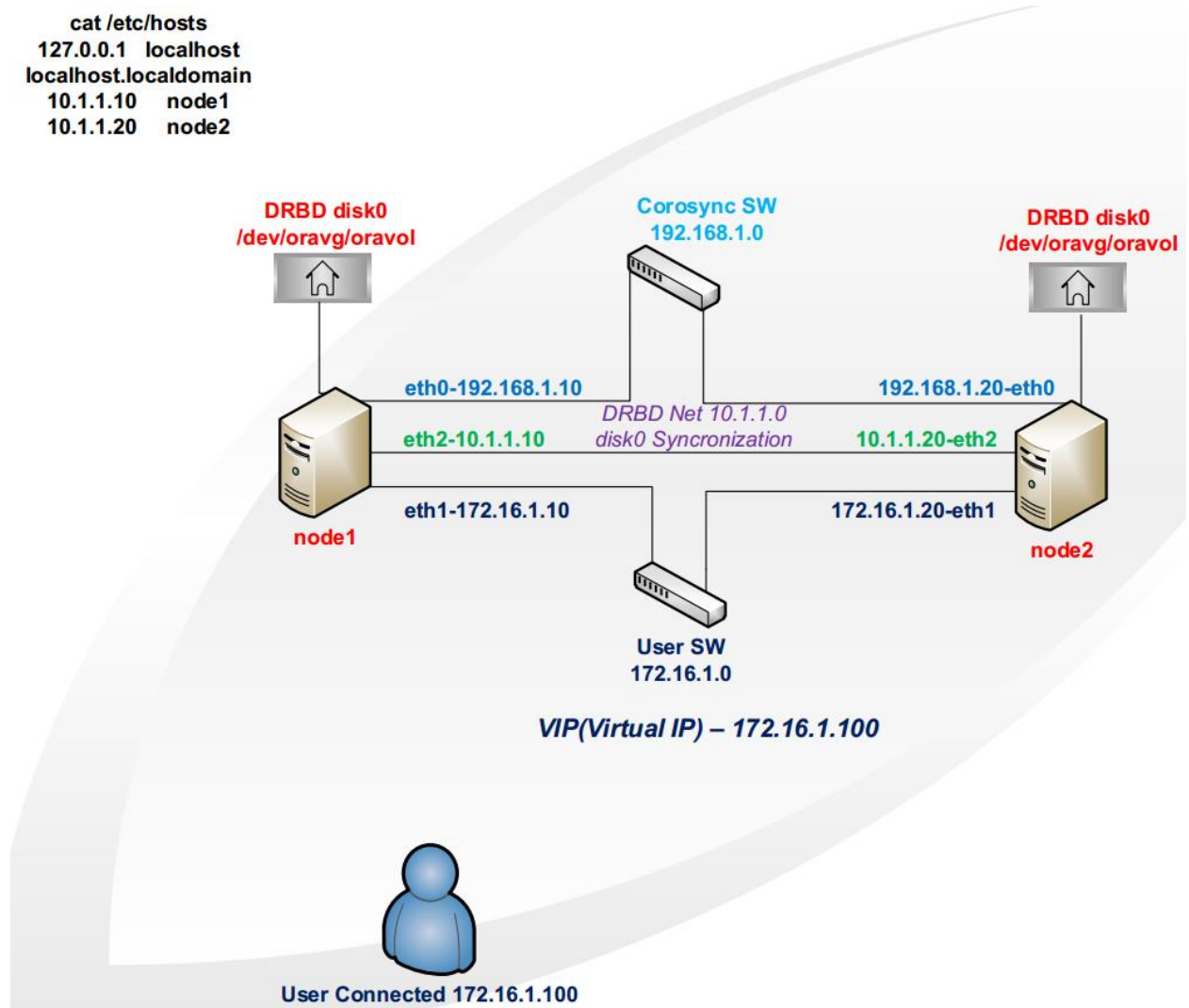


DRBD-Corosync-Pacemaker vasitəsilə Active/Passive Oracle Cluster

Bütün edəcəklərimiz VmWare-Workstation 9.0 üzərində sınaqdan keçirilmişdir. Qeyd etmək istəyirəm ki, bütün CentOS6.3_x64 maşınlar 192.168.1.1 GateWay üzərindən internetə çıxırlar.

Şəbəkə quruluşu aşağıdakı şəkildəki kimi olacaq:

```
cat /etc/hosts
127.0.0.1 localhost
localhost.localdomain
10.1.1.10 node1
10.1.1.20 node2
```



192.168.1.1 - FreeBSD maşındır (Nat ilə internet verir və VmNet3-də yerləşir)

Öncədən edəcəklərimiz.

Qeyd: Bütün işlər "root" istifadəçisi adından görülür.

2 ədəd serverimiz var. Hər serverimizin 3 şəbəkə kartı var.

1. 192.168.1.0/24 - Corosync NetWork (VmNet3)
2. 172.16.1.0/24 - User NetWork (VmNet4-LoopBack)
3. 10.1.1.0/24 - DRBD NetWork (VmNet7)

Hər iki serverdə **/etc/hosts** faylı identik olmalıdır.

```
cat /etc/hosts # Hər iki serverdə eynidir.  
127.0.0.1 localhost localhost.localdomain  
10.1.1.10 node1  
10.1.1.20 node2
```

node1 (Net IP addresses, Subnet MASK və Gateway)

```
eth0 - 192.168.1.10/24  
eth1 - 172.16.1.10/24  
eth2 - 10.1.1.10/24  
Gateway - 192.168.1.1
```

node2 (Net IP addresses, Subnet MASK və Gateway)

```
eth0 - 192.168.1.20/24  
eth1 - 172.16.1.20/24  
eth2 - 10.1.1.20/24  
Gateway - 192.168.1.1
```

Hər iki serverdə SSH qeydiyyatı tokenlə olmalıdır.

node1 üçün ediləcək işlər.

```
# ssh-keygen -t rsa # Token üçün şifrə daxil etmədən açar cütlüyü  
# yaradırıq(açarlar '/root/.ssh' qovluğunun altında  
# yaranacaq).  
  
# scp id_rsa.pub node2:/root/.ssh/node1.id_rsa.pub # 'node1'-in açarını  
# adını təyin edərək  
# 'node2' serverinə  
# nüsxələyirik.  
  
# cat node1.id_rsa.pub >> authorized_keys # Bu əmri isə 'node2'  
# serverində '/root/.ssh'  
# folderinin içindən  
# daxil edirik.
```

node2 üçün ediləcək işlər.

```
# ssh-keygen -t rsa # Token üçün şifrə daxil etmədən açar cütlüyü  
# yaradırıq(açarlar '/root/.ssh' qovluğunun  
# altında yaranacaq)  
  
# scp id_rsa.pub node1:/root/.ssh/node2.id_rsa.pub # 'node2'-in açarının  
# adını təyin edərək  
# 'node1' serverinə  
# nüsxələyirik.  
  
# cat node2.id_rsa.pub >> authorized_keys # Bu əmri isə 'node1'
```


Sistemə **oracle** istifadəçi adı üçün fayl və processlərə məhdudiyət təyin edirik.

```
# vi /etc/security/limits.conf      # Faylın sonuna aşağıdakı limitləri əlavə edirik.
```

```
#####
```

```
#### Settings required for Oracle 11
```

```
oracle soft nproc 2047
```

```
oracle hard nproc 16384
```

```
oracle soft nofile 1024
```

```
oracle hard nofile 65536
```

```
oracle soft stack 10240
```

```
oracle hard stack 32768
```

```
#####
```

Global Profayla tələb edilən şərtləri mənimsədirik ki, oracle istifadəçi adı sistemə yükləndikdə o şərtləri mənimsəsin.

```
# vi /etc/profile      # Faylın sonunda 'unset -i' və 'unset pathmunge'-dən önce aşağıdakı sətirləri əlavə edirik.
```

```
#####
```

```
#### Shell limits for Oracle 11 user accounts
```

```
if [ $USER = "oracle" ]; then
```

```
    ulimit -u 16384 -n 65536
```

```
    # -u istifadəçi
```

```
    processlərinin maksimum sayı,
```

```
fi
```

```
    # -n istifadəçi üçün maksimal açıla
```

```
    biləcək faylların sayı.
```

```
#####
```

```
unset i
```

```
unset -f pathmunge
```

```
# vi /etc/pam.d/login      # Oracle üçün PAM modulunu əlavə edirik. Fayla Aşağıdakı sətirləri əlavə edirik.
```

```
#####
```

```
#### Settings required for Oracle
```

```
session required pam_limits.so
```

```
#####
```

```
# vi /home/oracle/.bash_profile      # Oracle istifadəçisinin adından daxil olub onun profile faylına aşağıdakı dəyişənlərə mənə mənimsədirik.
```

```
umask 022
```

```
export ORACLE_BASE=/u01/app/oracle
```

```
# Oracle bu qovluğa yüklənəcək və
```

```
bu səbəbdən də onu mütləq əlimizlə
```

```
yaradırıq(u01 adı oracle tərəfindən
```

```
məsləhət edilir)
```

```
# Bu ünvana Oracle tərəfindən
yazılan istənilən program təminatı
yüklenir və bu dəyişəni oxuyur.
# Məs: WebLogic, Oracle DB, Oracle
IAS və.s
```

```
export ORACLE_HOME=$ORACLE_BASE/product/11.2 # Ancaq 'Oracle_HOME'
dəyişəni isə mütləq
verilənlər bazası üçün
istifadə edilən dəyişəndir.
(Qovluğu yaratmaq lazımdır)
```

```
export ORACLE_SID=oradb # Oracle prosesinin sistem
Identifikatorunun adı
```

```
export ORACLE_UNQNAME=oradb # Bu unikal ad DataGuard
istifadə edilən halda tələb
edilir və mütləq deyil.
```

```
export NLS_LANG=AMERICAN_AMERICA.AL32UTF8 # Baza Standartı bizim halda
UTF8 ən yaxşısıdır.
```

```
export PATH=$PATH:$ORACLE_HOME/bin # Əmrlər yerləşdiyi ünvan
export LD_LIBRARY_PATH=$ORACLE_HOME/lib # Kitabxanalar yerləşdiyi
ünvan
```

Qeyd: Yuxarıda yazdığımız dəyişənlərin işə düşməsi üçün '**oracle**' istifadəçi adı ilə sistmə yenidən daxil olmaq lazımdır.

```
# mkdir -p /u01/app/oracle/product/11.2 # Oracle-ın yüklənəcəyi
qovluğu yaradaq(root
istifadəçi adından
yaradılır).
```

```
# chown -R oracle:oinstall /u01/ # Qovluğu 'oracle' istifadəçi
və 'oinstall' qrupunun üzvü
edirik.
```

```
# chmod -R 755 /u01/ # Qovluğa bütövlükdə rekursiv
olaraq oracle istifadəçi
adına yetki veririk.
```

Sonra '**Oracle_11g_R2_Linux_x86-64**' program təminatını **oracle** ev qovluğuna yükləyirik və qovluğu '**oracle**' istifadəçi və '**oinstall**' qrupunun üzvü edirik.

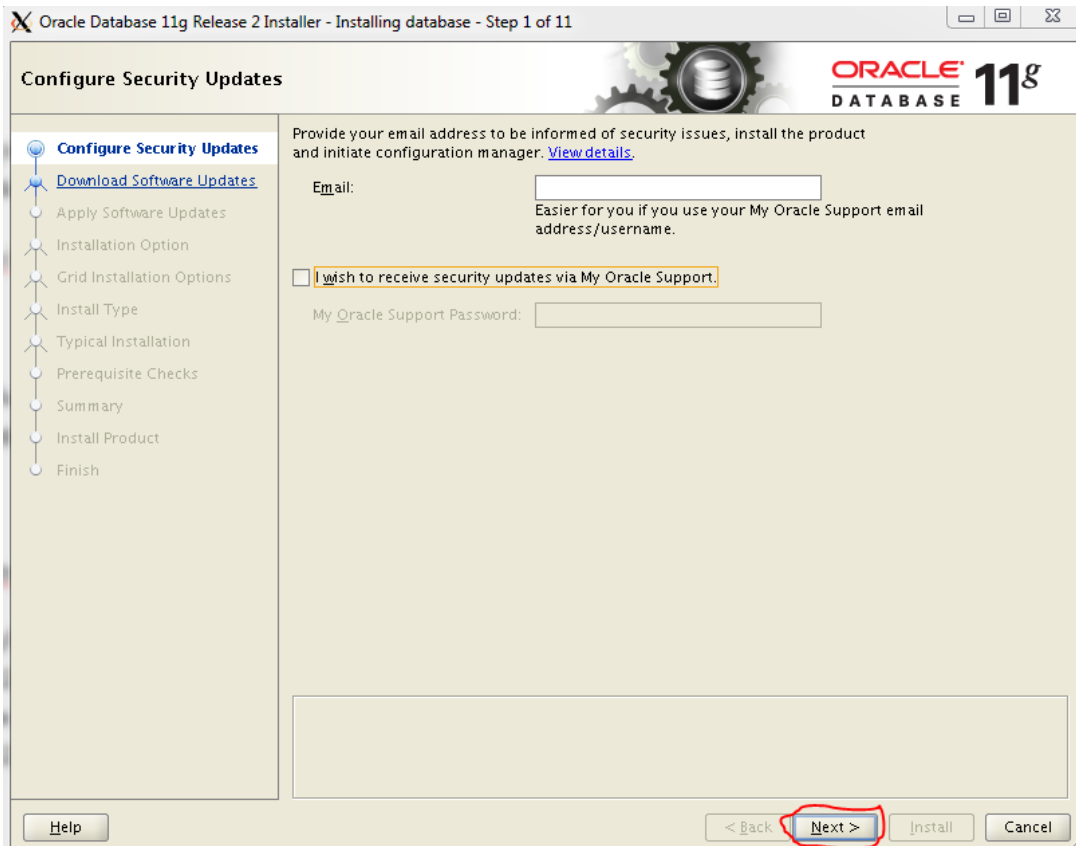
```
# chown -R oracle:oinstall /home/oracle/ # Distributiv home qovluğunda
olduğuna görə oracle
istifadəçi və oinstall qrupun
üzvü edirik.
```

```
# cd /home/oracle/Oracle_11g_R2_Linux_x86-64 # Oracle distributivi
# Oracle yüklədiyimiz qovluğa daxil
# oluruq. Orda iki zip fayl
# mövcuddur.

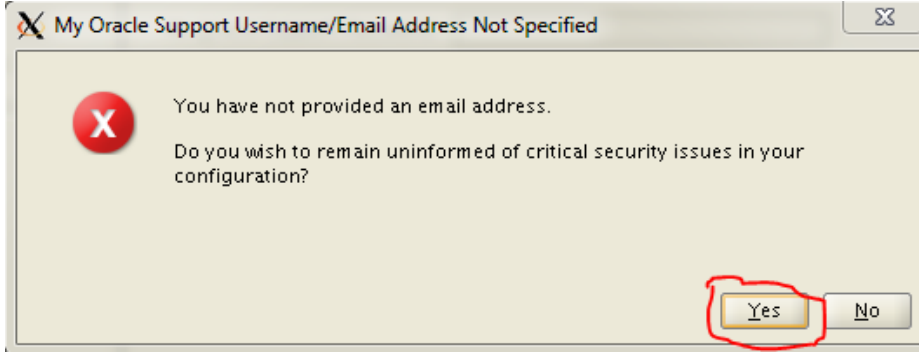
# unzip p10404530_112030_Linux-x86-64_1of7.zip; unzip p10404530_112030_Linux-
# x86-64_2of7.zip # Bu zip faylları ardıcıl
# olaraq açırıq.

# cd /home/oracle/Oracle_11g_R2_Linux_x86-64/database # Açdığımız qovluğa
# daxil oluruq.(X server
# artıq hazır olmalıdır)

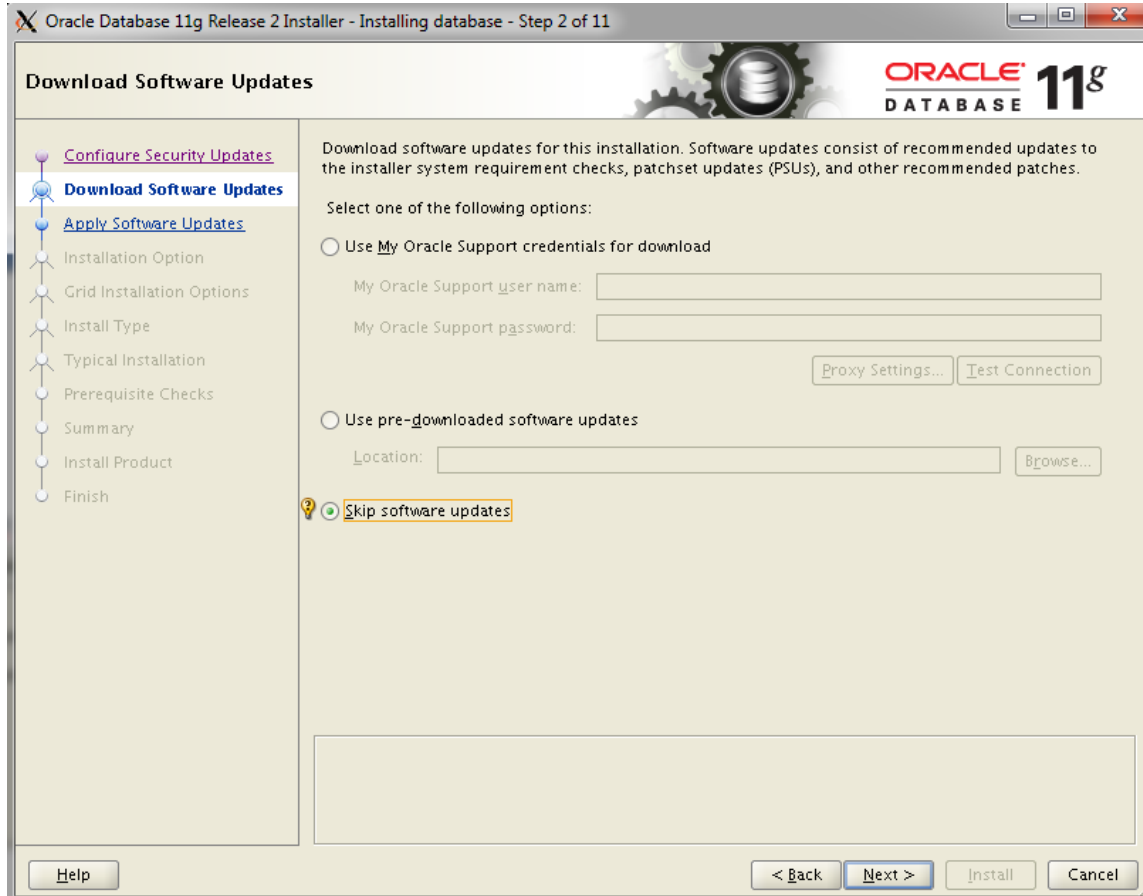
# ./runInstaller # Oracle yükləmə scriptini
# işə salırıq. Aşağıdakı
# şəkildəki kimi selector-dan
# seçimi götürürük və Next
```



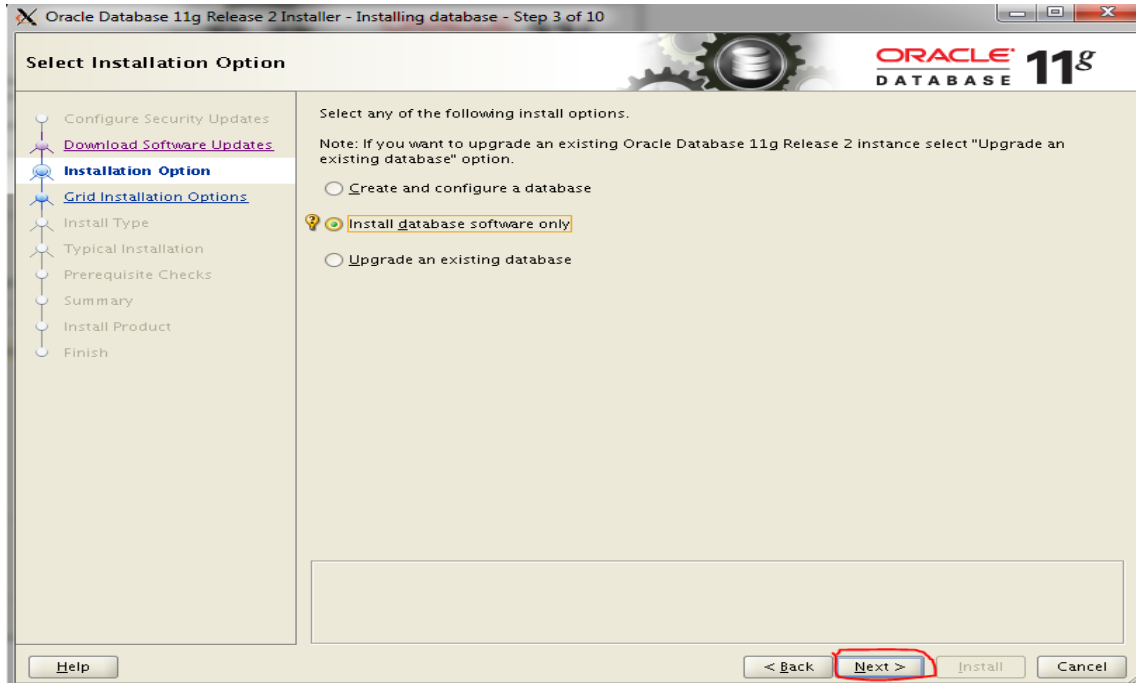
Növbəti şəkildə çıxan xəbərdarlığa fikir vermədən **yes** edirik.



