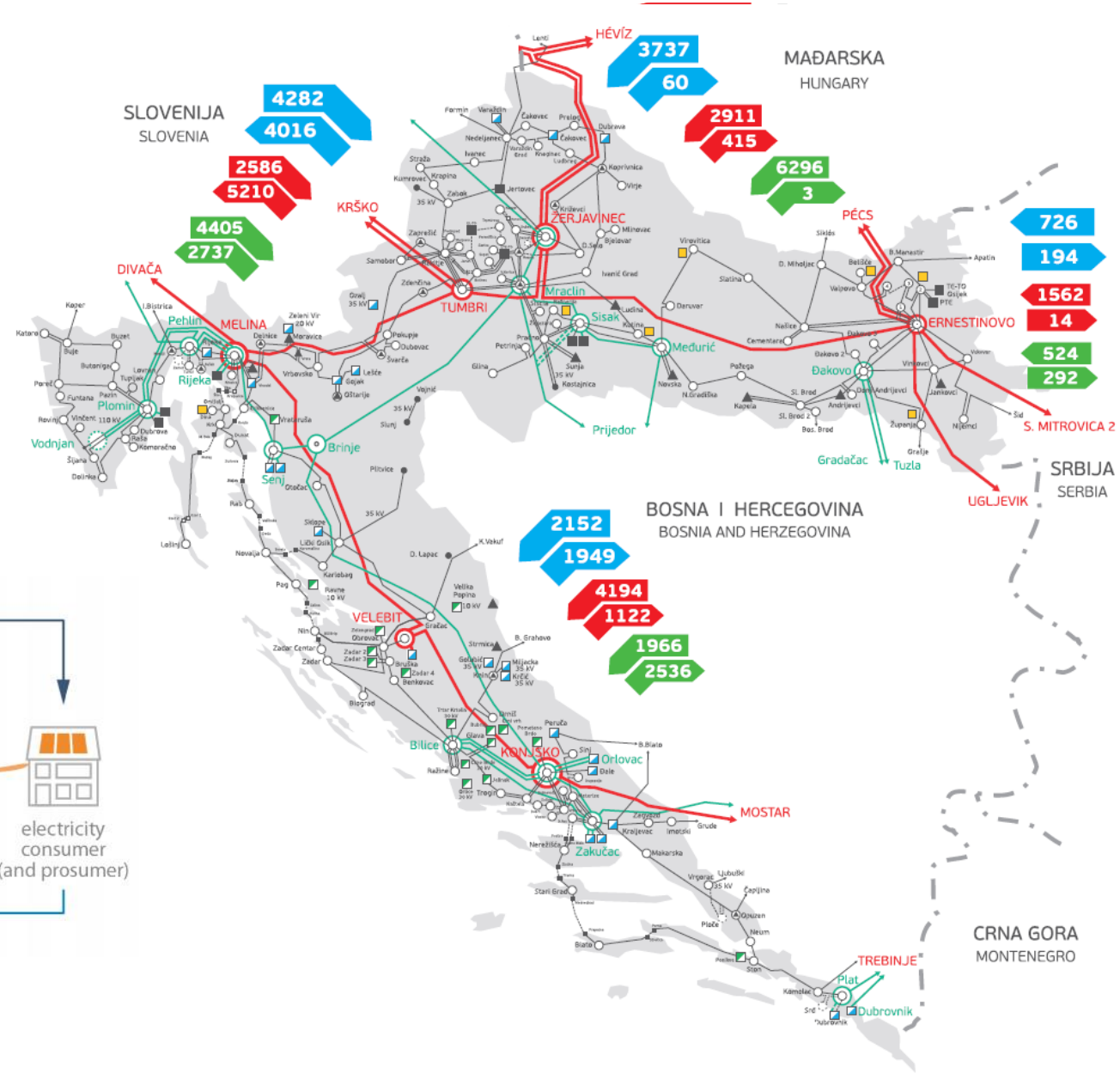


Figure 1 – Schematic overview of the electricity system

Graphic by EPRS.

https://www.cleanenergywire.org/factsheets/set-and-challenges-germanys-power-grid