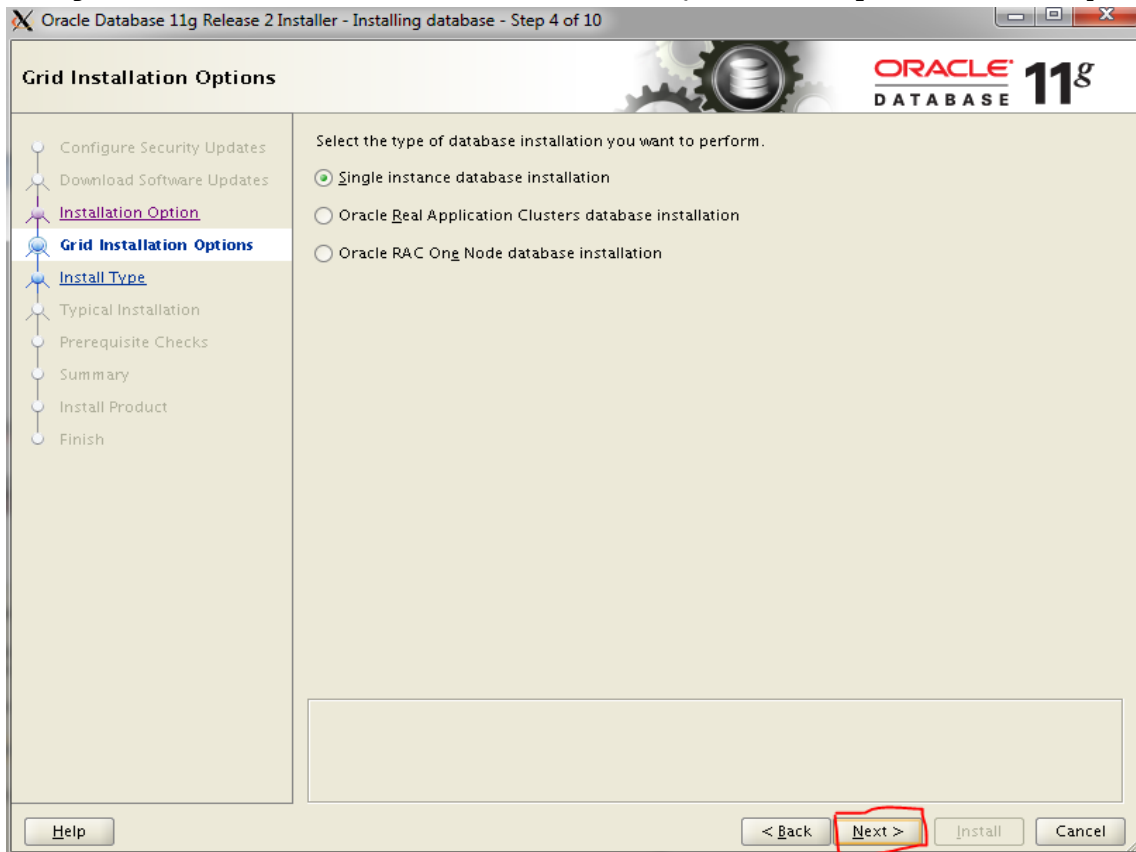
Və şəkildəki kimi **'skip software updates'** edirik.



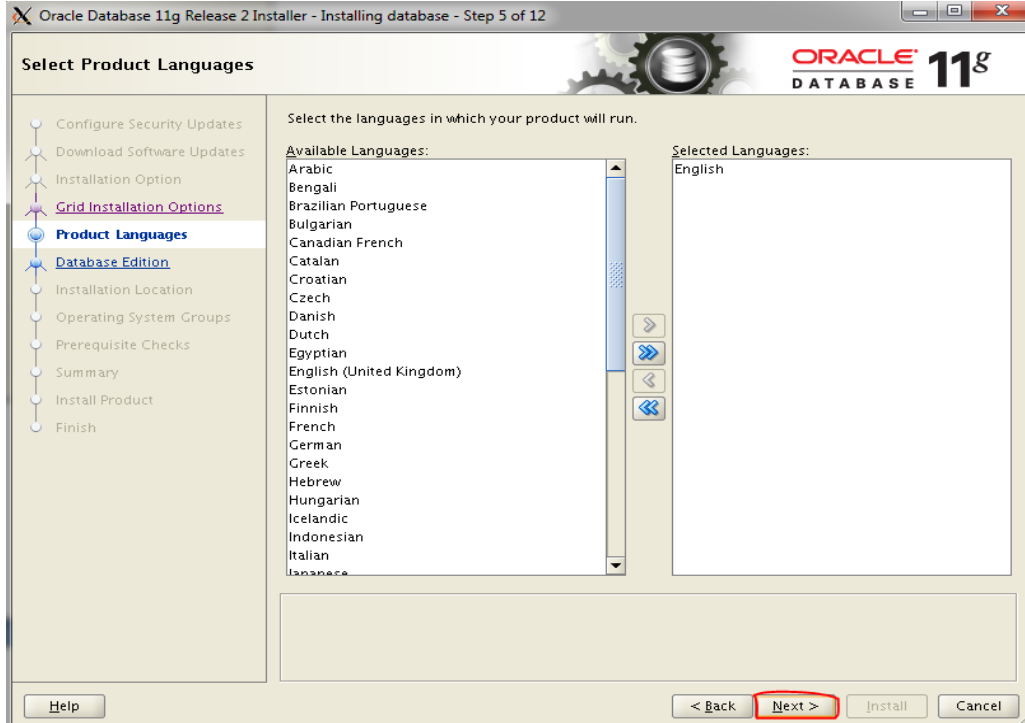
Sonra isə **'Install database software only'** seçib **Next** düyməsinə sıxırıq çünki, biz bazanı **DRBD** diske yükləyəcəyik sonra.



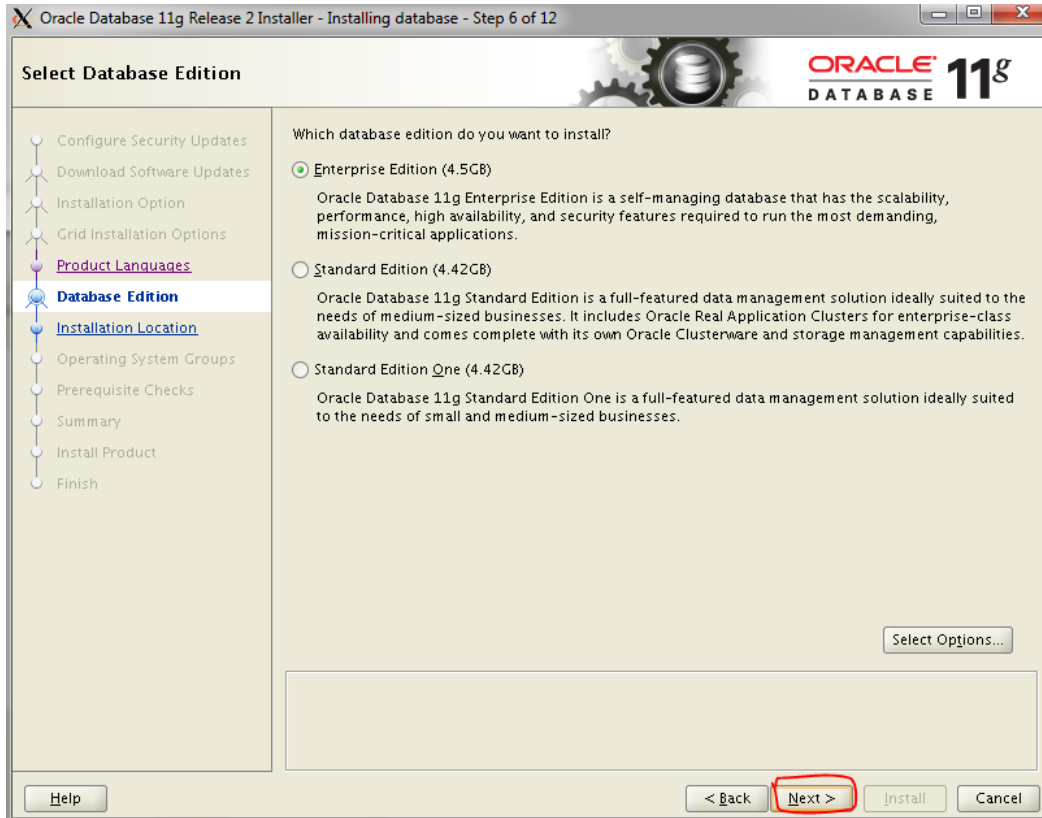
'Single instance database installation' seçib Next düyməsinə sıxırırq:



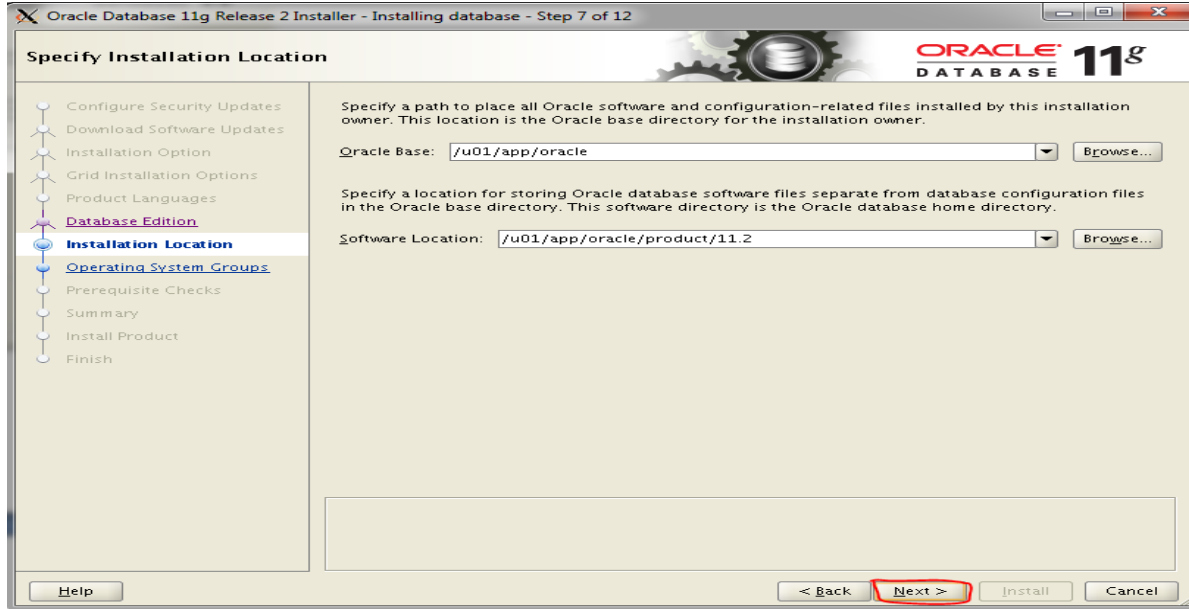
Yalnız ingilis dilini seçib **Next** düyməsini sıxırıq.



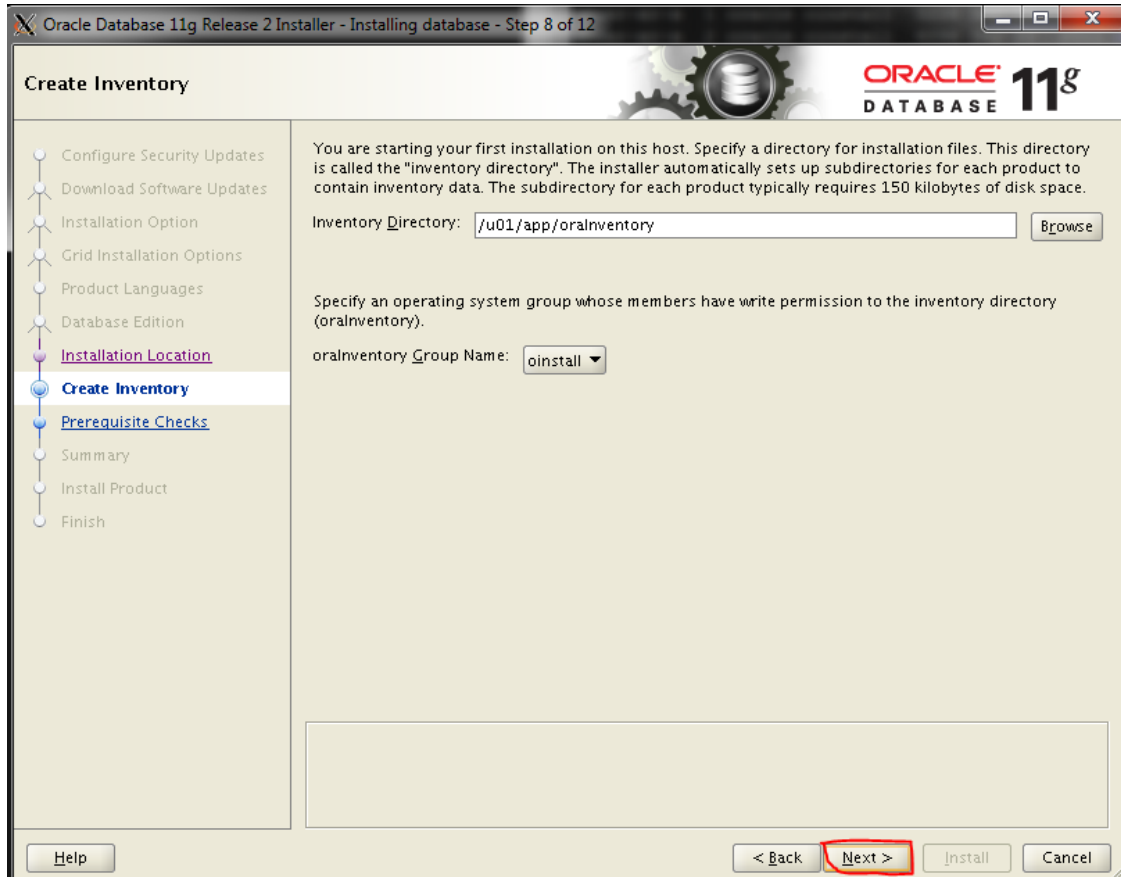
'Enterprise edition' seçib **Next** düyməsini sıxırıq.



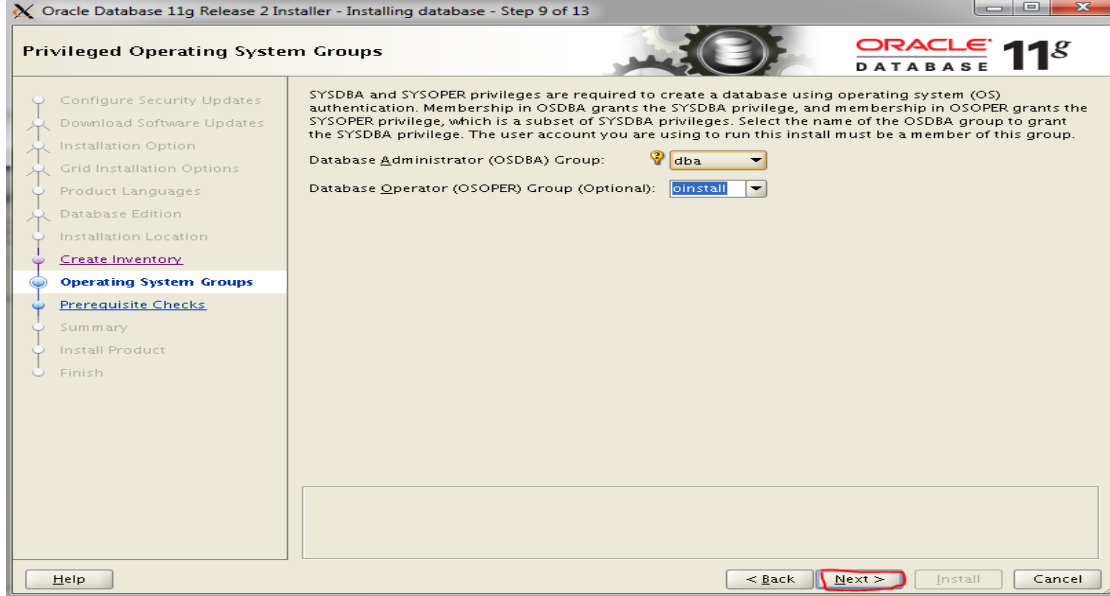
Ünvanlar olduğu kimi qalır çünki biz biraz öncə onları yaratmışdıq. Və **next** düyməsini sıxırıq.



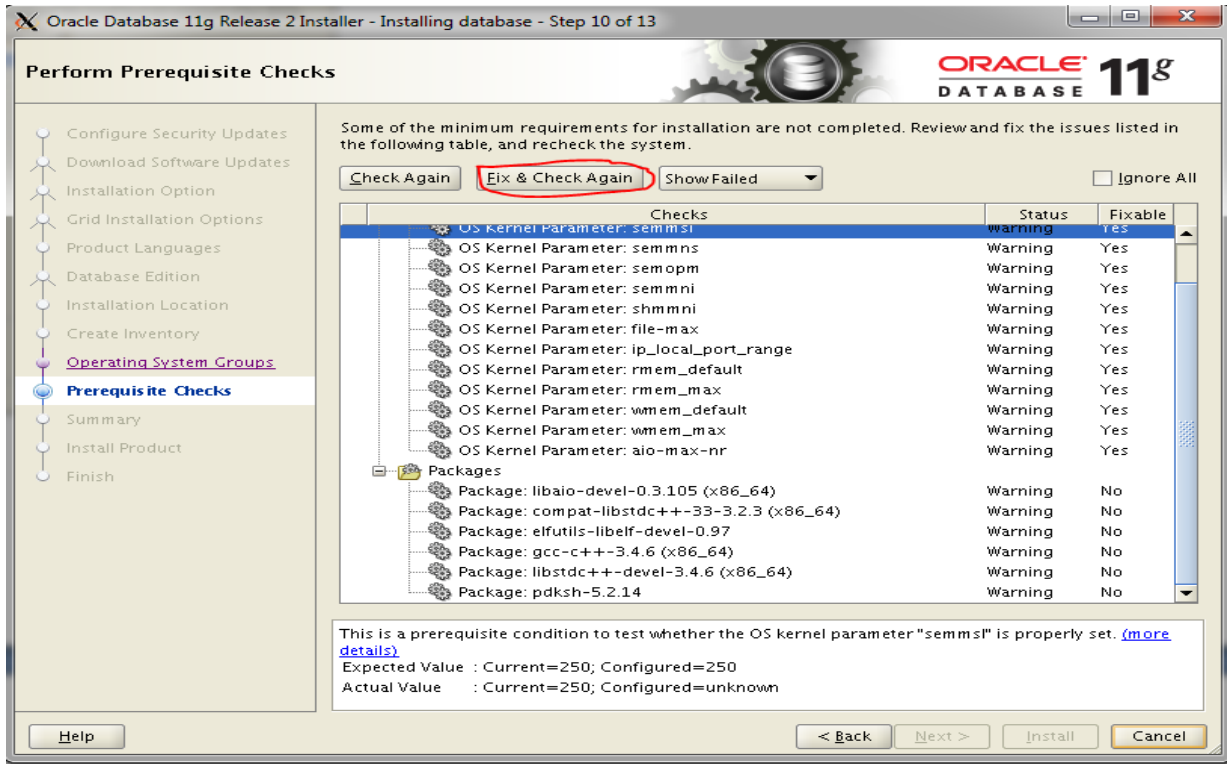
Növbəti şəkildə susmaya görə qalır heçnəyə toxunmuruq və **Next** düyməsinə sıxırıq:



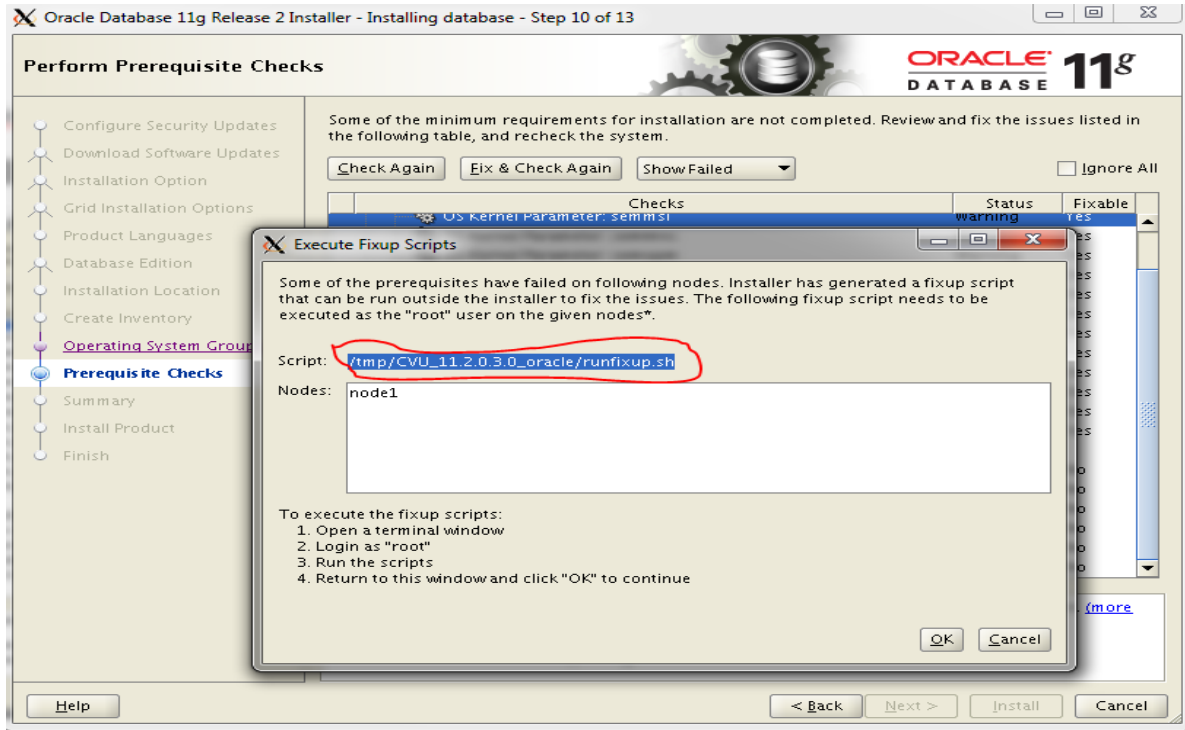
Növbəti şəkildə 'Database Administrator Group: DBA' və 'Database Operator Group: oinstall' seçirik və Next düyməsinə sıxırıq.