About team

- Dubravka Bartula
- Ivana Medvedec
- Karmen Karlušić
- Krunoslav Hrnjak
- Darko Matas

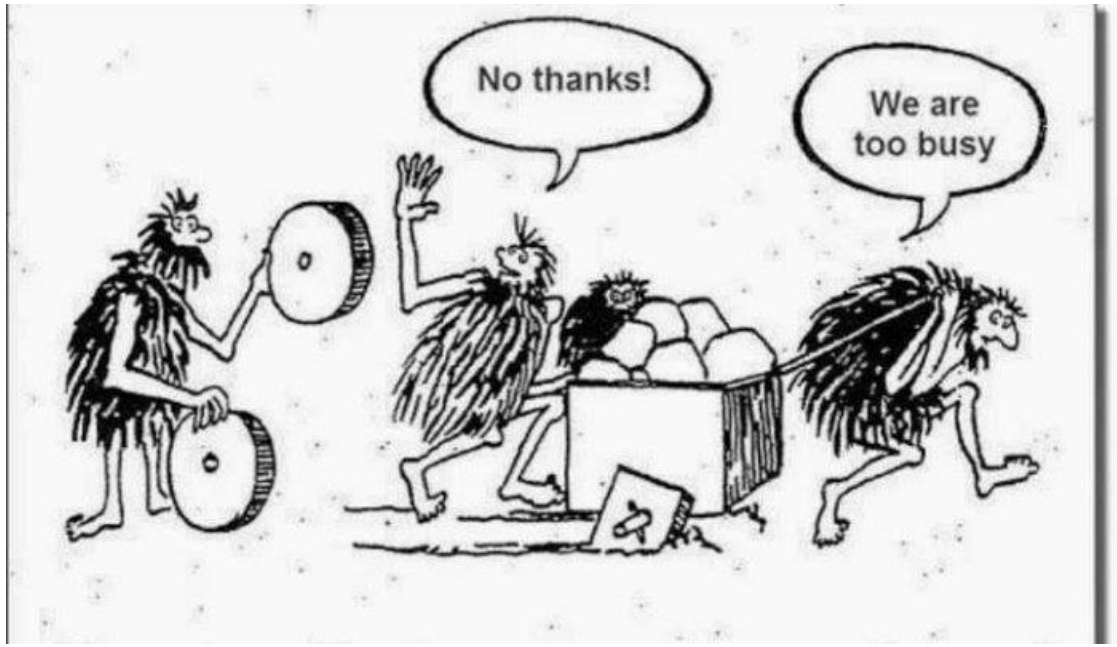
Personal experience

- GURU
  - Linux
  - Oracle DB
  - WL 12c
  - Tomcat
  - Apache
  - Java

Personal preference towards Java and FOSS based solutions

Is this presentation for me?

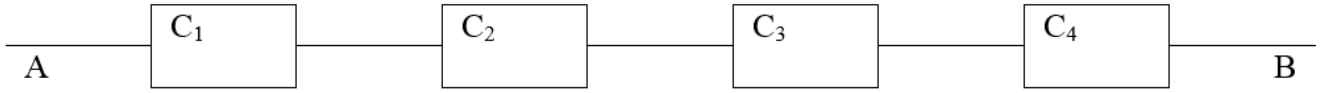
- Audience:
  - Oracle DBAs
  - IT architects



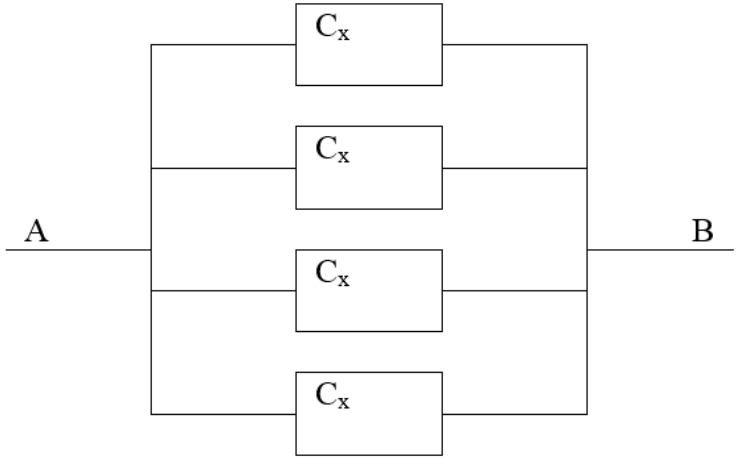
Problem - conceptual

- Out of the box HA solution for Oracle DB
- Free (no additional costs)
- KISS (keep it simple stupid)