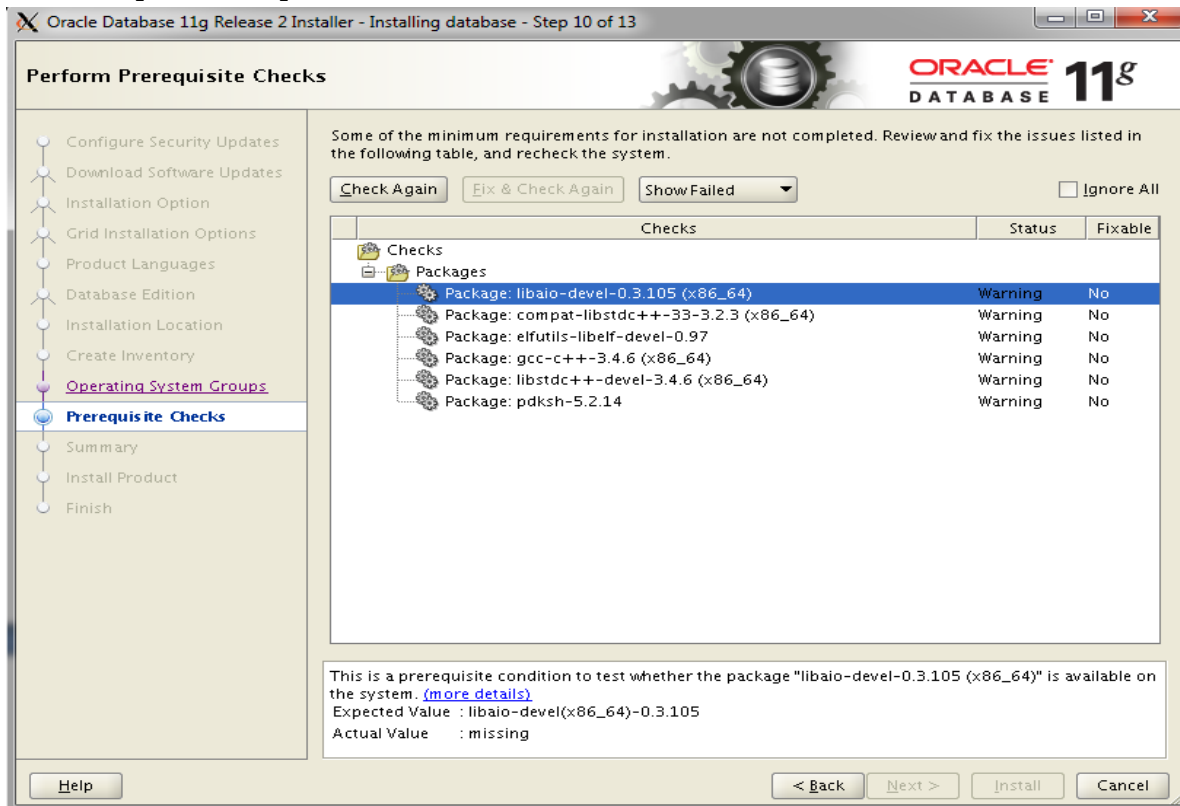
Sonra 'Fix and Check again' düyməsinə sıxırıq. Şəkildə Oracle-a tələb edilən paketlərin və kernel parametrlərin siyahısı çap edilir:



Oracle özü şəkildə göstərilən ünvanda script yaradacaq. Onu 'root' istifadəçi adından işə salırıq '/tmp/CVU_11.2.0.3.0_oracle/runfixup.sh'. Sonra işə ok düyməsinə sıxırıq.



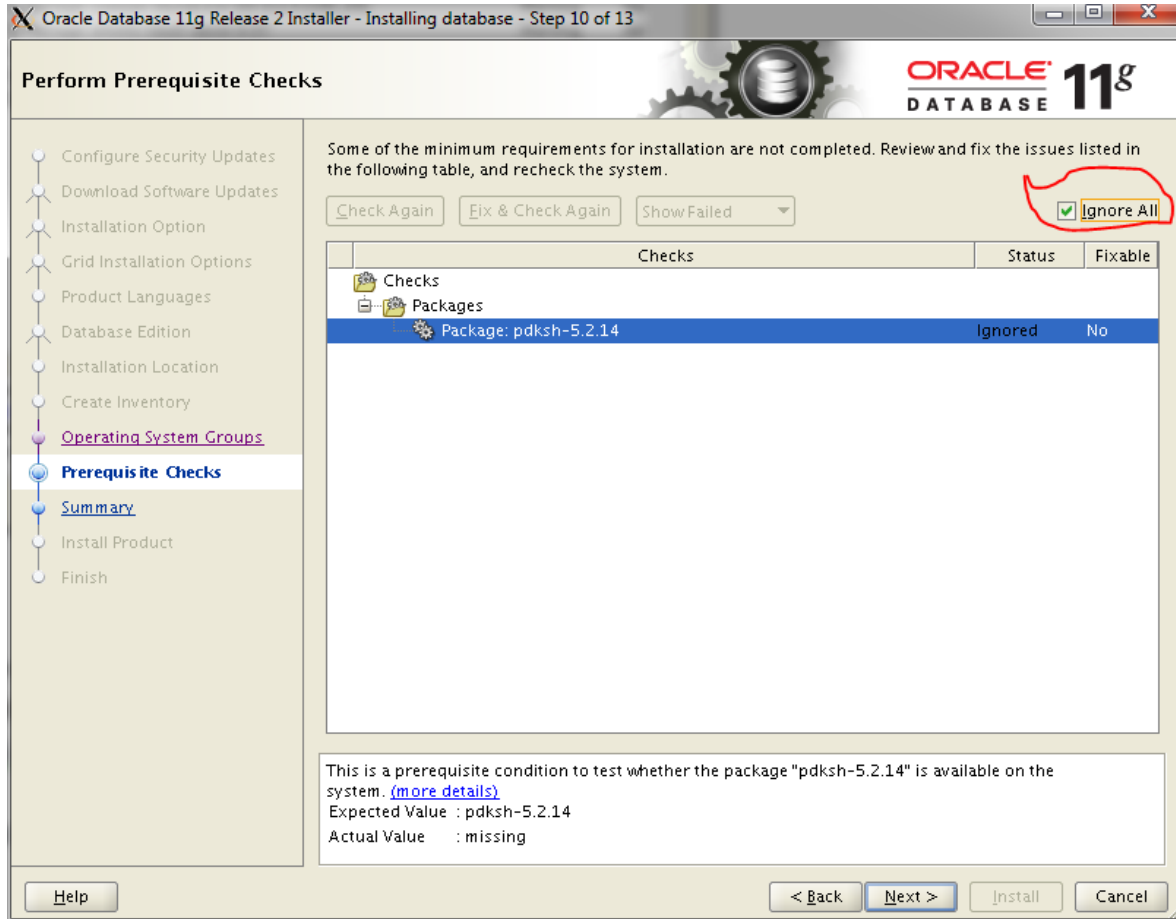
Şekildə göstərilən paketləri isə mütləq əlimizlə **root** istifadəçi adından serverə yükləməliyik.



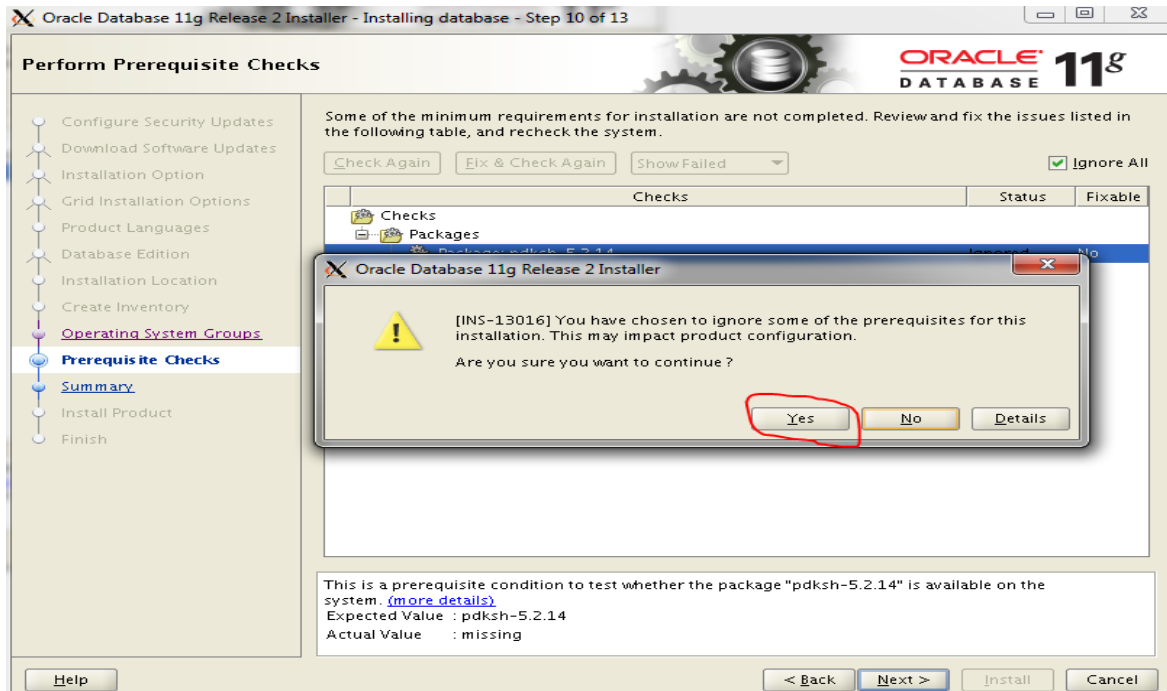
```
# yum -y install libaio-devel.x86_64 \  
\  
compat-libstdc++-33.x86_64 \  
elfutils-libelf-devel.x86_64 \  
gcc-c++.x86_64 \  
libstdc++-devel.x86_64 \  
mksh.x86_64
```

Tələb edilən paketləri
yükləyirik. Və sonra 'Check Again'
düyməsinə sıxırıq.

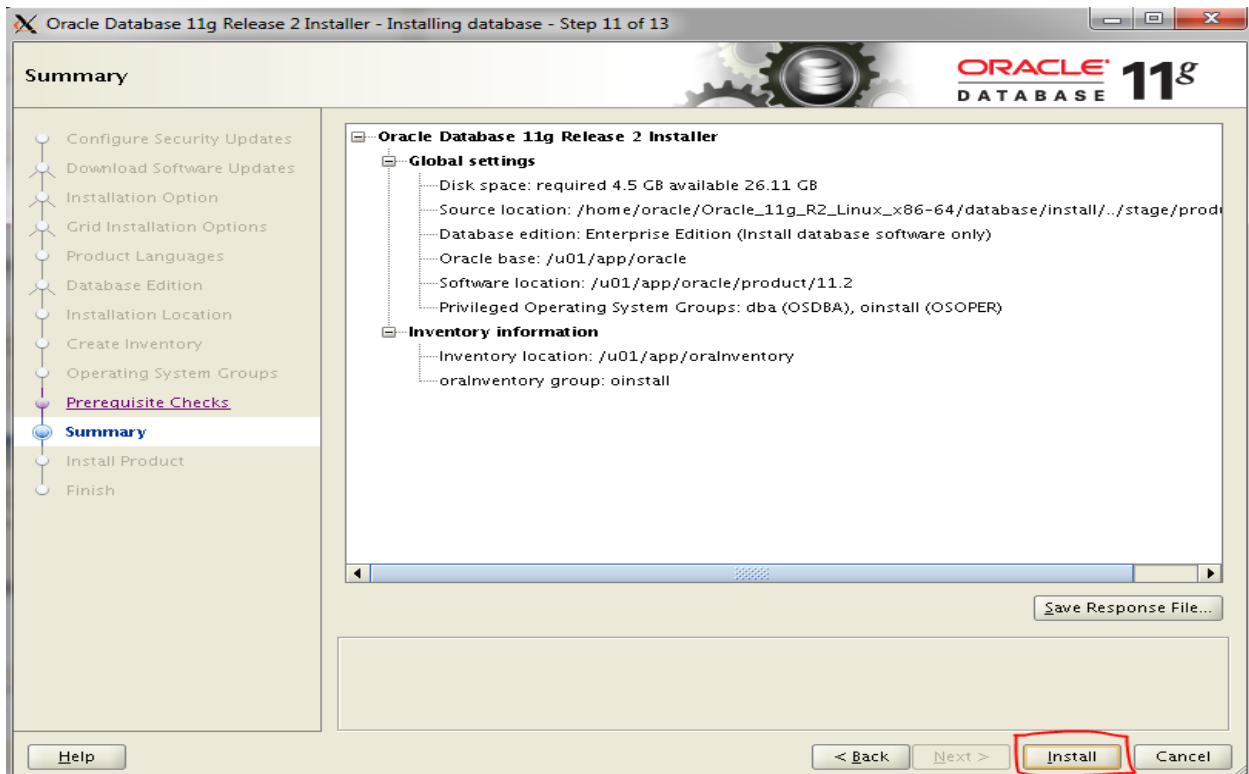
Çıxan səhifədə PDKSH üçün 'Ignore all' selectorunu seçərək Next düyməsini
sıxırıq.



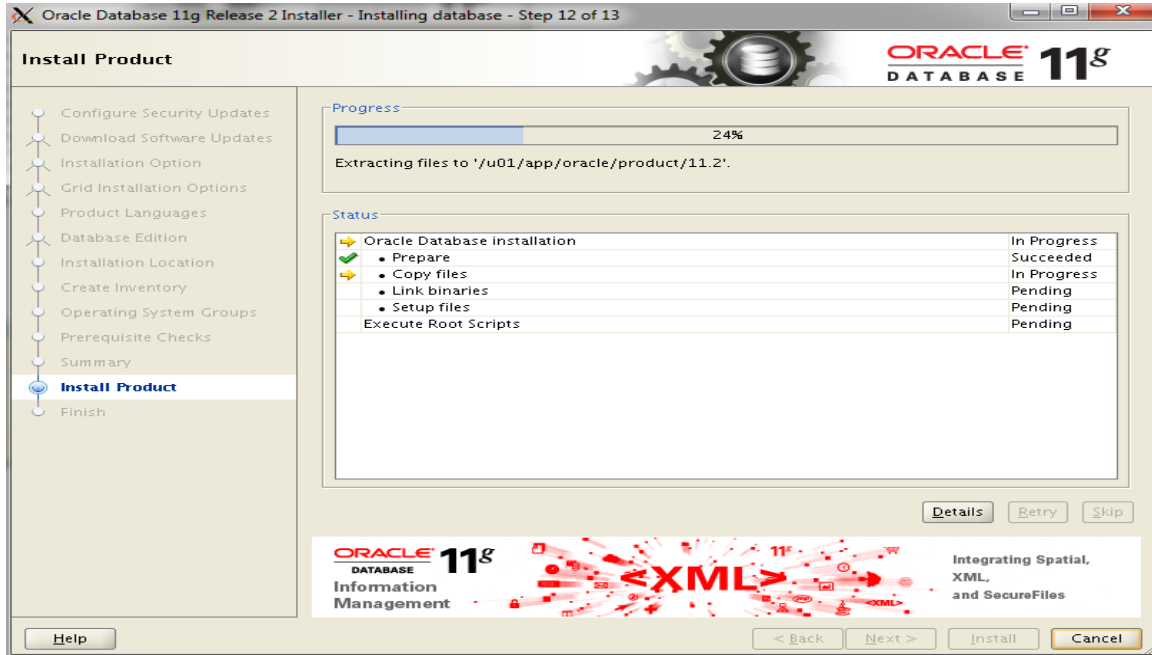
Çıxan şəkildə 'Yes' düyməsini sıxırıq.



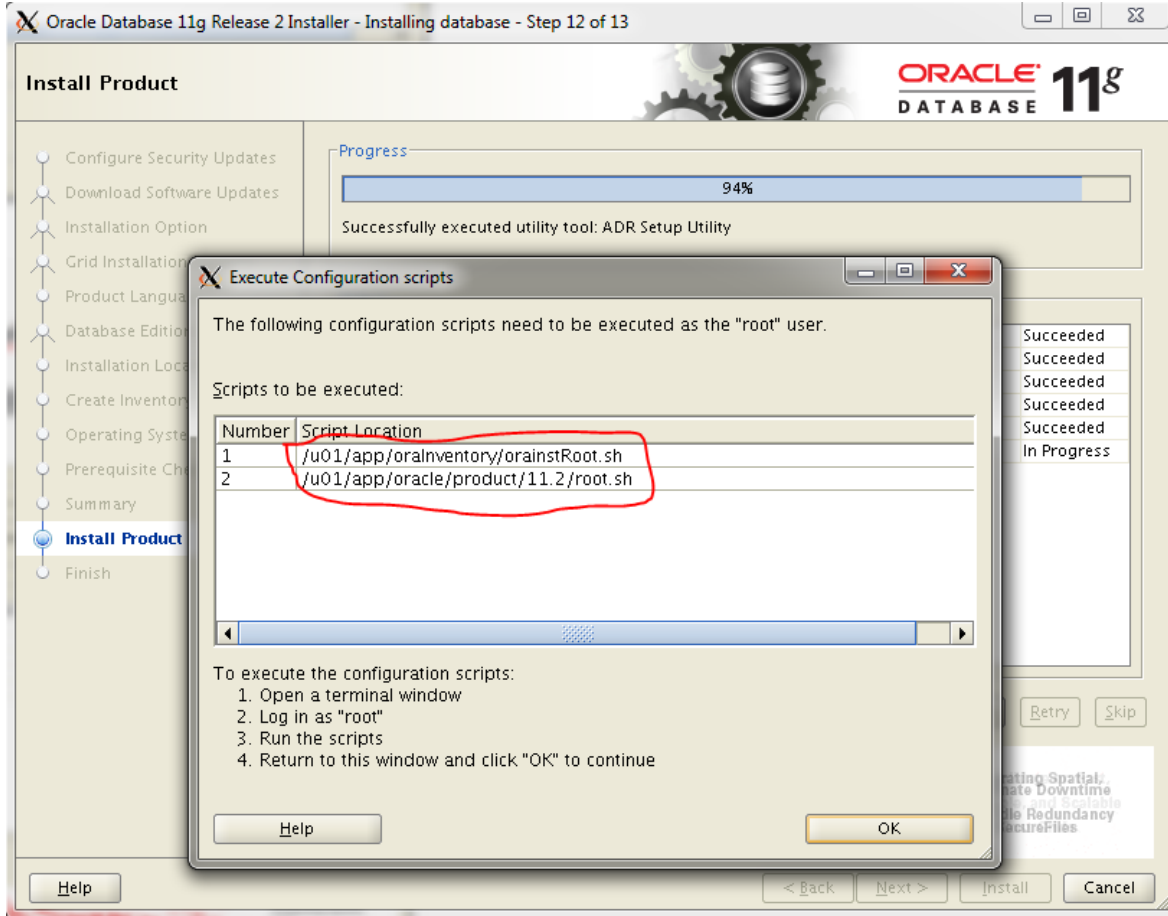
Nəhayət ki, sonda verilənlərə razılaşıb 'install' düyməsini sıxırıq.



Yüklənmə şəkilindəki kimi gedəcək.



Şəkilə çıxan scriptləri 'root' istifadəçi adından işə salmaq və sonra OK düyməsi sıxılmaq lazımdır.



```
[root@node1 ~]# /u01/app/oraInventory/orainstRoot.sh # Neticə aŝađıdaki kimi olmalıdır.
```

```
Changing permissions of /u01/app/oraInventory.  
Adding read,write permissions for group.  
Removing read,write,execute permissions for world.  
Changing groupname of /u01/app/oraInventory to oinstall.  
The execution of the script is complete.
```

```
[root@node1 ~]# /u01/app/oracle/product/11.2/root.sh # Neticə aŝađıdaki kimi olmalıdır.
```

```
Performing root user operation for Oracle 11g  
The following environment variables are set as:  
ORACLE_OWNER= oracle  
ORACLE_HOME= /u01/app/oracle/product/11.2  
Enter the full pathname of the local bin directory: [/usr/local/bin]:  
  Copying dbhome to /usr/local/bin ...  
  Copying oraenv to /usr/local/bin ...  
  Copying coraenv to /usr/local/bin ...  
Creating /etc/oratab file...  
Entries will be added to the /etc/oratab file as needed by  
Database Configuration Assistant when a database is created  
Finished running generic part of root script.  
Now product-specific root actions will be performed.  
Finished product-specific root actions.
```

Ən sonda ŝekildəki 'Close' düyməsi sıxılır.