HA calculation



$$A = \prod_{i=1}^{n} A_i$$



$$A = 1 - (1 - A_x)^n$$

Downtime

Availability	Max. downtime		
	[day]	[hour]	[minute]
95,000%	18,25	438	26280
99,000%	3,65	87,6	5256
99,900%	0,365	8,76	525,6
99,990%	0,0365	0,876	52,56
99,999%	0,00365	0,0876	5,256

Options

- Oracle Real Application Clusters (RAC) allows a database to be spread across multiple servers.
- Oracle RAC One Node provides an Oracle Database failover solution for that facilitates the clustered infrastructure.
- **Single Instance Oracle Database protected by Oracle Clusterware**

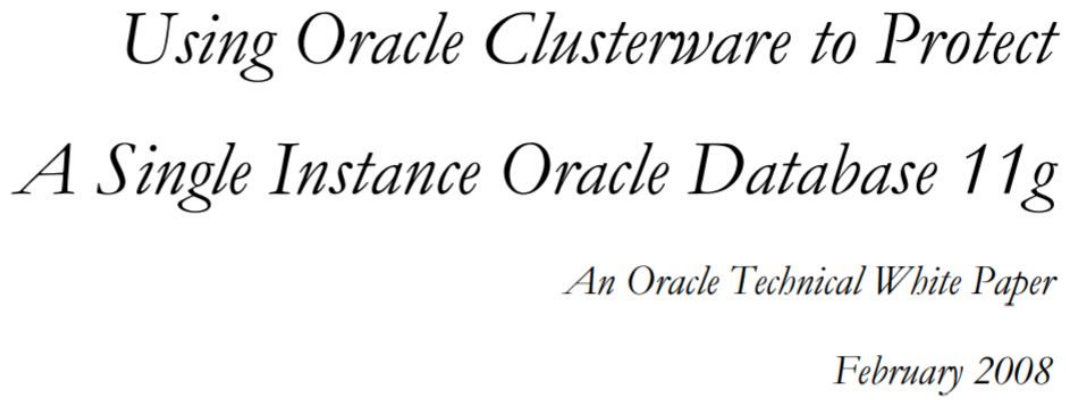
- **Single Instance** is a simple and basic option of deploying oracle database. In single instance database there is only one instance of database running on one node. It is simple to install and maintain.

Solution

- Oracle Clusterware can be used to protect any application (restarting or failing over the application in the event of a failure), free of charge, if one or more of the following conditions are met:
- 1. The server OS is supported by a valid Oracle Unbreakable Linux support contract.
- 2. The product to be protected is either:
  - - Any Oracle product (e.g. Oracle Applications, Siebel, Hyperion,
  - Oracle Database EE, Oracle Database XE)
  - - Any third-party product that directly or indirectly stores data in an Oracle database
- 3. At least one of the servers in the cluster is licensed for Oracle Database (SE or EE)

- For active/passive Failover Cluster environment you can benefit from the “10-day rule usage”
- In this type of environment, Oracle permits its licensed Technology customers to run the Technology Programs (listed on the Technology Price List) on an unlicensed spare computer for up to a total of ten separate days in any given calendar year.
- Only one failover node per clustered environment is at no charge for up to ten separate days even if multiple nodes are configured as failover nodes.