DRBD-ni yükləyirik. (Hər iki node-da edirik)

Bunun üçün öncə DRBD üçün anbarları hər iki serverimizə yükləməliyik.

```
# vi /root/drbd-repos.sh # İşimizi avtomatlaşdıraraq.
# wget http://elrepo.org/elrepo-release-6-4.el6.elrepo.noarch.rpm
# rpm -Uvh elrepo-release-6-4.el6.elrepo.noarch.rpm
# yum update
# yum search drbd

# chmod +x /root/drbd-repos.sh # Yerinə yetirilmə yetkisi veririk ki,
#                               bir başa işimizi həll edək.
# ./root/drbd-repos.sh
# yum install drbd84-utils kmod-drbd84 # Hər iki node-da DRBD-ni
#                                       yükləyirik.

# yum install corosync # Hər node üçün Corosync yükləyirik.

# yum install pacemaker # Hər iki node üçün PaceMaker-i yükləyirik.
```

Qeyd: Artıq CentOS və RedHAT pacemakeri əsaslı istifadəsinə başladığına görə orda CRM əmri olmayacaq. Ona görə də onu ayrıca yükləmək lazımdır.

Aşağıdakı əmrləri hər iki node-da daxil edirik.

```
# wget -P /etc/yum.repos.d/
http://download.opensuse.org/repositories/network:/ha-clustering/RedHat_RHEL-
6/network:ha-clustering.repo
# yum install crmsh.x86_64
```

Hər iki serverdə DRBD üçün nəzərdə tutulan disk fiziki disk olaraq yaradıb onu Volume Group-a əlavə edib sonra Logical Volume yaratmaq lazımdır. Diqqətimizdə olsun ki, bizim halda hər iki serverdə **20GB** həcmində **'/dev/sdb'** disk ikinci diskdir hansı ki, DRBD synxronizasiyası üçün istifadə ediləcək.

Aşağıdakı addımları hər iki serverdə edirik.

```
# fdisk -l | grep Disk | grep sd | awk '{ print $2 }' # Bütün disklərimizi bu
#                                                       əmrlə çap edirik və
#                                                       yeni olanı görürük.
#                                                       Bizim halda '/dev/sdb'
#                                                       ən yeni olandır.

# pvcreate /dev/sdb # Fiziki disk yaradırıq.

# vgcreate oravg /dev/sdb # 'oravg' adlı volume group
#                           yaradırıq.
```

```
# lvcreate -l 100%FREE -n oravol oravg
```

```
# 'oravg' Volume Groupunda,  
'oravol' adlı Logical Volume  
yaradırıq və diskin tam  
həcmi ona veririk .
```

```
# vi /etc/drbd.conf
```

```
# Hər iki node-da global  
quraşdırma faylının qarşısına  
şərh yerləşdiririk.
```

```
#include "drbd.d/global_common.conf";  
include "drbd.d/*.res";
```

```
# Şərh yerləşdiririk  
# Deyirik ki, '/etc/drbd.d'  
qovluğunda olan 'res'  
genişlənməli istənilən faylda  
'drbd.conf' faylının bir  
hissəsidir.
```

```
# vi /etc/drbd.d/clusterdb_res.res
```

```
# Cluster bazası üçün  
'clusterdb_res.res' adlı yeni  
quraşdırma faylı yaradırıq və  
tərkibinə aşağıdakıları əlavə  
edirik.
```

```
resource clusterdb_res {  
protocol C;
```

```
handlers {  
    pri-on-incon-degr "/usr/lib/drbd/notify-pri-on-incon-degr.sh;  
    /usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger ;  
    reboot -f";  
    pri-lost-after-sb "/usr/lib/drbd/notify-pri-lost-after-sb.sh;  
    /usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger ;  
    reboot -f";  
    local-io-error "/usr/lib/drbd/notify-io-error.sh; /usr/lib/drbd/notify-  
    emergency-shutdown.sh; echo o > /proc/sysrq-trigger ; halt -f";  
    fence-peer "/usr/lib/drbd/crm-fence-peer.sh";  
}
```

```
startup {  
    degr-wfc-timeout 120;    # 2 minutes.  
    outdated-wfc-timeout 2; # 2 seconds.  
}
```

```
disk {  
    on-io-error detach;  
}
```

```
net {  
    cram-hmac-alg "sha1";  
    shared-secret "clusterdb";  
    after-sb-0pri disconnect;
```

```

    after-sb-1pri disconnect;
    after-sb-2pri disconnect;
    rr-conflict disconnect;
}

syncer {
    rate 10M;
    al-extents 257;
    on-no-data-accessible io-error;
}

on node1 {
    device      /dev/drbd0;
    disk        /dev/oravg/oravol;
    address     10.1.1.10:7788;
    flexible-meta-disk internal;
}

on node2 {
    device      /dev/drbd0;
    disk        /dev/oravg/oravol;
    address     10.1.1.20:7788;
    meta-disk internal;
}
}

```

Öncə yazdığımız quraşdırmada diqqətə alacağımız məqamları araşdıraq. Quraşdırmada olan '**clusterdb_res**' bizim resursun adıdır (Yeni sinxronizasiya ediləcək DISK). DRBD bu resursun üzərində öz işini təyin edir. 'Protocol C' (**Protocol C** - Daxili və Uzaq Disk bir-birlərinə uğurlu yazılan informasiya haqqında məlumat ötürür) isə iki node arasında olan disklərin sinxronizasiya metodudur.

node1 - 1-ci Node
node2 - 2-ci Node

/dev/drbd0 - Hər iki node-da DRBD üçün yaradılacaq virtual alət.
Hansı ki, sinxronizasiya onun üzərindən keçəcək.

/dev/oravg/oravol - Hər iki serverimizdə olan real disk.

10.1.1.10 - Node1 7788-ci port ilə danışır.

10.1.1.20 - Node2 7788-ci port ilə danışır.

meta-disk - Bu disk DRBD-nin özünün daxili məlumatlarının saxlanılması üçün istifadə elədiyi yerdədir. (**internal** - yeni **/dev/drbd0**-ın içində saxla)

olacaq. Ancaq unutmayın ki, əmri daxil etdikdən sonra **node1** və **node2** arasında disklərdə sinxronizasiya başlayacaq. Yalnız sinxronizasiya bitdikdən sonra işinizi davam etdirə bilərsiniz. Sinxronizasiya statusuna isə aşağıdakı əmrlə baxa bilərsiniz.

```
[root@node1 ~]# /etc/init.d/drbd status # Sinxronizasiya statusunu
faizlərlə göstərir.

drbd driver loaded OK; device status:
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
m:res          cs          ro          ds          p
mounted fstype
...           sync'ed:    9.0%          (18636/20472)M
0:clusterdb_res SyncSource Primary/Secondary UpToDate/Inconsistent C
```

```
[root@node1 ~]# cat /proc/drbd # Eynilə Sinxronizasiya
statusunu bu əmrlə də görə
bilərsiniz.

version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r---n-
ns:4549380 nr:0 dw:0 dr:4550296 al:0 bm:277 lo:0 pe:2 ua:1 ap:0 ep:1 wo:f
oos:16419164
[==>.....] sync'ed: 21.8% (16032/20472)M
finish: 0:02:37 speed: 103,992 (101,056) K/sec
```

Eynidə Node2 üçün də baxırıq.

```
[root@node2 ~]# /etc/init.d/drbd status # Node2-nin sinxronizasiya
statusuna baxırıq.

drbd driver loaded OK; device status:
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
m:res          cs          ro          ds          p
mounted fstype
...           sync'ed:    9.4%          (18556/20472)M
0:clusterdb_res SyncTarget Secondary/Primary Inconsistent/UpToDate C
```

```
[root@node2 ~]# cat /proc/drbd # Node2-nin sinxronizasiya
statusuna baxırıq.

version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
```

```
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
ns:0 nr:4646912 dw:4645888 dr:0 al:0 bm:283 lo:2 pe:5 ua:1 ap:0 ep:1 wo:f
oos:16320860
[==>.....] sync'ed: 22.2% (15936/20472)M
finish: 0:02:37 speed: 103,692 (100,996) want: 102,360 K/sec
```

Bitdikdən sonra informasiyanı ötürən node üçün status '**Primary/Secondary**' və **UpToDate** olmalıdır.

```
[root@node1 ~]# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
ns:20966748 nr:0 dw:0 dr:20967412 al:0 bm:1280 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f
oos:0
```

Bitdikdən sonra informasiyanı qəbul edən node üçün status '**Secondary/Primary**' və **UpToDate** olmalıdır.

```
[root@node2 ~]# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by dag@Build64R6,
2012-09-06 08:16:10
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
ns:0 nr:20966748 dw:20966748 dr:0 al:0 bm:1280 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f
oos:0
```

```
[root@node1 ~]# mkfs -t ext3 /dev/drbd0 # Sonra yalnız node1-də DRBD
diski format edirik ext3 file
sistem ilə.
```

Corosync/PaceMaker resurslarının quraşdırılması.

Artıq bizdə DRBD yüklənib və quraşdırılıb. Eynilə də Oracle yüklənib və DRBD diskdə quraşdırılıb. Həmçinin Cluster üçün lazımı paketlər yüklənilib. Ancaq onlar hələ quraşdırılmayıb.

```
[root@node1 ~]# vi /root/openais-exports.sh # OpenAIS üçün tələb edilən
quraşdırmalarımızı fayla
node1-də əlavə edirik.
Aşağıdakı sətirləri fayla
əlavə edirik.
export ais_mcast=226.99.1.1 # Multicast IP ünvanımız
export ais_port=4000 # Multicast IP-nin istifadə
edəcəyi port
export ais_addr=192.168.1.0 # Hansı şəbəkə üzərindən
Multicast işləyəcək.

# cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf
# Corosync example faylını
```

quraşdırma faylına
nüsxələyirik.

```
[root@node1 ~]# chmod +x /root/openais-exports.sh # Yaratdığımız fayla  
# yerinə yetirilmə  
# yetkisi veririk ki, işə  
# sala bilək.  
[root@node1 ~]# /root/openais-exports.sh # Faylı işə salırıq.
```

```
[root@node1 ~]# vi /etc/corosync/corosync.conf # Node1-də  
# Corosync-in  
# quraşdırma  
# faylına aşağıdakı  
# sətirləri əlavə  
# edirik.
```

```
# Please read the corosync.conf.5 manual page  
compatibility: whitetank
```

```
totem {  
    version: 2  
    secauth: off  
    threads: 0  
    interface {  
        ringnumber: 0  
        bindnetaddr: 192.168.1.0  
        mcastaddr: 226.99.1.1  
        mcastport: 4000  
        ttl: 1  
    }  
}  
logging {  
    fileline: off  
    to_stderr: no  
    to_logfile: yes  
    to_syslog: yes  
    logfile: /var/log/cluster/corosync.log  
    debug: off  
    timestamp: on  
    logger_subsys {  
        subsys: AMF  
        debug: off  
    }  
}  
amf {  
    mode: disabled  
}
```

Corosync-ə deyək ki, PaceMaker Cluster Resurs Manageri yükləsin.

```
[root@node1 /]# vi /etc/corosync/service.d/pcmk # Faylın içinə  
# aşağıdakıları əlavə  
# edirik.
```

```
service {  
# Load the Pacemaker Cluster Resource Manager
```

```
name: pacemaker
ver: 1
}
```

İndi isə eyni quraşdırmaları ardıcıl olaraq node2 üçün edirik.

```
[root@node1 ~]# scp /root/openais-exports.sh node2:/root/ # OpenAIS-in
export üçün
yaratdığımız scripti
node2-yə nüsxələyirik.
```

```
[root@node2 ~]# /root/openais-exports.sh # node2-də scripti işə
salırıq.
```

Qeyd: Bu işi mütləq Corosync fayllarında dəyişiklik eləməzdən öncə görmək lazımdır.

```
[root@node1 ~]# scp /etc/corosync/corosync.conf
node2:/etc/corosync/corosync.conf # Sonra 'corosync.conf'
faylıni node1-dən
node2-yə nüsxələyirik
```

```
[root@node1 ~]# scp /etc/corosync/service.d/pcmk
node2:/etc/corosync/service.d # Cluster Manager faylıni da
node1-dən node2-yə
nüsxələyirik.
```

```
# /etc/init.d/corosync start # Hər iki Node-da Corosync-i
işə salırıq.
```

```
# /etc/init.d/corosync status # Hər iki Node-da Corosync-in
statusuna baxırıq.
```

```
# /etc/init.d/pacemaker start # Hər iki Node-da PaceMaker-i işə
salırıq.
```

```
# /etc/init.d/pacemaker status # Hər iki Node-da PaceMaker-in
statusuna baxırıq. Səhvlər
'/var/log/messages' faylına
yığılır.
```

```
[root@node1 ~]# crm_mon -1 # Node1-in Cluster statusuna
baxırıq.
```

=====

Last updated: Tue Feb 12 22:23:58 2013

Last change: Tue Feb 12 22:13:53 2013 via crmd on node1

Stack: openais

Current DC: node2 - partition with quorum


```
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
0 Resources configured.
```

```
=====
```

```
Online: [ node1 node2 ]
```

```
[root@node2 ~]# crm_mon -1 # Node2-də də həmçinin
cluster statusuna baxırıq.
```

```
=====
```

```
Last updated: Tue Feb 12 22:25:48 2013
Last change: Tue Feb 12 22:13:53 2013 via crmd on node1
Stack: openais
Current DC: node2 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
0 Resources configured.
```

```
=====
```

```
Online: [ node1 node2 ]
```

Biz Clusteri iki node-dan ibarət qurduğumuz üçün Əgər Node-un biri digərini görməzsə o halda ciddi problemlər çıxır və Cluster dəqiq qərar vermir ki, Node-da problem çıxıb onu ayırsın ya yox? Və ya Node PaceMaker tərəfindən söndürülməlidirmi? Məhz bunun üçün də birazdan digər bir Router-ə ping edərək resurs yaradacağıq ki, Cluster həmin ping üçün qərar versin.

```
[root@node1 ~]# crm configure property no-quorum-policy=ignore
# Nodel-də PaceMaker quraşdırmalarını
edirik və avtomatik node2-yə gedir.
```

Pacemaker '**resource stickiness**' parametrini resursun node-lar arasında nə zaman miqrasiya ediləcəyinə qərar verir. Dəqiq məna o dərəcədə önəmli deyil. Bizim halda **100**-dür.

```
[root@node1 ~]# crm configure rsc_defaults resource-stickiness=100
# Nodel-də təyin edirik avtomatik 2-yə
gedir.
```

STONITH (Shoot The Other Node In The Head) - Digər halda isə **fencing** kimi adlanır. STONITH(fencing) NODE-un birində problem görərsə, digərinə müraciət edərək problemləli olan NODE-u öldürməyə çalışır. Bu riski aradan qaldırmaq üçün **split-brain** ssenarisini söndürürük.

```
[root@node1 ~]# crm configure property stonith-enabled=false # Split-Brain
(Stonith )
söndürürük
```

Quraşdırma ediləcək ilk resursumuz (**primitive p_drbd_oracle**) DRBD-dir ancaq, onu öncə konfiqruasiya etdiyimiz üçün o servisdə UP-dır. Bu səbədən də onu hər iki Node-da dayandırırıq.

```
[root@node1 ~]# /etc/init.d/drbd stop # Node1 üçün dayandırırıq.
[root@node2 ~]# /etc/init.d/drbd stop # Node2 üçün dayandırırıq.
[root@node1 ~]# crm configure # Node1-də Clusterin
quraşdırma rejiminə daxil
oluruq.
```

Cluster quraşdırma rejimində DRBD-i üçün **p_drbd_oracle** adlı **Cluster resurs** əlavə edirik hansı ki, **'cluster_res'** adlı DRBD diskini **15** saniyə intervalı ilə monitoring edir. Çıxan **WARNING**-lər ciddi bir şey deyil.

```
crm(live)configure# primitive p_drbd_oracle ocf:linbit:drbd params
drbd_resource="clusterdb_res" op monitor interval="15s"
WARNING: p_drbd_oracle: default timeout 20s for start is smaller than the
advised 240
WARNING: p_drbd_oracle: default timeout 20s for stop is smaller than the
advised 100
WARNING: p_drbd_oracle: action monitor not advertised in meta-data, it may
not be supported by the RA
```

'ms_drbd_oracle' adlı **master-slave** əlaqəsi təyin edir ki, istənilən halda **'p_drbd_oracle'** primitivinin 1 masteri-i ola bilər.

```
crm(live)configure# ms ms_drbd_oracle p_drbd_oracle meta master-max="1"
master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
```

Növbəti **'p_fs_oracle'** adlı yaradılan **primitive**-i isə DRBD alətin üstündə işləyən file sistemin təyin edilməsi üçün istifadə edilir və o **'/oradisk'** qovluğuna (Hansı ki, **Oracle** üçün nəzərdə tutulub) **'ext3'** fayl sistem tipində mount ediləcək. Hər iki node-da **'/oradisk'** adlı qovluq yaratmağı unutmayın.

```
crm(live)configure# primitive p_fs_oracle ocf:heartbeat:Filesystem params
device="/dev/drbd0" directory="/oradisk" fstype="ext3"
WARNING: p_fs_oracle: default timeout 20s for start is smaller than the
advised 60
WARNING: p_fs_oracle: default timeout 20s for stop is smaller than the
advised 60
```

Çəkdiyim Şəkildə Göründüyü kimi istifadəçilər Oracle-a **VIP (Virtual IP Address)**-lə qoşulacaq. Bizim halda o **172.16.1.100**-dür. Və mütləq dəqiq qeyd edin ki, hər iki node-da **172.16.1.0** şəbəkəsində olan **şəbəkə kartı** identik (eyni adla) olmalıdır. Bizim halda o hər iki node-da **'eth1'**-dir.

```
crm(live)configure# primitive p_ip_oracle ocf:heartbeat:IPaddr2 params
ip="172.16.1.100" cidr_netmask="24" nic="eth1"
```

İndi isə Oracle bazanın hansı ünvdan əmrlər istifadə edəcəyini **'home="/u01/app/oracle/product/11.2"'**, hansı SID-lə **'sid="oradb"'** çalışacağını və hansı istifadəçi adından **'user="oracle"'** işə düşəcəyini təyin edirik. Timeout-u böyük qoyuruq çünki Oracle böyük baza ilə işləyəndə gec qalxa və düşə bilər.

```
crm(live)configure# primitive ora_db ocf:heartbeat:oracle \
```

```
op monitor interval="120s" timeout="30s" \  
op start interval="0" timeout="7200" \  
op stop interval="0" timeout="7200" \  
params sid="oradb" home="/u01/app/oracle/product/11.2"  
user="oracle"
```

Artıq Oracle-ın Listenerini təyin edək ki, həmin IP ünvan üzərində işə düşsün. Biz həmin listener faylı birazdan quraşdıracağıq və həmin faylın içində bizim quraşdırdığımız VIP-ni yazacağıq(Yeni: **172.16.1.100**). Listenerin adı(**listener="LISTENER"**), hansı istifadəçi adından(**user="oracle"**), hansı sid adından işə düşəcək(**sid="LISTENER"**)

```
crm(live)configure# primitive ora_lsnr ocf:heartbeat:oralsnr \  
op monitor interval="120s" timeout="30s" \  
op start interval="0" timeout="120" \  
op stop interval="0" timeout="120" \  
params sid="LISTENER" home="/u01/app/oracle/product/11.2"  
user="oracle" listener="LISTENER"
```

'**g_oracle**' adlı resurs qrup yaradırıq ki, resurslar qrup halında bir node-dan digər noda ardıcılıqla miqrasiya edilsin. Öncə fayl sistem ardınca VIP və sonra OraListener sonda isə OracleDB servisi.

```
crm(live)configure# group g_oracle p_fs_oracle p_ip_oracle ora_lsnr ora_db
```

Oracle servis group DRBD master işləyən host-dan asılı olduğuna görə, bu əlaqə yerləşmənin təyinatı və düzgün məhdudlaşdırılmanın qurulması üçün əlavə edilir ki, Oracle group-un DRBD master-lə eyni ünvanda yerləşdiyini zəmanət altına alsın və DRBD-nin master host üzərinə keçidi Oracle group-un işə düşməsindən əvvəl yerinə yetirilməlidir.

```
crm(live)configure# colocation c_oracle_on_drbd inf: g_oracle  
ms_drbd_oracle:Master
```

```
crm(live)configure# order o_drbd_before_oracle inf: ms_drbd_oracle:promote  
g_oracle:start
```

Artıq biz demək olar ki, bütün quraşdırmalarımızı etmişik yalnız yadda saxlamaq qalıb. Və '**commit**' əmri ilə biz onu edirik.

```
crm(live)configure# commit
```

```
crm(live)configure# quit # Sonra Cluster Console-undan çıxırıq
```

Split-Brain ssenarisini yaratmaq üçün isə PaceMaker aidiyyatı olmayan şəbəkə alətlərinə **ping** edərək istifadə edə bilər ki, O görünməyən halda tez təyin etsin ki, həmin hostun DRBD Master olmasının qarşısını alsın və onu izolyasiya eləsin. İlk olaraq **p_ping** adlı resurs yaradırıq. **host_list** isə şəbəkənin yoxlanışı üçün IP ünvanların siyahısını təşkil edir. Nəzərə alın ki, resursun '**stickiness**' (hərəkətsizliyini) aşmaq üçün **multiplier** parametri ilə təyin edilən **host_list** çoxluğu **resource-stickiness** parametrindən çox olmalıdır hansı ki, biraz öndə Cluster node-ları yaradılarda təyin edilmişdi. (Susmaya görə bütün resurslar üçün təyin edilən stickiness **100** idi). Multiplier-in mənası yeni **1000** bizim pinglərimizin sayıdır. Həmin ötürülən

pinglərin sayına görə uğurlu sayılan pinglərin sayı ən azı **50%**-dən çox olmalıdır ki, Cluster işləsin. Əks halda əgər node-un birində Ping digərinə nisbətən **50%**-dən az olsa Cluster resursların hamısı digər node-a ötürüləcək.

```
[root@node1 ~]# crm configure
```

```
crm(live)configure# primitive p_ping ocf:pacemaker:ping params name="ping" \  
multiplier="1000" host_list="172.16.1.1" op monitor interval="15s"  
timeout="60s" start timeout="60s"  
WARNING: p_ping: default timeout 20s for start is smaller than the advised  
60WARNING
```

Cluster-də olan hər bir host ping işlətməlidir ki, özlərinin şəbəkələrini təyin edə bilsinlər və **clone (cl_ping)** sadəcə onunçün yaradılmışdır ki, bu resurs clusterin hər bir node-nda işə düşsün.

```
crm(live)configure# clone cl_ping p_ping meta interleave="true"
```

Artıq ping hər bir host üçün təyin edilmişdir və PaceMaker-ə demək lazımdır ki, o həmin pinglərin cavabını necə emal eləsin. Bu misalda işə yeni yerləşmə məhdudiyyətini (**l_drbd_master_on_ping**) **DRBD Master(ms_drbd_oracle** resursu)-in yerləşməsini hosta xüsusi etirazedicisi sonsuzluq işarəsi (**-inf**) təyin edərək idarə edəcək(əgər təyin edilmiş host-da heç bir ping cavab vermirsə və ya **<=0'0**-dan kiçiksə' ya da **PaceMaker** sintaksisi ilə **number:lte 0**)

```
crm(live)configure# location l_drbd_master_on_ping ms_drbd_oracle rule  
$role="Master" -inf: not_defined ping or ping number:lte 0
```

Və Clusterimizin daha üstün olan node-nu node1-ə təyin edirik.

```
crm(live)configure# location cli-prefer-g_oracle g_oracle rule inf: #uname eq  
node1
```

```
crm(live)configure# commit
```

```
WARNING: p_ping: default timeout 20s for start is smaller than the advised 60
```

Clusterimizin düzgün qaydada işə düşməsini monitoring edirik.

```
[root@node1 ~]# crm_mon
```

```
=====
```

```
Last updated: Thu Mar 14 15:05:19 2013
```

```
Last change: Thu Mar 14 13:47:28 2013 via crm_resource on node1
```

```
Stack: openais
```

```
Current DC: node2 - partition with quorum
```

```
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
```

```
2 Nodes configured, 2 expected votes
```

```
8 Resources configured.
```

```
=====
```

```
Online: [ node2 node1 ]
```

```
Master/Slave Set: ms_drbd_oracle [p_drbd_oracle]
```

```
Masters: [ node1 ]
Slaves: [ node2 ]
Clone Set: cl_ping [p_ping]
Started: [ node1 node2 ]
Resource Group: g_oracle
  p_fs_oracle      (ocf::heartbeat:Filesystem):    Started node1
  p_ip_oracle      (ocf::heartbeat:IPaddr2):          Started node1
  ora_lsnr         (ocf::heartbeat:oralsnr):          Started node1
  ora_db           (ocf::heartbeat:oracle):          Started node1
```

Servislerin yenidenyüklendikten sonra düzgün işlemesini idare edek. Her iki node için.