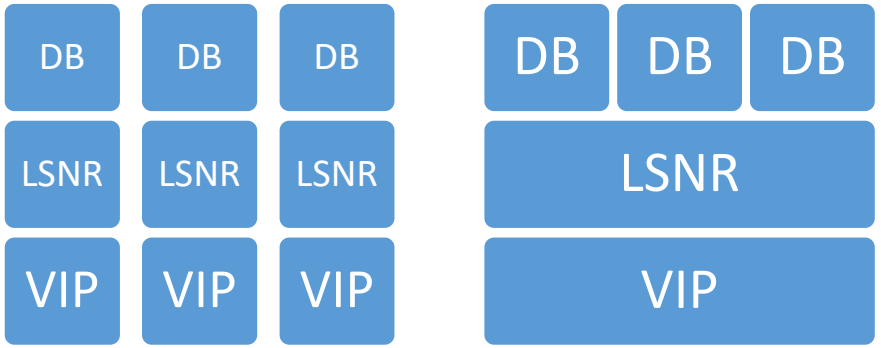
Solution - implementation



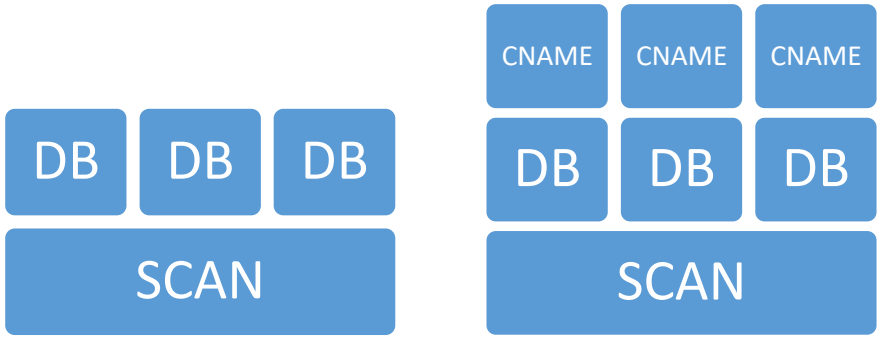
- The scripts provided as part of this paper are sample code which can be used to base your own scripts on. These scripts have been tested on an Oracle Enterprise Linux - 2 node cluster. It is expected that they should work on all Oracle Clusterware supported platforms. **Oracle Support cannot provide any direct support for these scripts.**
- https://www.oracle.com/technetwork/products/clusterware/overview/si-db-failover-11g-134623.pdf

Solution - concept

- Traditional approach



- SCAN concept (enhanced)



v10g vs. 12c - concept

- Scripts
  - creation scripts
    - crea_dir
    - crea_res
    - del_dir
    - del_res
  - action scripts
    - start,
    - stop,
    - Clean,
    - check
- uses VIP

- No scripts!
- uses SCAN
  - flexibility trough DNS pointer

Script - example

```
#!/bin/sh

export ORA_CRS_HOME=$_CRS_ORA_CRS_HOME
export ORACLE_HOME=$_CRS_ORACLE_HOME
export ORACLE_SID=$_CRS_ORACLE_SID
export SPFILE=$_CRS_SPFILE
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH

NODE_NAME=$(${ORA_CRS_HOME}/bin/olsnodes -l)
VIP_IP=$(${ORA_CRS_HOME}/bin/srvctl config vip -n $NODE_NAME | awk '{print $3}' | cut -
d"/" -f3)

check_dbstatus() {
  OUTPUT=`$ORACLE_HOME/bin/sqlplus -s /nolog<<EOF
connect / as sysdba
set echo off define off heading off pagesize 0
SET LINESIZE 100
COLUMN l_output FORMAT A100
SELECT i.status || ' ' || pa.value || '/' || 'ora_' || p.spid || '.aud' AS l_output
FROM v\\$session s,
     v\\$process p,
     v\\$parameter pa,
     v\\$instance i
WHERE pa.name = 'audit_file_dest'
AND s.paddr = p.addr
AND s.sid = (select sid from v\\$mystat where rownum=1)
and s.audsid = sys_context('userenv','sessionid');
exit
EOF`
  DBSTATUS=$(echo $OUTPUT | awk '{print $1}')
  AUDITFILE=$(echo $OUTPUT | awk '{print $2}')
  rm -f $AUDITFILE 2>/dev/null

  if [ "$DBSTATUS" == "OPEN" ]
  then
    return 0
  else
    return 1
  fi
}

case $1 in

'start')
echo "spfile='${SPFILE}'" > /tmp/init${ORACLE_SID}.ora
$ORACLE_HOME/bin/sqlplus /nolog <<EOF
connect / as sysdba
startup pfile=/tmp/init${ORACLE_SID}.ora
alter system set
local_listener='(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST
=${VIP_IP})(PORT=1521))))' scope=memory;
EOF
check_dbstatus
RET=$?
;;

'stop')
$ORACLE_HOME/bin/sqlplus /nolog <<EOF
connect / as sysdba
shutdown immediate
EOF
NUM=`ps -ef | grep -i smon_${ORACLE_SID} | grep -v grep | wc -l`
if [ $NUM = 0 ]; then
  RET=0
else
  RET=1
fi
;;

'clean')
$ORACLE_HOME/bin/sqlplus /nolog <<EOF
connect / as sysdba
shutdown abort
EOF
##for i in `ps -ef | grep -i mon_${ORACLE_SID} | awk '{print $2}' ` ;do
kill -9 $i; done
NUM=`ps -ef | grep -i smon_${ORACLE_SID} | grep -v grep | wc -l`
if [ $NUM = 0 ]; then
  RET=0
else
  RET=1
fi
;;

'check')
check_dbstatus
RET=$?
;;

'*')
RET=0
;;
esac

if [ $RET -eq 0 ]; then
  exit 0
else
  exit 1
fi
```

v10g vs. 12c – simple approach

- crsctl stat res
  - NAME=ora.dbtest.db
  - TYPE=application
  - TARGET=ONLINE
  - STATE=ONLINE on szgdb1

  - NAME=ora.dbtest.dbtest.inst
  - TYPE=application
  - TARGET=ONLINE
  - STATE=ONLINE on szgdb1

  - NAME=ora.lsdbtest.lsnr
  - TYPE=ora.scan_listener.type
  - TARGET=ONLINE
  - STATE=ONLINE on szgdb1

  - NAME=ora.dbtestfo.vip
  - TYPE=ora.cluster_vip.type
  - TARGET=ONLINE
  - STATE=ONLINE on szgdb1

- crsctl stat res
  - **NAME=ora.dbtest.db**
  - **TYPE=ora.database.type**
  - **TARGET=ONLINE**
  - **STATE=ONLNE on host1**

  - NAME=ora.ons
  - TYPE=ora.ons.type
  - TARGET=ONLINE , ONLINE
  - STATE=ONLINE on host1, ONLINE on host2