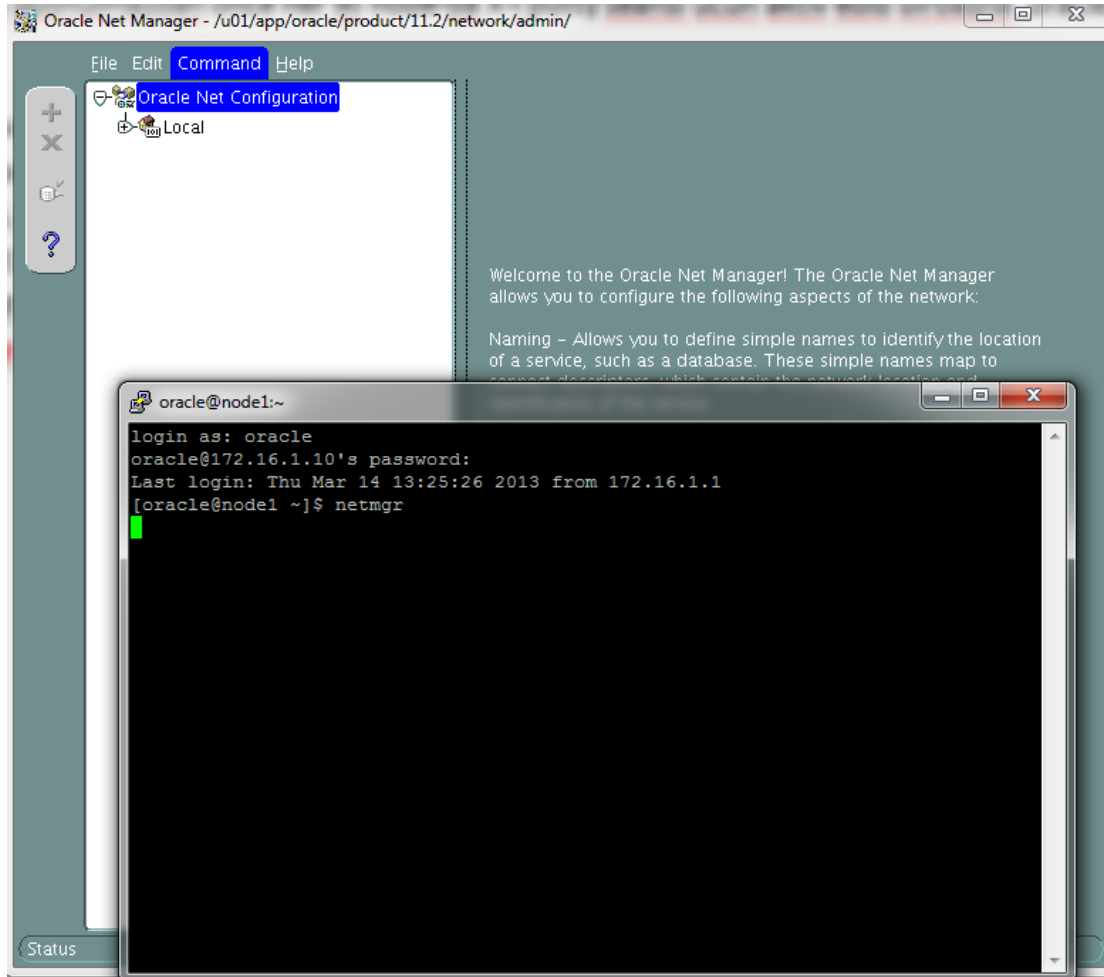
```
[root@node1 ~]# chkconfig drbd off
[root@node1 ~]# chkconfig corosync on
[root@node1 ~]# chkconfig pacemaker on
```

```
[root@node2 ~]# chkconfig drbd off
[root@node2 ~]# chkconfig corosync on
[root@node2 ~]# chkconfig pacemaker on
```

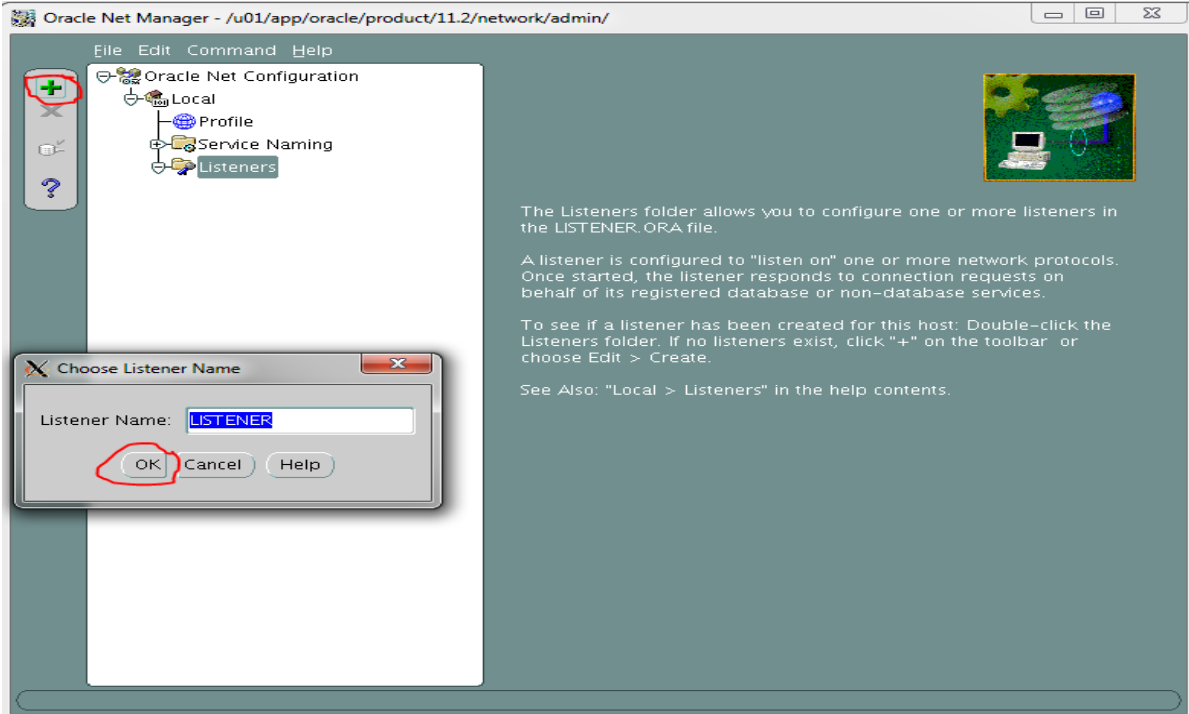
Sonda her iki node-a restart edib geçirik Oracle-ın Listenerini quraşdıraraq və bazanı qaldıraraq. Etdiyimiz quraşdırmalara görə reboot-dan sonra **'/oradisk'** adlı qovluq **'node1'**-də mount olunacaq. Oracle üçün Listeneri Static olaraq təyin edirik. node-lara qoşulmazdan öncə hər iki node-da X Forwarding-i putty seansı üçün aktiv edib birbaşa **'oracle'** istifadəçi ilə daxil olmaq lazımdır.

```
[root@node1 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg_node1-lv_root
36G  12G  23G  34% /
tmpfs            935M  24M  912M   3% /dev/shm
/dev/sda1        485M  141M  319M  31% /boot
/dev/drbd0      40G  177M  38G   1% /oradisk
```

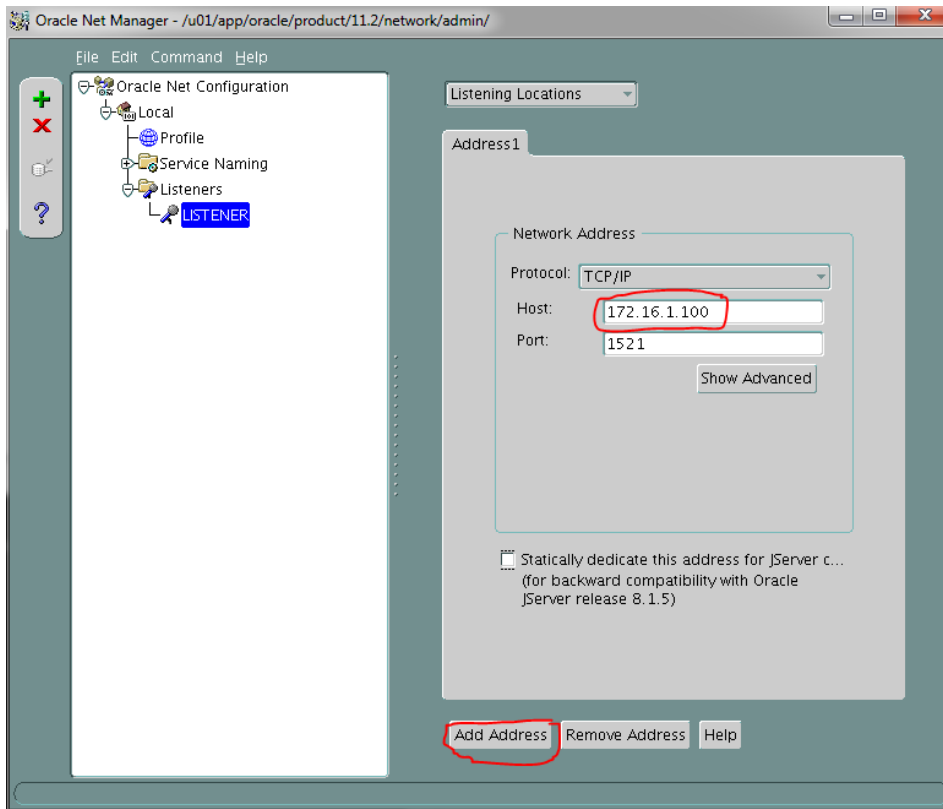
node1-də oracle istifadəçi ilə daxil olub **'netmgr'** əmrini işə salırıq. (Şəkildəki kimi qrafik formada ekran gələcək.)



Sonra **Local-> Listeners** və sonra yaşıl rəngli '+' simvolunu sıxıb **LISTENER-**imizin adını daxil edirik və 'OK' düyməsini sıxırıq. (Şəkildəki kimi)

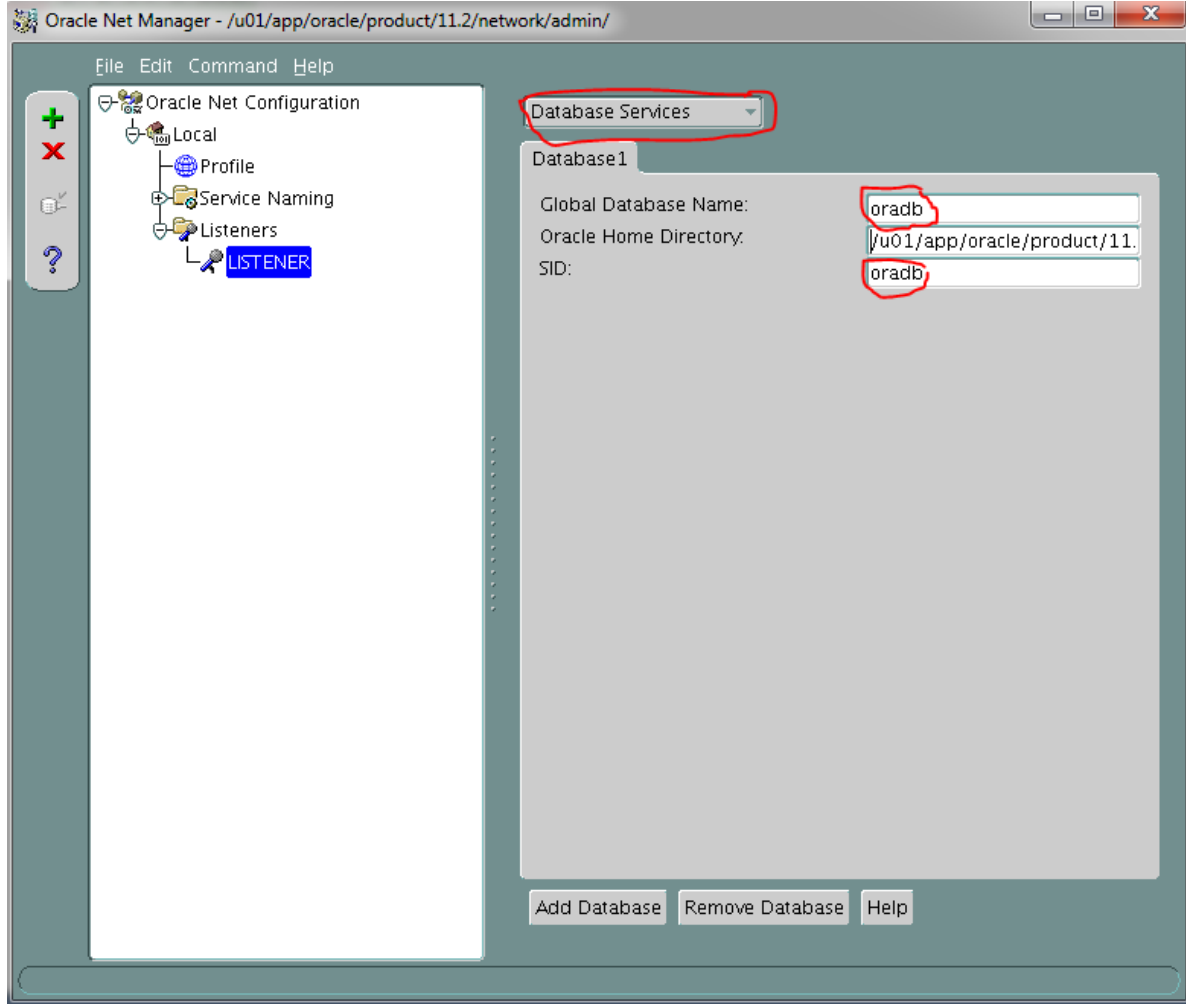


Sonra '**Add Address**' düyməsini sıxıb Clusterdə quraşdırdığımız VIP-ni daxil edirik, yeni **172.16.1.100**.



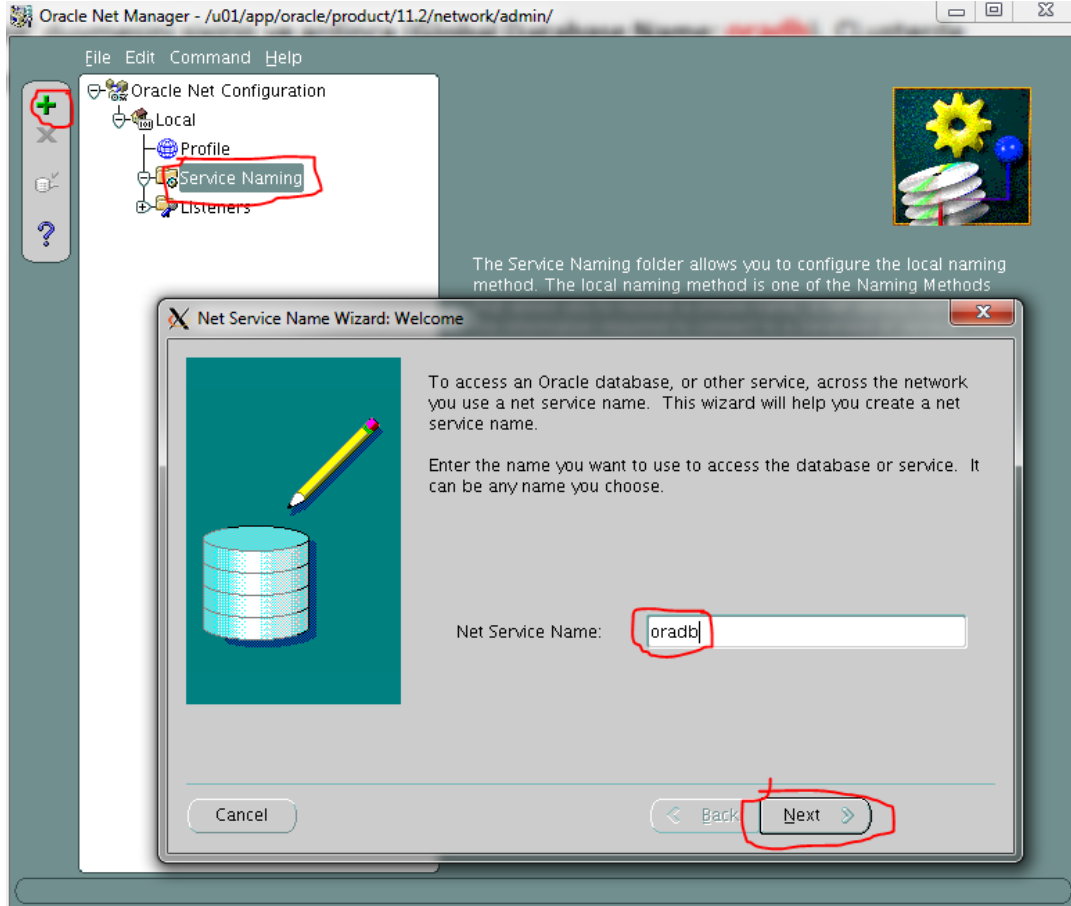
Sonra isə '**Listenin locations**' olan bölmədə '**Database Services**' seçib '**Add Database**' düyməsini sıxırıq və ardınca (**Global Database Name: oradb**),

Clusterdə göstərdiyimiz quraşdırma ünvanını susmaya görə olaraq dəyişmirik (**Oracle Home Directory**-ünvana toxunuruq) və Clusterdə təyin elədiyimiz **SID** (**oradb**).

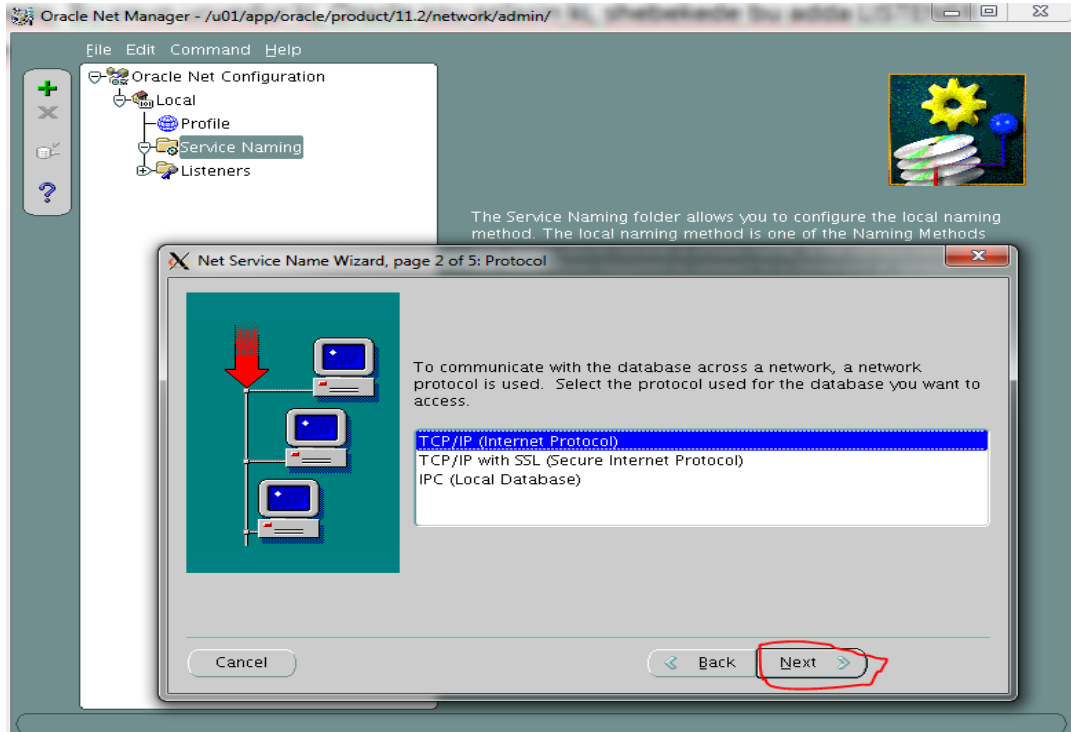


Sonra **File -> Save Network Configurations** edirik.

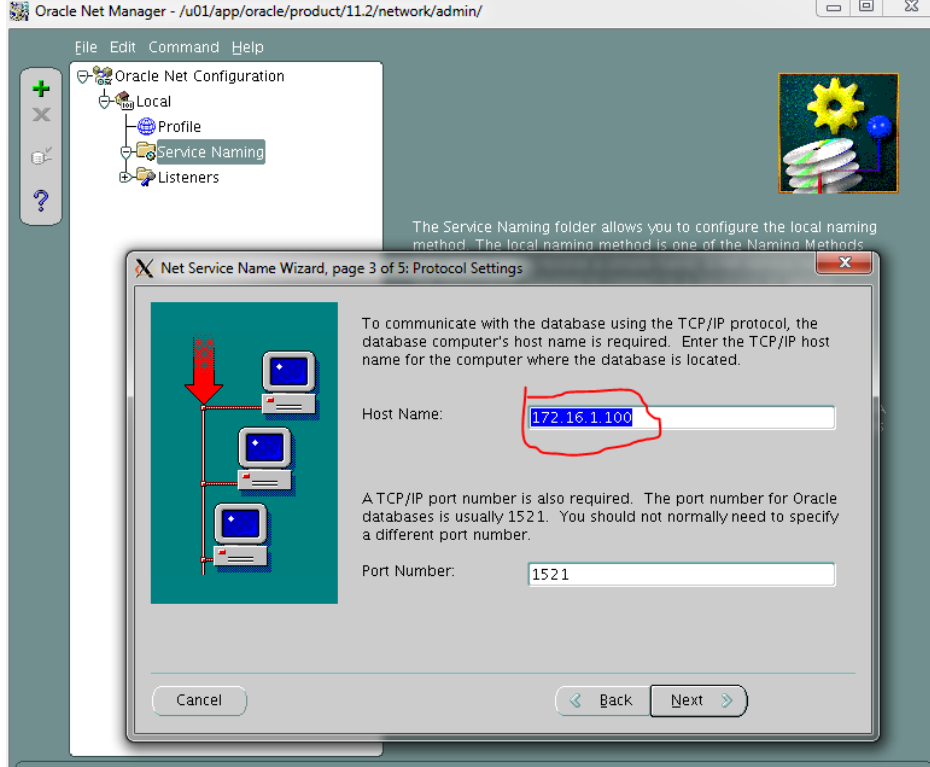
Sonra **Service Naming** -> '+' simvolunu daxil edib '**oradb**' yazıb **Next** düyməsini sıxırıq. Bu onun üçündür ki, Oracle özü anlasın ki, şəbəkədə bu adda LISTENER var. Yəni ki, bu adı Oracle-a qoşulan bütün inzibatçıların **tnsnames.ora** faylında yazmaq tələb olunur.



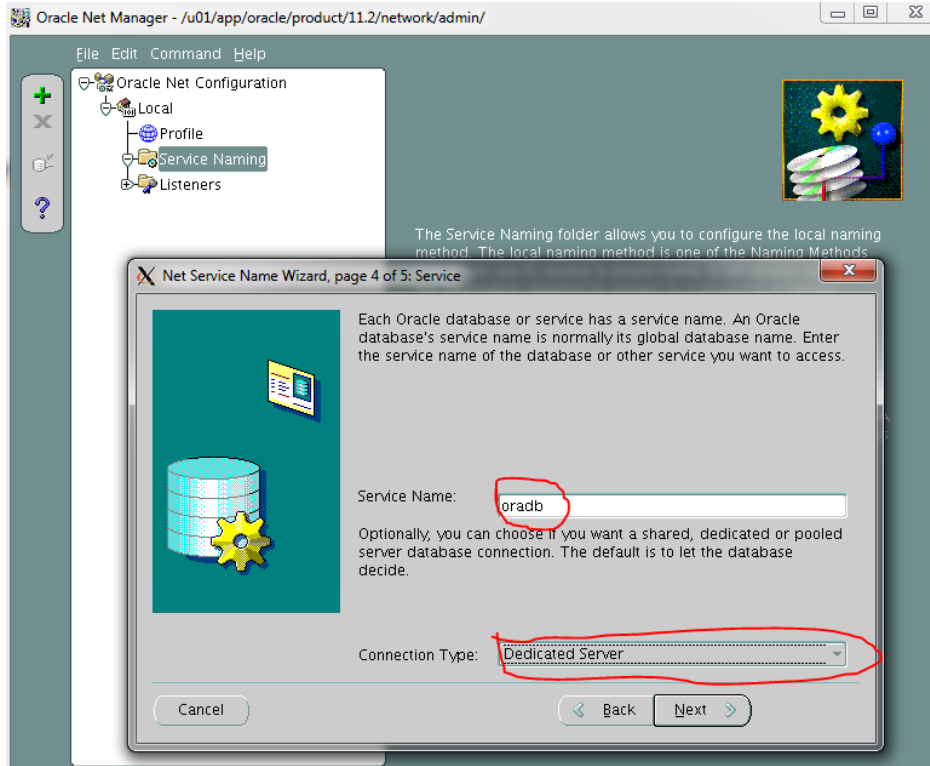
Sonra 'TCP/IP (Internet Protocol)' seçirik və Next düyməsini sıxırıq.



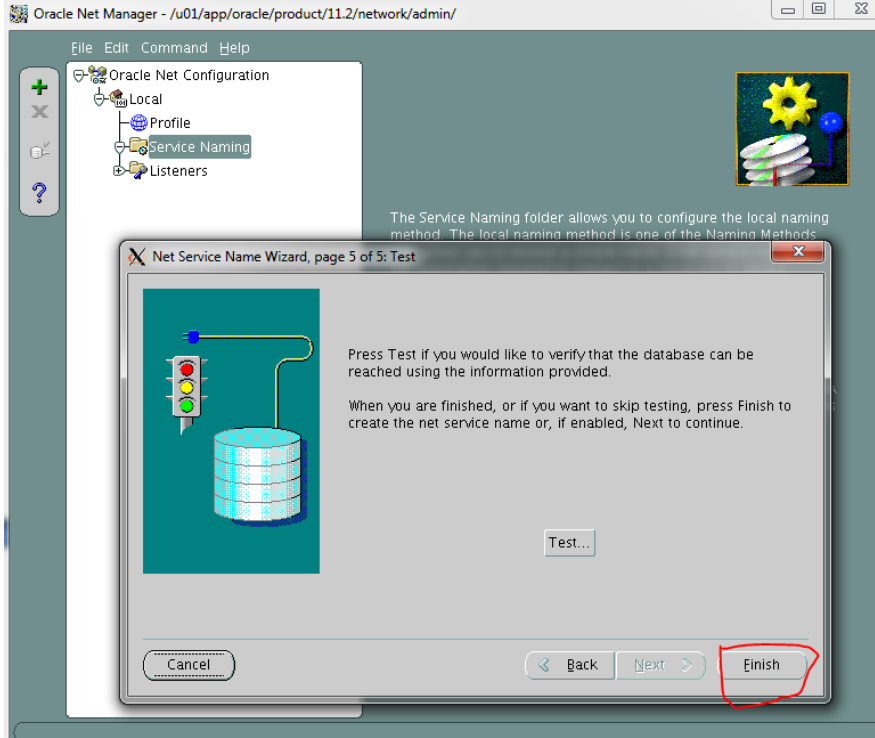
Sonra istifadəçilər qoşulacaq IP ünvan yeni VIP **172.16.1.100**-ü "Host Name" bölümündə yazırıq və **Next** düyməsinə sıxırıq.



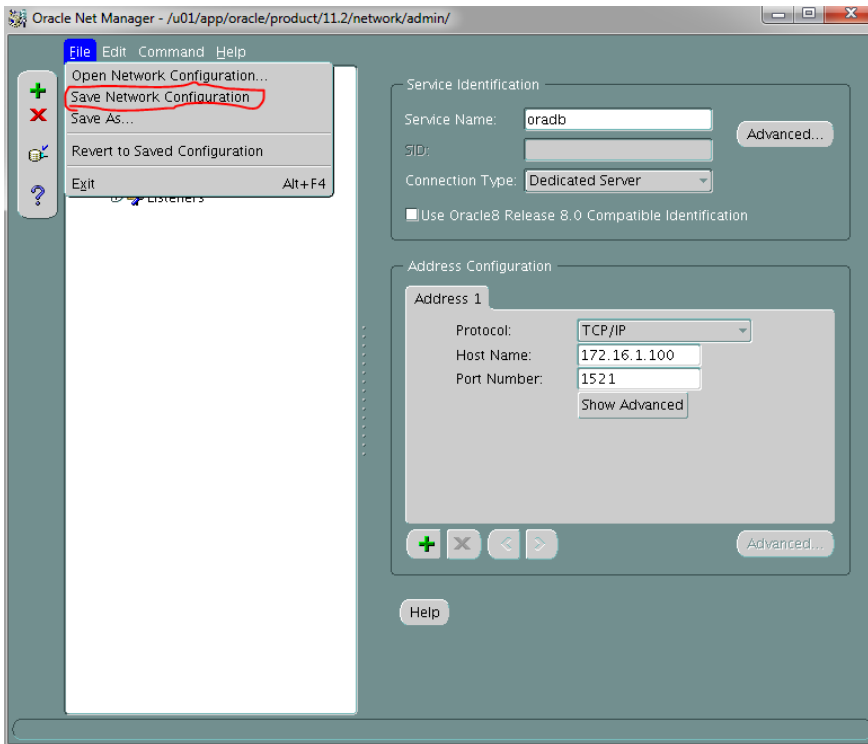
Sonra '**Service Name: oradb**', '**Connection Type: Dedicated Server**' (Yəni hər sessiya üçün ayrı process yaradılsın) edirik. Sonra **next** düyməsinə sıxırıq.



Sonda isə **Finish** düyməsini sıxırıq.



Sonra isə Yynidən **File -> Save Network Configuration** edirik. Sonra **File -> Exit** edib Çıxırıq.



```
# cd /u01/app/oracle/product/11.2/network/admin # 'oracle' istifadəçi adı ilə  
daxil olub bu ünvana keçirik. Və  
'tnsnames.ora' faylının sonuna  
aşağıdakı sətirləri əlavə edirik.
```

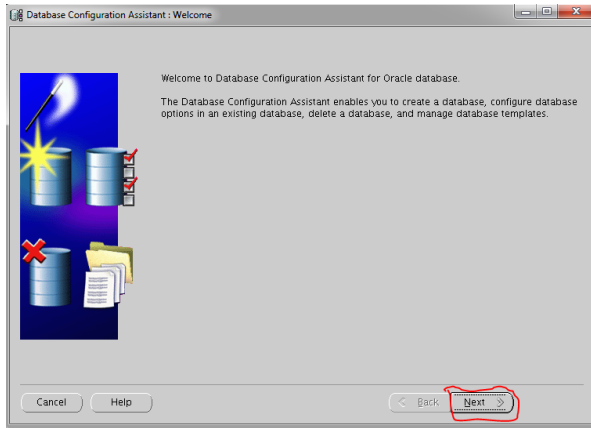
Bu Oracle-in inisializasiya parametrinə **local_listener** əlavə etmək üçündür.

```
# vi tnsnames.ora # Faylın sonuna aşağıdakı sətirləri əlavə edirik.  
LISTENER =  
  (DESCRIPTION =  
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST =  
172.16.1.100) (PORT = 1521)))  
    (SID = 'oradb')  
  )
```

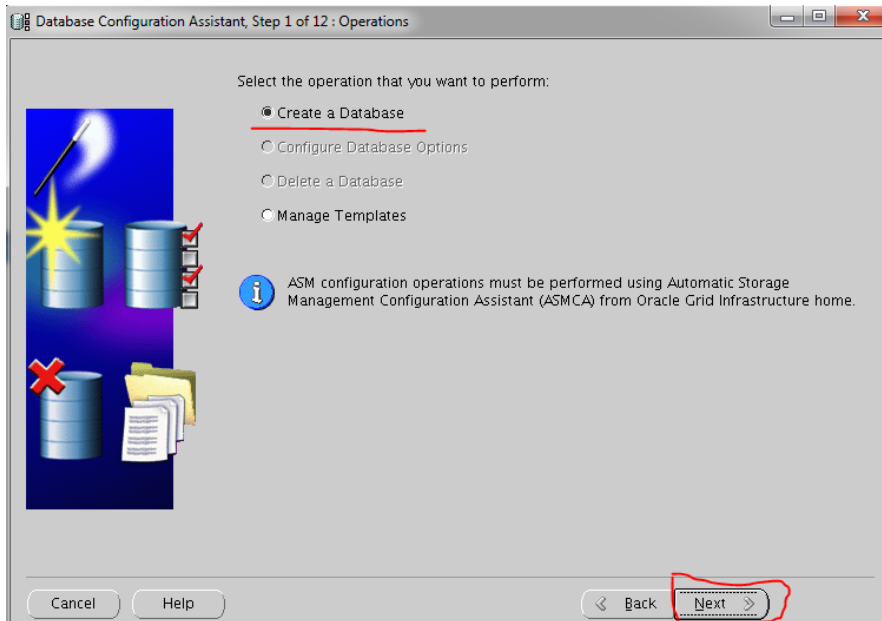
Artıq Bazanı yarada bilərik.

```
[oracle@node1 admin]$ dbca
```

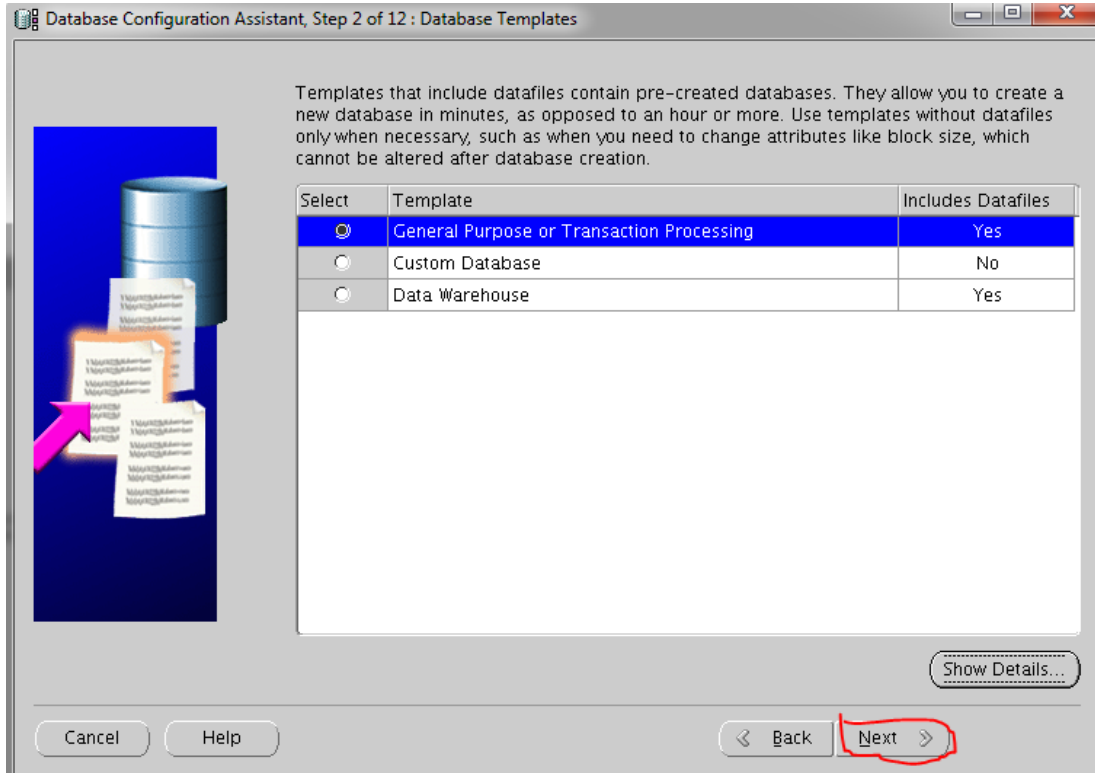
```
# Database Configuration Assistant əmrini  
node1-də daxil edirik. Aşağıdakı şəkil  
gələcək. Next düyməsinə sıxırıq.
```



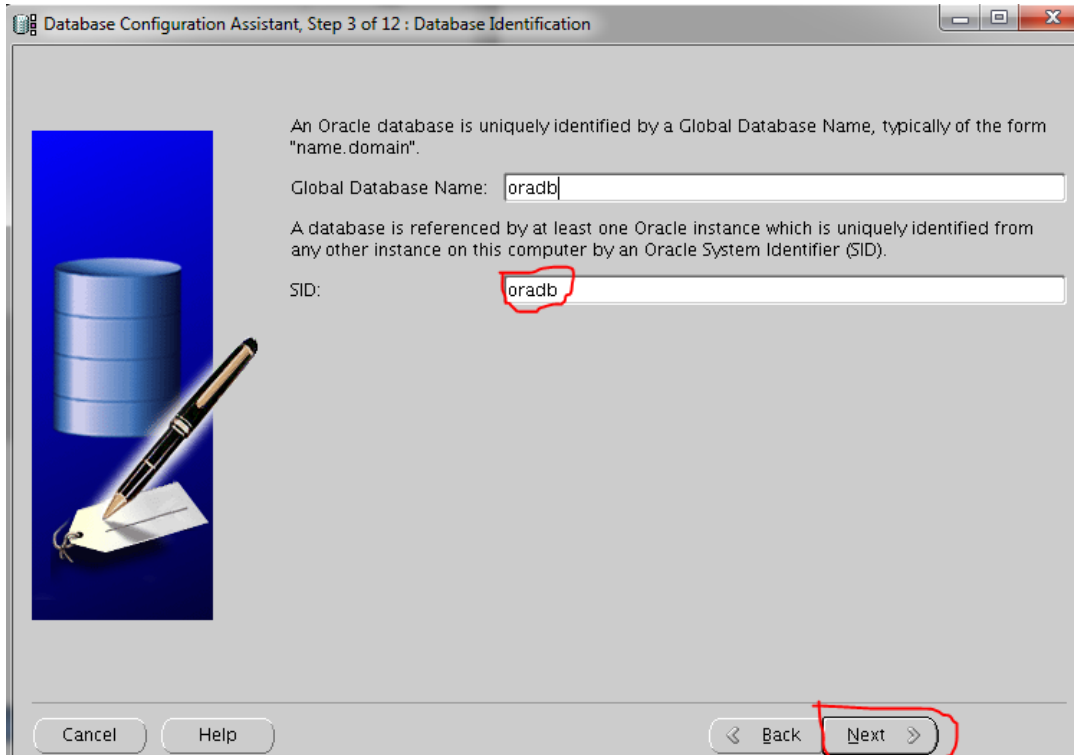
Create Database seçirik və **Next** düyməsinə sıxırıq.