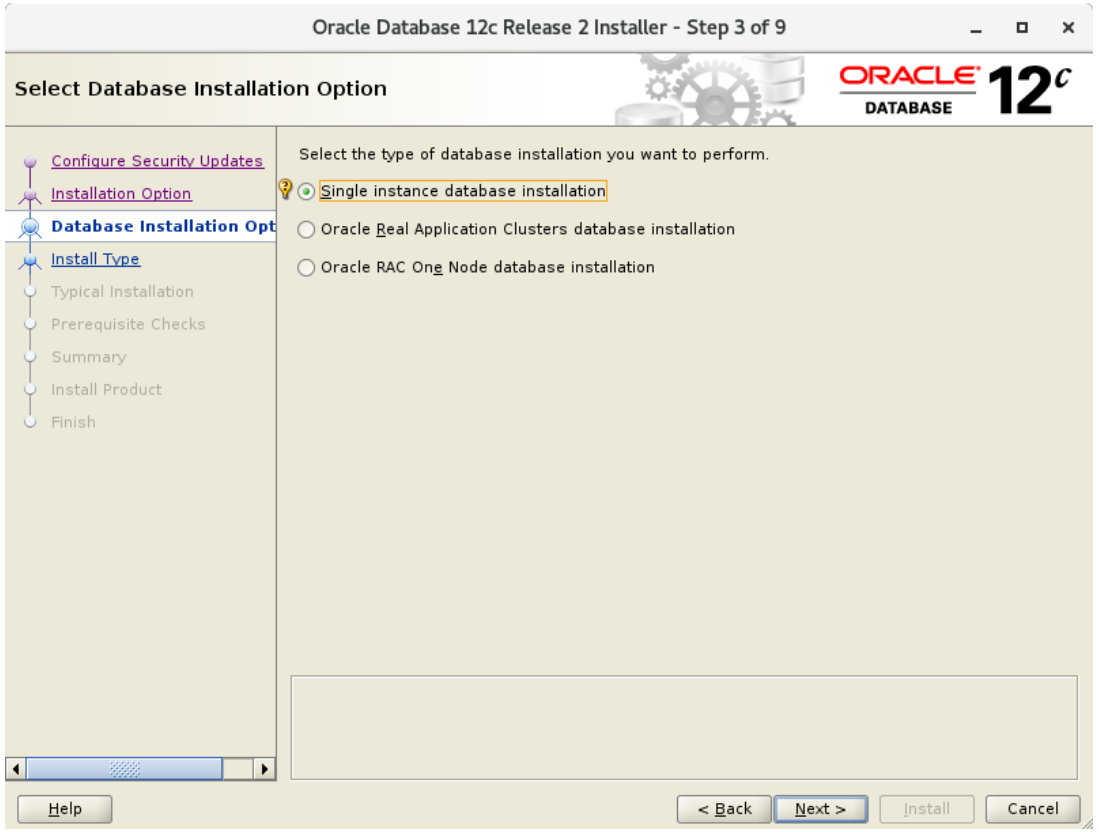
  - NAME=ora.scan1.vip
  - TYPE=ora.scan_vip.type
  - TARGET=ONLINE
  - STATE=ONLINE on host1

- nslookup dbtest.local
  - Server: 10.1.1.100
  - Address: 10.1.1.100#53
  - **dbtest.local canonical name = oracle-scan.local**
  - Name: oracle-scan.local
  - Address: 10.1.1.31

Installation steps

- 1. Install operating system on all nodes
- 2. Install Oracle Grid Infrastructure (automatically deployed on all nodes in a cluster)
- 3. Install Oracle Database software on each node separately !!!!
- 4. Create DBTEST DB using DBCA (Database Configuration Assistant)
- 5. Copy init.ora file to other hosts
  - cat initDBTEST.ora
  - spfile='+DATA/DBTEST/spfileDBTEST.ora'

Installation steps – warning!



- On top of the Grid Infrastructure build, you do a local installation of the RDBMS. It is important that you do not choose a cluster installation when prompted so; otherwise, you risk violating your license agreement with Oracle.

Magic – no scripts required

- Create a database using DBCA
  - DBCA makes registration with grid
- Check the current properties of your database - dbtest here
  - `crsctl status resource ora.dbtest.db -p`
- Change the placement to FAVORED for your database
  - **`crsctl modify resource ora.dbtest.db -attr "PLACEMENT=favored" -unsupported`**
- Change the list of HOSTs that can receive the instance of your database (here host1 or host2)
  - **`crsctl modify resource ora.dbtest.db -attr "HOSTING_MEMBERS=host1 host2" -unsupported`**
- Check back the new properties of your database - dbtest here
  - `crsctl status resource ora.dbtest.db -p`

Info about DB

- [oracle@host1 ~]$. oraenv
- ORACLE_SID = [oracle] ? +ASM1

- crsctl status resource ora.dbtest.db -p
- NAME=ora.dbtest.db
- TYPE=ora.database.type
- STATE=ONLINE
- TARGET=ONLINE
- DATABASE_TYPE=SINGLE
- DESCRIPTION=Oracle Database resource
- ...
- HOSTING_MEMBERS=odb1-test odb2-test
- PLACEMENT=favored
- ...

Relocate DB

- Relocate one DB
  - `crsctl relocate resource  -s host1 -f -w "NAME = ora.dbtest.db"`
- Relocate all DBs
  - `crsctl relocate resource  -s host1 -f -w "TYPE = ora.database.type"`
- Relocate all resources
  - `crsctl relocate resource  -all -s host1 -f`

DB downtime

```
START=$(date +%s)
crsctl relocate resource  -s odb2-test -f -w "NAME = ora.dbtest.db"
END=$(date +%s)
DIFF=$(( $END - $START ))
echo "It took $DIFF seconds"

> It took 47 seconds
```

alert_dbtest.log

host1

cat /u01/app/oracle/diag/rdbms/dbtest/dbtest/trace/alert_dbtest.log

Shutting down instance (immediate)

...

Instance shutdown complete

host2

cat /u01/app/oracle/diag/rdbms/dbtest/dbtest/trace/alert_dbtest.log

Starting ORACLE instance (normal)

...

System parameters with non-default values:

...

local_listener = "(ADDRESS=(PROTOCOL=TCP)(HOST=10.11.1.123)(PORT=1521))"

remote_listener = " db-scan:1521"

...

Completed: ALTER DATABASE OPEN

SCAN listener

```
SQL> show parameter LISTENER

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------
local_listener                       string
          (ADDRESS=(PROTOCOL=TCP)(HOST=host1-vip.local)(PORT=1521))
remote_listener                      string
          oracle-scan.local:1521
```

```
$ ps aux | grep -i lreg
oracle   13942  …   Feb14   0:06 ora_lreg_dbtest
```

- An Oracle Database 12c release 2 (12.2) database service automatically registers with the listeners specified in the database initialization parameters LOCAL_LISTENER and REMOTE_LISTENER.

- Parameter **LOCAL_LISTENER** by default points to the local host and port 1521, **REMOTE_LISTENER** is set by the *DBCA* after creation to SCAN name and port

RMAN DUPLICATE from an Active Database

- To create standby database from active database *RMAN* requires to connect from primary site to auxiliary instance on remote standby site.
- Auxiliary instance started in nomount mode is dynamically registered with SCAN listener in BLOCKED status.
- Service must be statically, i.e. manually registered to the listener

Add static listener

- vi $GRID_HOME/network/admin/listener.ora

```
SID_LIST_LISTENER =
   (SID_LIST =
       (SID_DESC =
          (ORACLE_HOME= /u01/app/oracle/product/11.2.0.4/db)
          (SID_NAME = dbtest)
          (GLOBAL_DBNAME= DBTESTHOST1)
       )
)
```

- vi $GRID_HOME/network/admin/tnsnames.ora

```
DBTESTHOST1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = host1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DBTESTHOST1)
    )
  )
```

Add static listener

- **lsnrctl status**

- LSNRCTL for Linux: Version 12.2.0.1.0 - Production on 16-FEB-2019 16:10:23
- Copyright (c) 1991, 2016, Oracle. All rights reserved.
- Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
- STATUS of the LISTENER
- ------------------------
- Alias LISTENER
- Version TNSLSNR for Linux: Version 12.2.0.1.0 - Production
- Start Date 23-JAN-2019 18:37:10
- Uptime 23 days 21 hr. 33 min. 13 sec
- Trace Level off
- Security ON: Local OS Authentication
- SNMP OFF
- Listener Parameter File /u01/app/12.2.0.1/grid/network/admin/listener.ora
- Listener Log File /u01/app/oracle/diag/tnslsnr/host1/listener/alert/log.xml

- Listening Endpoints Summary...
- (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
- (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.31.1.217)(PORT=1521)))
- (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.31.1.227)(PORT=1521)))
- Services Summary...
- Service "+ASM" has 1 instance(s).
- Instance "+ASM1", status READY, has 1 handler(s) for this service...
- Service "+ASM_CRS" has 1 instance(s).
- Instance "+ASM1", status READY, has 1 handler(s) for this service...
- Service "+ASM_DATA" has 1 instance(s).
- Instance "+ASM1", status READY, has 1 handler(s) for this service...
- Service "+ASM_FRA" has 1 instance(s).
- Instance "+ASM1", status READY, has 1 handler(s) for this service...
- **Service "dbtesthost1" has 1 instance(s).**
- **Instance "dbtest", status UNKNOWN, has 1 handler(s) for this service...**
- Service "dbtest" has 1 instance(s).
- Instance "dbtest", status READY, has 1 handler(s) for this service...
- The command completed successfully