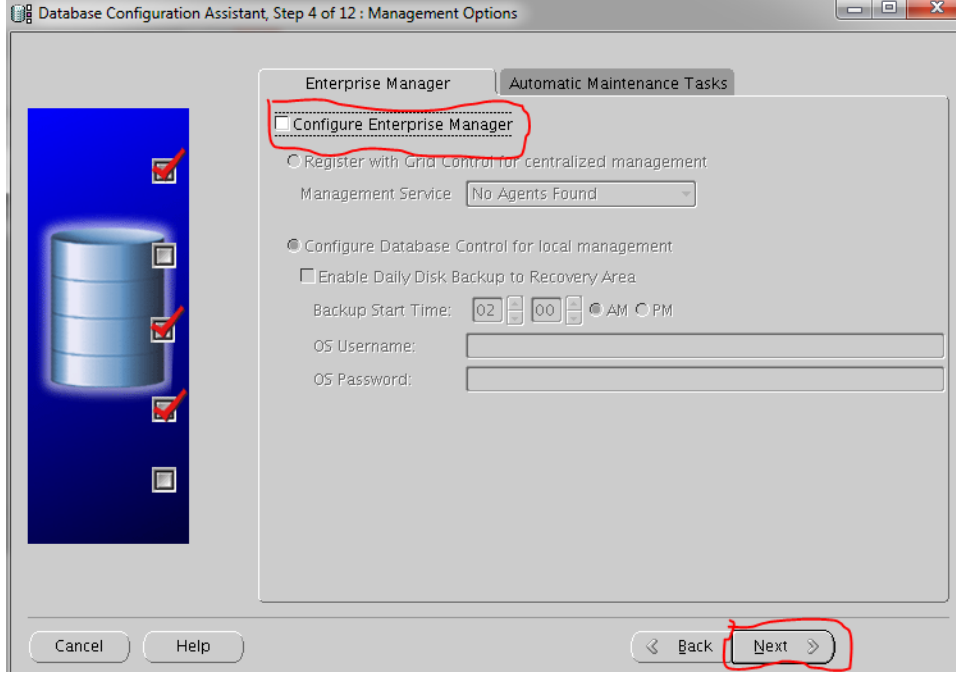
Sonra 'General Purpose or Transaction Processing' və Next düyməsinə sıxırıq



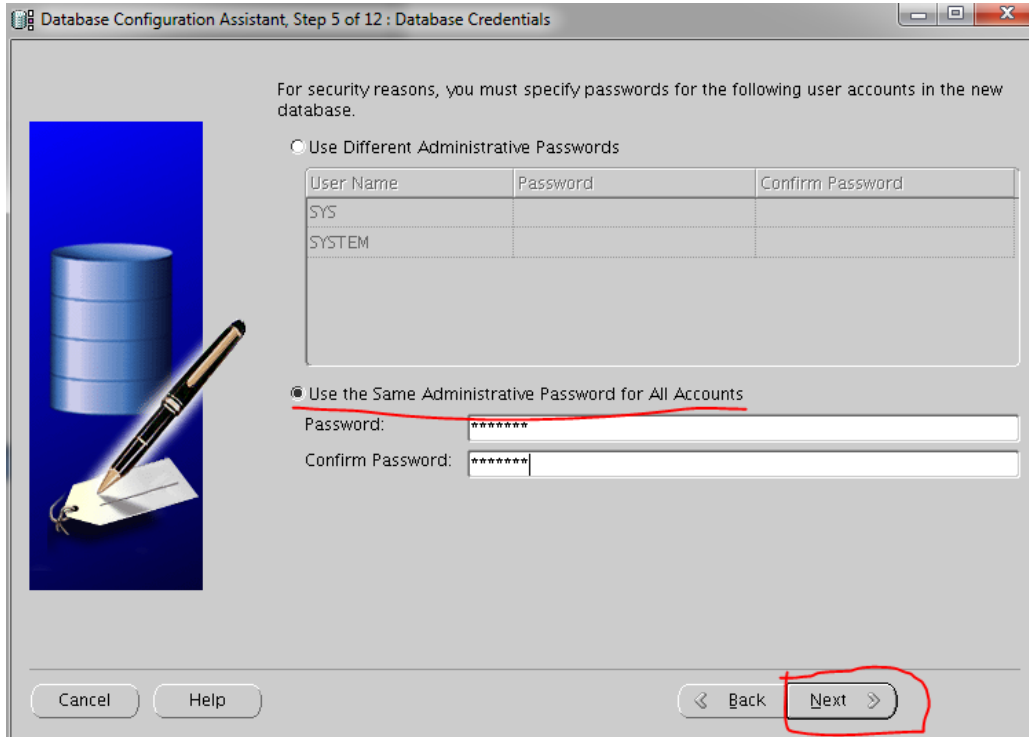
Global Database Name: oradb və SID: oradb daxil edib Next düyməsini sıxırıq. Global Database name adətən Domain istifadə edilən hallarda olur. Bizim halda yoxdur və oradb (SID name) olaraq seçirik.



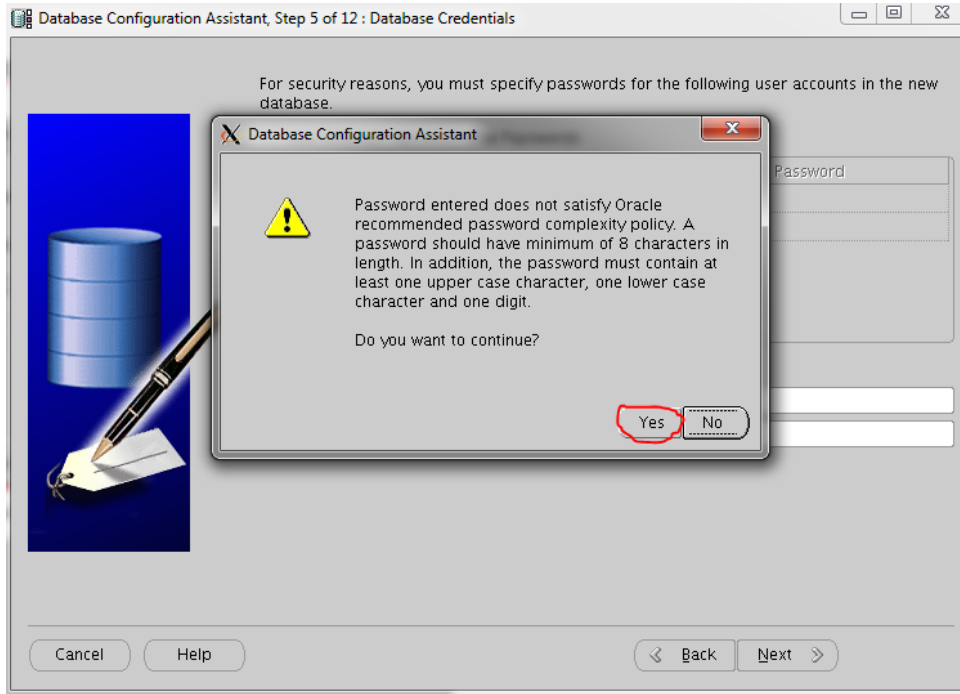
Enterprise Manager-in qıraşdırmasından selectoru götürürük və **Next** düyməsinə sızırıq.



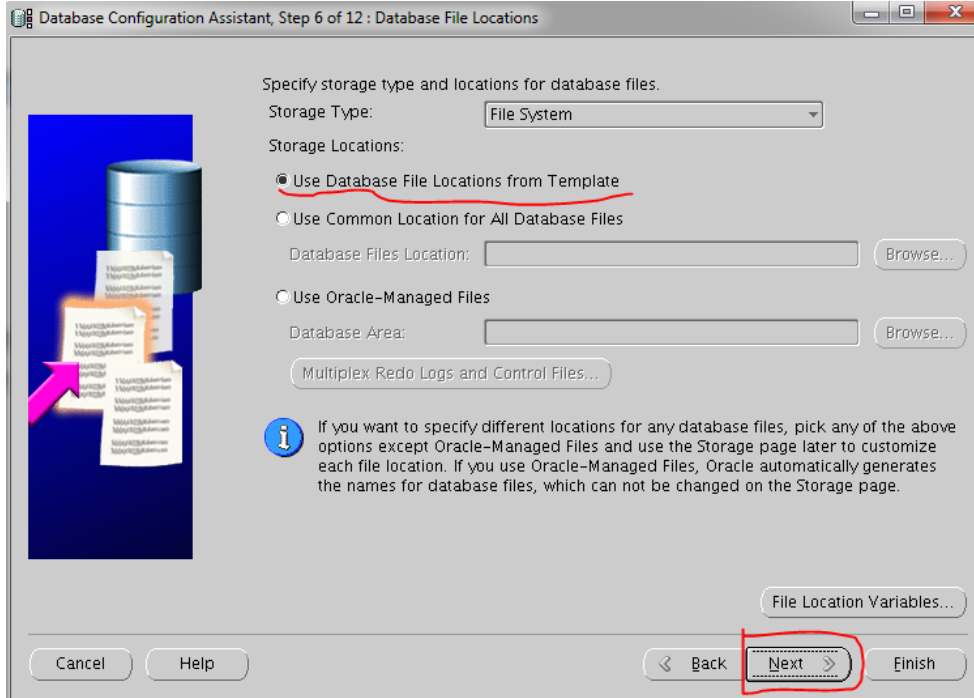
SYS və **SYSTEM** istifadəçisi üçün eyni şifrə daxil edirik və **Next** düyməsinə sızırıq.



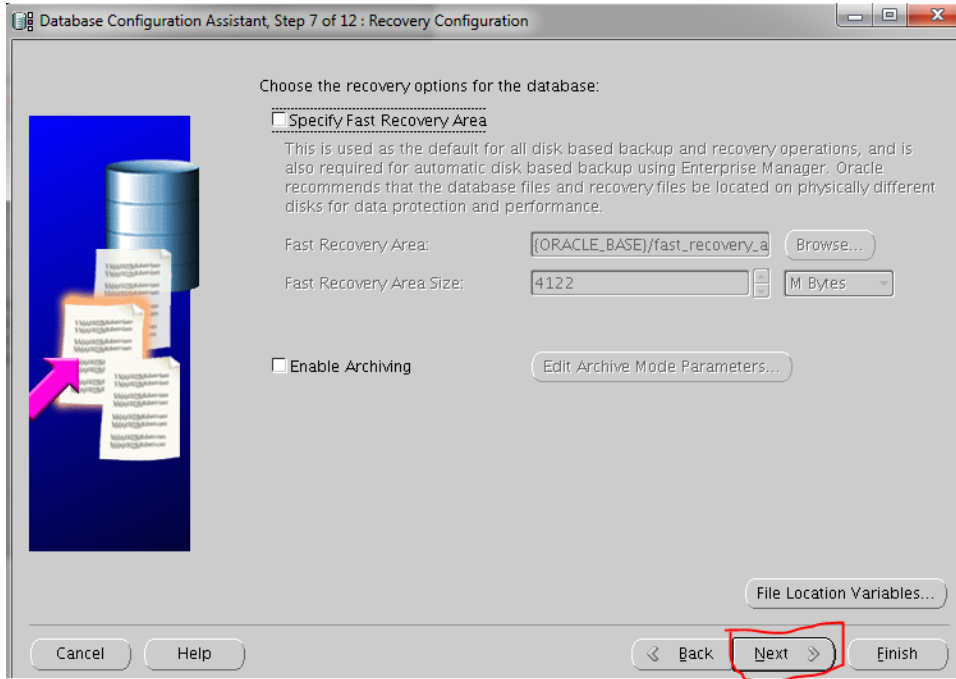
Çıxan şifrə siyasətinə **Yes** deyib davam edirik.



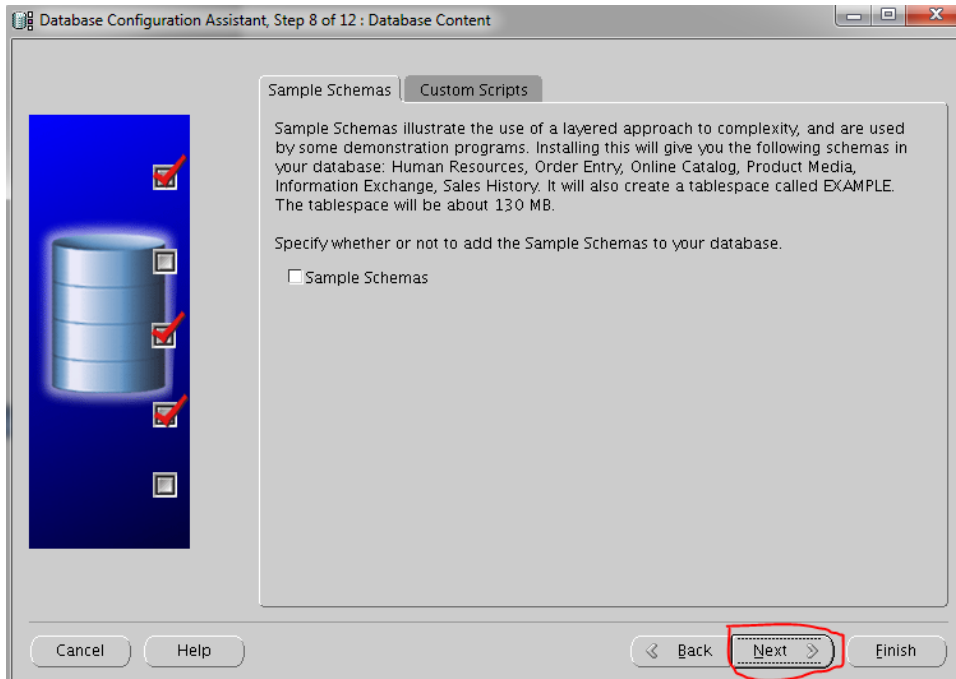
Artıq biz bazamızın ünvanı göstəriləcək quraşdırma yerinə gəldik. Oracle-ın imkanı var ki, jurnalları və redo jurnalları və bazanı müxtəlif ünvanlarda təyin eləsin. Ancaq biz bütün bazanı bir diskdə göstərəcəyik və onun daxilində isə bütün jurnalları müxtəlif ünvanlara yönəldəcəyik. Ona görə də **'Use Database File Locations from Template'** seçirik və **Next** düyməsinə sıxırıq.



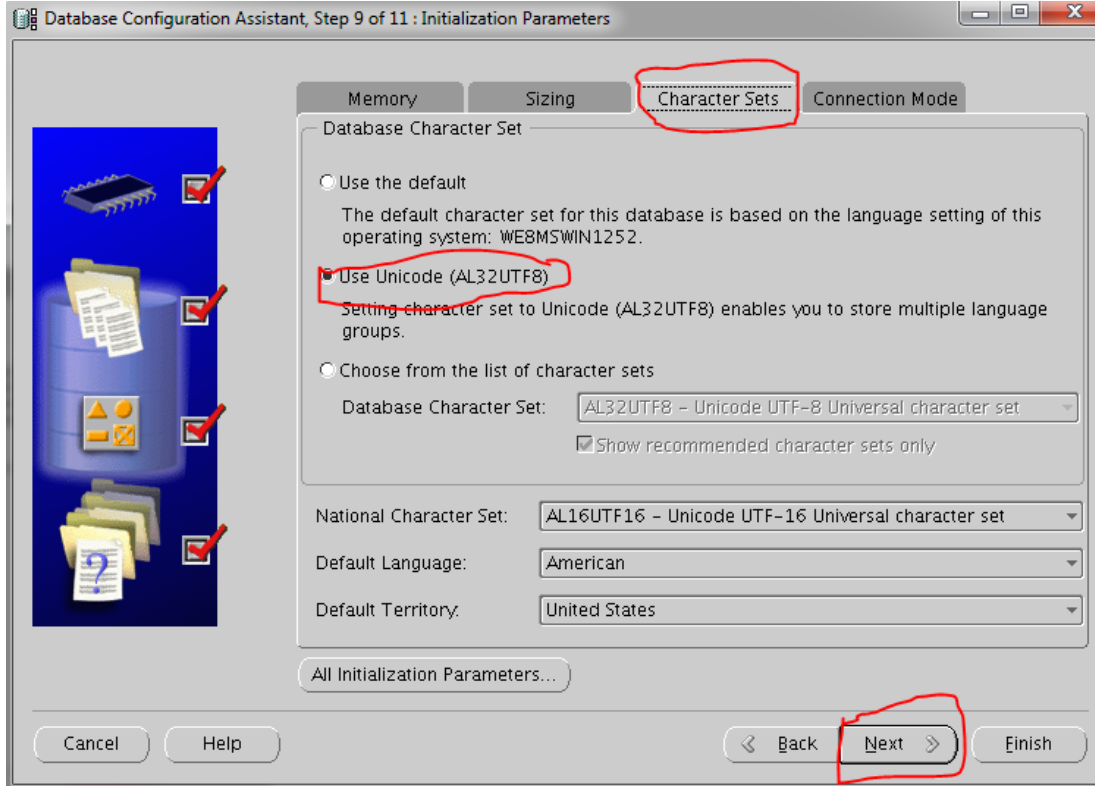
Specify Fast Recovery Areadan selektoru götürüb **Next** düyməsinə sıxırıq.



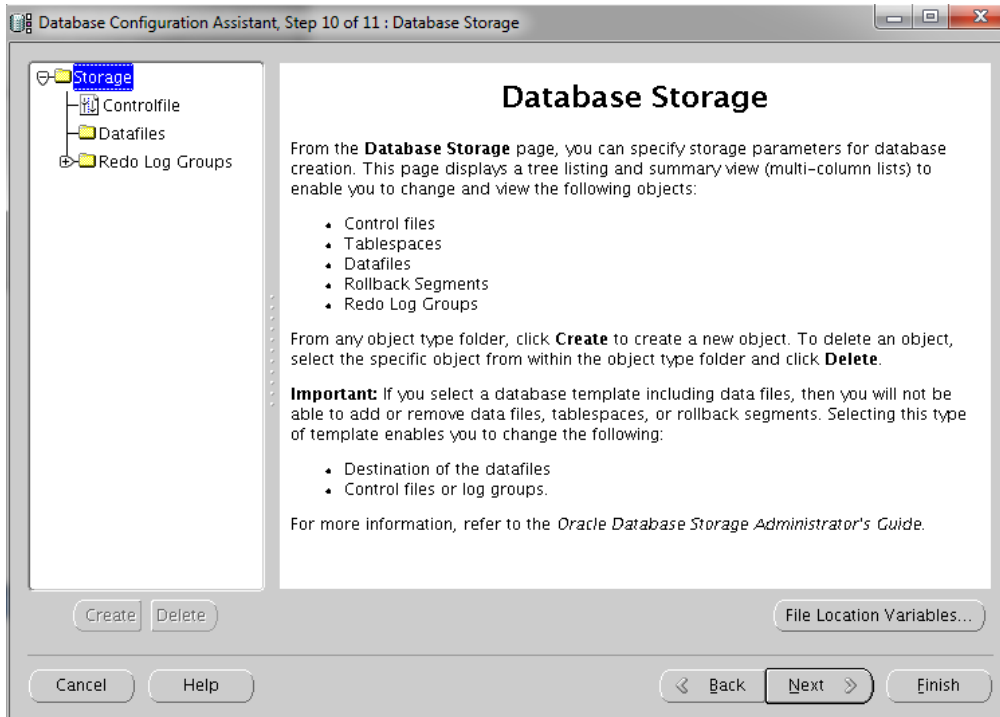
Növbəti səhifədə heçnə dəyişmədən **Next** düyməsinə sıxırıq.



Və **Character Sets** -> **Use Unicode (AL32UTF8)** və **Next** düyməsini sıxırıq.



Artıq aşağıdakı şəkil gəldikdə hansı faylların hansı ünvanında yerləşdirilməsinin quraşdırması gələcək. Bunun üçün öncə sistemdə **oracle** istifadəçi adından tələb edilən qovluqları yaratmaq lazımdır.



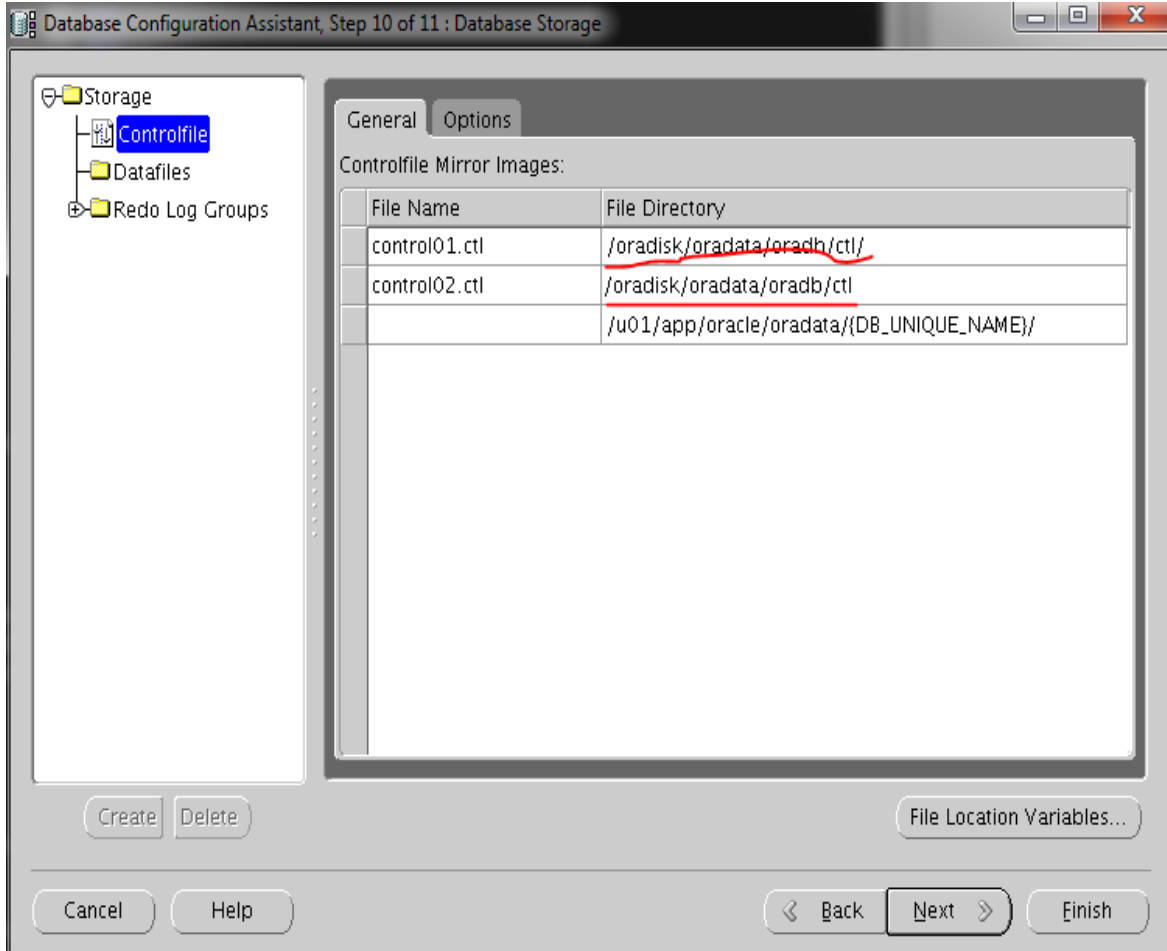
Öncə **root** istifadəçi adından **'/oradisk'** qovluğuna tələb edilən yetkiləri verək.

```
[root@node1 ~]# chown -R oracle:oinstall /oradisk/
[root@node1 ~]# chmod -R 755 /oradisk/
```