RAC check

- **How to Check Whether Oracle Binary/Instance is RAC Enabled and Relink Oracle Binary in RAC (Doc ID 284785.1)**
  - Oracle binary is RAC enabled
    - `cd $ORACLE_HOME/rdbms/lib`
    - `ar -t libknlopt.a | grep -c kcsm.o`
  - Running instance is a RAC instance
    - **ps -ef| grep lmon | grep <ORACLE_SID>**
    - oracle 627 1 0 Apr 15 ? 1:02 ora_lmon_racdb1
  - Other options:
    - 1. Check sqlplus banner
    - Check cluster_database parameter

Usefull links

- https://aychin.wordpress.com/2015/06/25/about-listener-static-registration-and-standby-db-from-active-database-creation-in-rac-environemnt/
- https://dbaspot.wordpress.com/2013/06/09/action-script-for-clusterware-11g-activepassive-failover-configuration/
- http://msutic.blogspot.com/2014/04/how-to-configure-activepassive-failover.html
- http://www.hroug.hr/content/download/4954/76172/file/716_Patari%C4%87-NeslKlasteri.pdf
- https://gjilevski.com/2012/01/09/build-active-passive-ha-configuration-for-single-instance-database-with-oracle-gi-11-2-0-3/
- https://mathijsbruggink.com/2013/07/04/thoughts-on-adding-a-single-instance-to-oracle-clusterware-grid-infra/
- https://mishrabagish.wordpress.com/2012/05/01/oracle-single-instance-vs-rac-vs-rac-one-node/
- https://oraganism.wordpress.com/2014/04/13/protect-single-instance-with-gi-without-rac-one-node/
- https://www.doag.org/formes/pubfiles/2489339/DOAG_2011-03-31_Failover.DB.Cluster.with.Grid.Infrastructure.pdf
- https://www.informatik-aktuell.de/fileadmin/templates/wr/pics/Seminare/Pr%C3%A4sentationen/Jerome_Witt_IT-Tage_2015.pdf
- https://www.oracle.com/technetwork/products/clusterware/overview/si-db-failover-11g-134623.pdf

Thank you! Q&A

HOPS

About author

CRO

- Krunoslav voli FOSS (slobodan softver i softver otvorenog izvornog koda) s posebnim naglaskom na Java, Linux i Oracle tehnologije. On je dinamičan, fleksibilan i visoko kvalificirani profesionalac programskog inženjerstva. Posjeduje dobre sposobnosti upravljanja projektima, interakcije i koordinacije. Uvijek zainteresiran za pronalaženje i donošenje novih rješenja i metoda na modernim tehnologijama. Radi i kao sudski vještak za informatiku i telekomunikacije, a također je i predavač na Sveučilišnom računskom centru (SRCE).

ENG

- Krunoslav Hrnjak, M. Sc. E.E., PMP, is ICT professional with ten years' experience. Krunoslav loves FOSS (Free and Open Source Software) with special interest in Java, Linux and Oracle. He is dynamic, flexible and high skilled professional on software engineering. Good capabilities of project management, interaction and coordination. Always interested on finding and adopting new solutions and methods at the top of the technology. He also works as Court Witness Expert for Informatics and Telecommunications and also as an IT course lecturer at University Computing Centre (SRCE).

Abstract –
Oracle Unbreakable Cluster v.12c

CRO

- Oracle Grid infrastruktura u verziji 12c nudi sve potrebne komponente za uspostavu visoko dostupnog sustava Oracle 11g i 12c baze podataka bez potrebe za dodatnim licencama.
- Korištenjem najbolje prakse u implementaciji postiže se fleksibilnost i robusnost servisa, a ujedno i jednostavnost bez potrebe za skriptiranjem.

ENG

- The Oracle Grid infrastructure in Version 12c offers all the components required to establish a highly available Oracle 11g and 12c database system without the need for additional licenses.
- Using best practices in implementation, the flexibility and robustness of the service is achieved, while simplicity without the need for scripting.