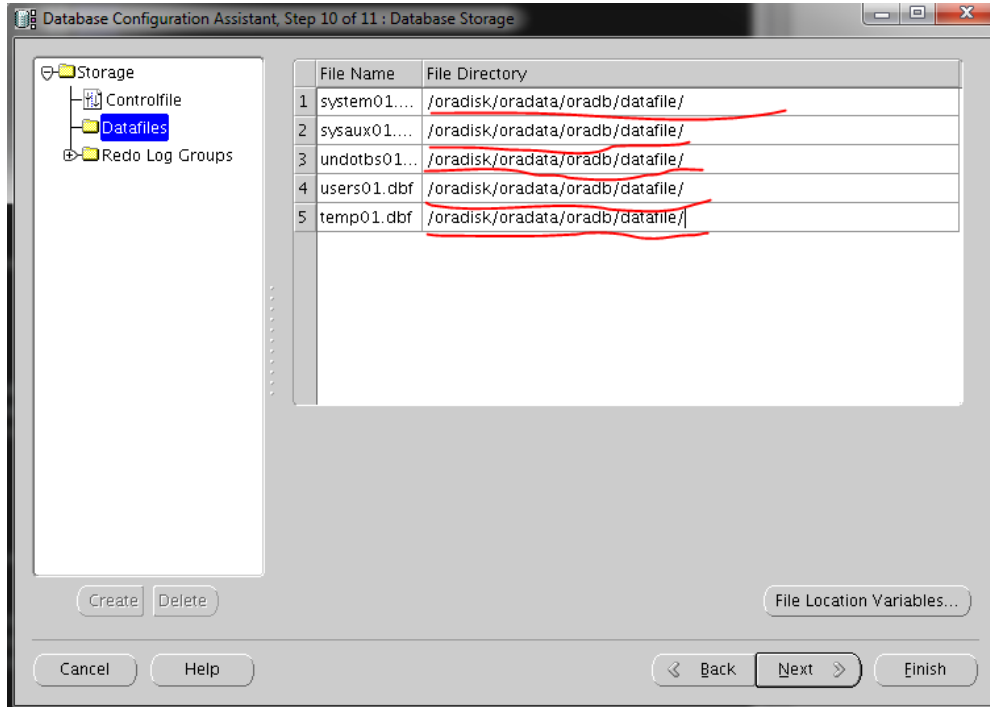
Sonra isə **oracle** istifadəçi adı ilə daxil olub bizə lazım olan qovluqları yaradaq.

```
[oracle@node1 ~]$ mkdir -p /oradisk/oradata/oradb
[oracle@node1 ~]$ cd /oradisk/oradata/oradb/
[oracle@node1 oradb]$ mkdir -p datafile ctl redo archlog
```

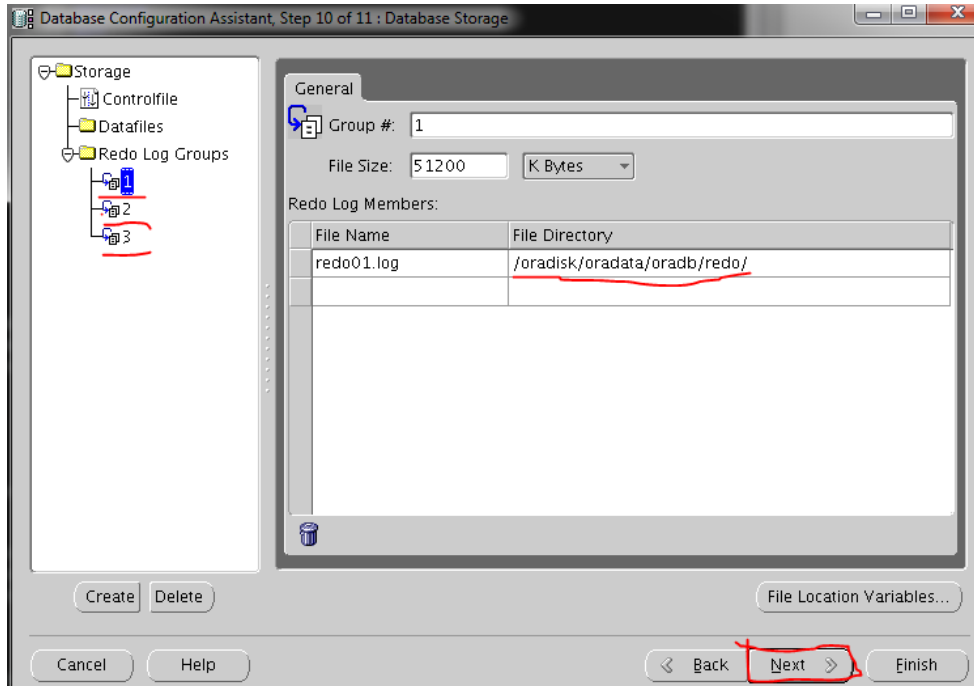
Ardınca **Control** fayllar üçün tələb olan faylların ünvanını əlavə edirik. **Control** jurnal faylların hamısının bir ünvanda yerləşməsi əslində ayrı-ayrı olmalıdır. Bizim halda isə bu kifayət edir.



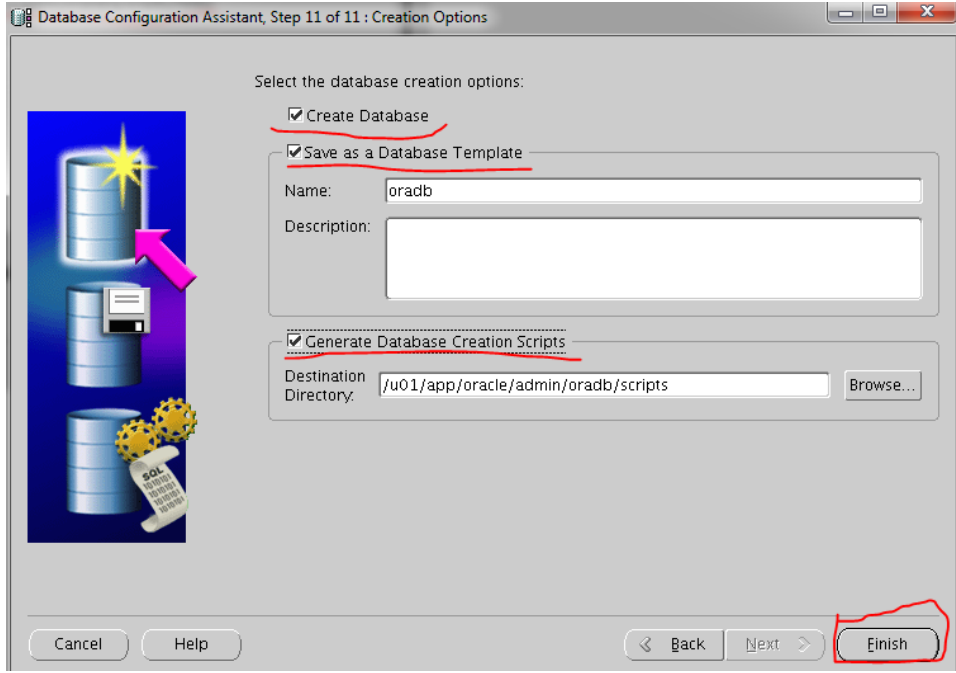
Sonra data faylların ünvanını təyin edirik.



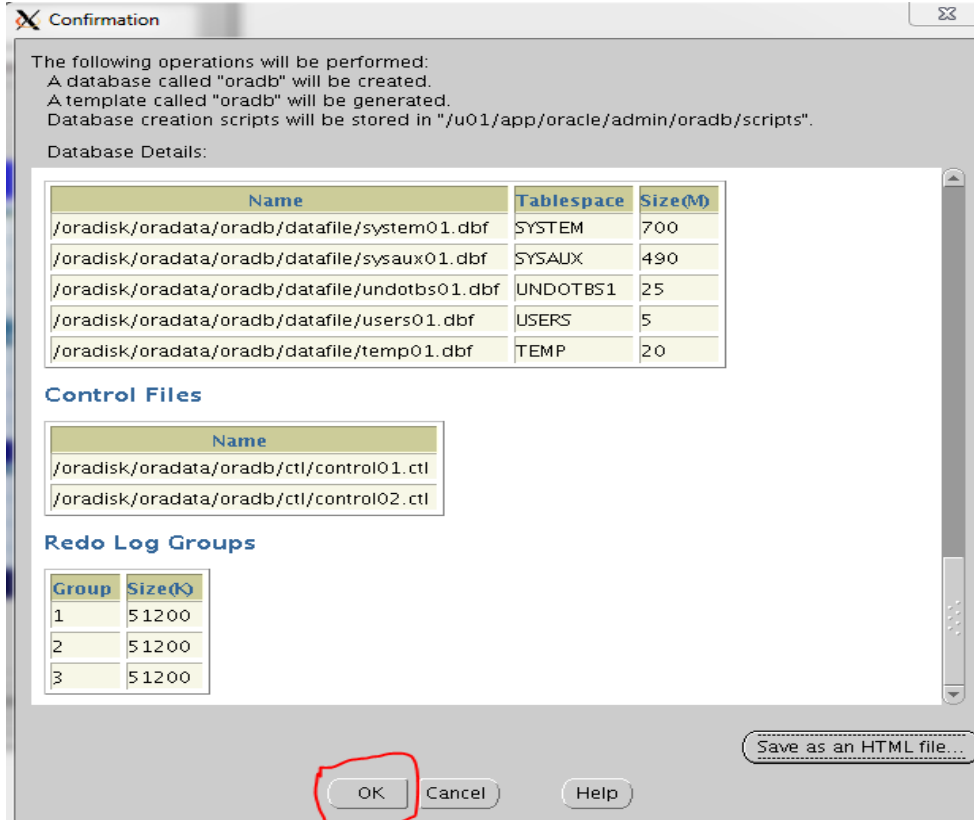
Bu etapda 'Redo Log Groups'-un altında həm 1, həm 2 və həm də 3-də şəkildə görünən yaratdığımız qovluq təyin edirik və **Next** düyməsini sıxırıq. Əslində bu faylları müxtəlif disklərdə saxlasanız yaxşı olar. Bunu etmək üçün isə çoxlu drbd disklər yaradıb onun üstünə köçürə bilərsiniz. Ancaq bizim məqsədimiz ümumi məntiq öyrənməkdir. Andınca istənilən dizaynı özünüz qura bilərsiniz.



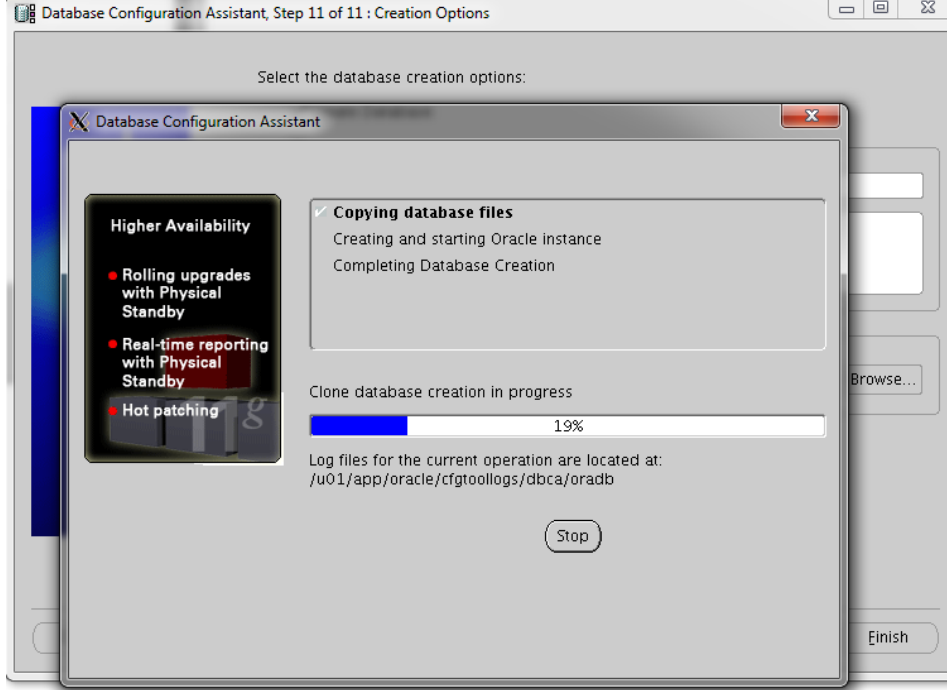
Bazanı yaradıırıq, özümüze Şablon nüsxə edirik və scriptləri göstərilən ünvarda yadda saxlayırıq.



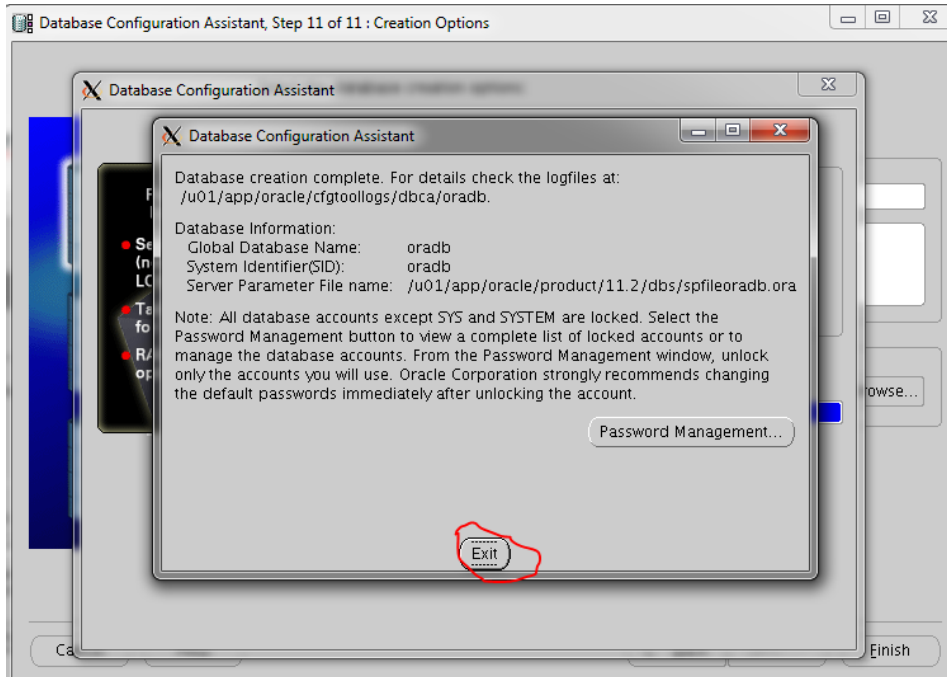
Şekildəki kimi OK, OK və yenidən OK düyməsini sıxırıq.



Və şəkildəki kimi bazanın yaradılması prosesi başlayır.



Aşağıda çıxan şəkildə 'Exit' düyməsini sıxırıq.



```
# cd /u01/app/oracle/product/11.2/network/admin # Sonra 'oracle' istifadəçi  
bu qovluqdan aşağıdakı  
faylları digər noda-a oracle  
istifadəçi adı ilə SCP  
vasitəsilə ilə nüsxələyirik.
```

```
[oracle@node1 admin]$ scp listener.ora
oracle@node2:/u01/app/oracle/product/11.2/network/admin
```

```
[oracle@node1 admin]$ scp tnsnames.ora
oracle@node2:/u01/app/oracle/product/11.2/network/admin
```

```
[oracle@node1 admin]$ cd $ORACLE_HOME/dbs # Sonra 'oracle' istifadəçi
bu qovluqdan aşağıdakı
faylları digər noda-a oracle
istifadəçi adı ilə SCP
vaxitəsilə ilə nüsxələyirik.
```

```
[oracle@node1 dbs]$ scp orapworadb
oracle@node2:/u01/app/oracle/product/11.2/dbs
```

Ardınca **oracle** istifadəçisinin konsolundan SQL-də aşağıdakı dəyişiklikləri əlavə edərək deyirik ki, Oracle göstərdiyimiz Local Listenerdə qulaq assın.

```
[oracle@node1 dbs]$ sqlplus / as sysdba # Oracle-a daxil oluruq
SQL*Plus: Release 11.2.0.3.0 Production on Thu Mar 14 18:18:11 2013
Copyright (c) 1982, 2011, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> alter system set local_listener='LISTENER'; # LISTENER-i teyin edirik.
```

```
[oracle@node1 admin]$ cd $ORACLE_HOME/dbs # Sonra 'oracle' istifadəçisi
bu qovluqdan aşağıda
göstərilən faylları digər
noda-a oracle istifadəçi adı
ilə SCP vasitəsilə ilə
nüsxələyirik.
```

```
[oracle@node1 dbs]$ scp spfileoradb.ora
oracle@node2:/u01/app/oracle/product/11.2/dbs
```

```
[oracle@node1 dbs]$ vi spfileoradb.ora # Faylın içində
*.audit_file_dest='/u01/app/o
racle/admin/oradb/adump'
təyin edilən ünvanı node2-də
yaradırıq
```

```
[oracle@node2 ~]$ mkdir -p /u01/app/oracle/admin/oradb/adump # Həmin qovluğu
node2-də
yaradırıq.
```

Ardınca serverlərimizi söndürüb yenidən işə salırıq. Clusterin statusuna baxırıq.

```
[root@node1 ~]# crm_mon -l
```

Last updated: Thu Mar 14 18:31:36 2013

Last change: Thu Mar 14 15:50:02 2013 via cibadmin on node1

Stack: classic openais (with plugin)

Current DC: node2 - partition with quorum

Version: 1.1.8-7.el6-394e906

2 Nodes configured, 2 expected votes

8 Resources configured.

Online: [node1 node2]

Master/Slave Set: ms_drbd_oracle [p_drbd_oracle]

Masters: [node1]

Slaves: [node2]

Resource Group: g_oracle

p_fs_oracle (ocf::heartbeat:Filesystem): **Started node1**

p_ip_oracle (ocf::heartbeat:IPaddr2): **Started node1**

ora_lsnr (ocf::heartbeat:oralsnr): **Started node1**

ora_db (ocf::heartbeat:oracle): **Started node1**

Clone Set: cl_ping [p_ping]

Started: [node1 node2]

```
[oracle@node1 ~]$ lsnrctl status
```

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 14-MAR-2013 18:29:24

Copyright (c) 1991, 2011, Oracle. All rights reserved.

Connecting to

(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=172.16.1.100)(PORT=1521)))

STATUS of the LISTENER

Alias LISTENER

Version TNSLSNR for Linux: Version 11.2.0.3.0 - Production

Start Date 14-MAR-2013 18:28:33

Uptime 0 days 0 hr. 0 min. 51 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File

/u01/app/oracle/product/11.2/network/admin/listener.ora

Listener Log File

/u01/app/oracle/diag/tnslsnr/node1/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=172.16.1.100)(PORT=1521)))

Services Summary...

Service "oradb" has 2 instance(s).

Instance "oradb", status UNKNOWN, has 1 handler(s) for this service...

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "oradbXDB" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...
The command completed successfully

Sınaqlarımızı edək.

Sonra isə Oracle istifadəçi adından sistemə daxil olub **oracle**-a qoşuluruq və cədvəl yaradırıq. Həmin cədvəlin içinə bir neçə informasiya əlavə edək ki, yoxlayaq görək bazamız bir node-dan digərinə sinxronizasiya edilir ya yox.

```
[root@node1 ~]# su - oracle
```

```
[oracle@node1 ~]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu Mar 14 18:35:15 2013
```

```
Copyright (c) 1982, 2011, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> CREATE USER gosha identified by freebsd; # Gosha adlı istifadəçi əlavə  
edirik.(şifrə: freebsd)
```

```
SQL> GRANT CONNECT, CREATE TABLE to gosha; # Gosha adlı istifadəçi üçün  
qoşulmağa və cədvəl yaratmağa  
izin veririk.
```

```
SQL> alter user gosha default tablespace users; # gosha istifadəçi üçün  
default tablespace yaratdıq.
```

```
SQL> alter user gosha quota unlimited on users; # gosha istifadəçisinə users  
table ünvanına istənilən  
qədər məlumat yazmasına izin  
verdik
```

Ardınca **node1**-i söndürürük və **node2**-də Clusterimizi yoxlayırıq.

```
[root@node2 ~]# crm_mon -1
```

```
Last updated: Thu Mar 14 20:11:14 2013
```

```
Last change: Thu Mar 14 19:39:40 2013 via crm_resource on node2
```

```
Stack: classic openais (with plugin)
```

```
Current DC: node2 - partition WITHOUT quorum
```

```
Version: 1.1.8-7.el6-394e906
```

```
2 Nodes configured, 2 expected votes
```

```
8 Resources configured.
```

```
Node node1: OFFLINE (standby)
```

```
Online: [ node2 ]
```

```
Master/Slave Set: ms_drbd_oracle [p_drbd_oracle]
```

```
Masters: [ node2 ]
```

```
Stopped: [ p_drbd_oracle:1 ]
```

```
Resource Group: g_oracle
```

```
p_fs_oracle (ocf::heartbeat:Filesystem): Started node2
```

```
p_ip_oracle (ocf::heartbeat:IPaddr2): Started node2
```

```
ora_lsnr (ocf::heartbeat:oralsnr): Started node2
```



```
ora_db      (ocf::heartbeat:oracle):      Started node2
Clone Set: cl_ping [p_ping]
Started: [ node2 ]
Stopped: [ p_ping:1 ]
```

gosha istifadəçi adı ilə bazaya qoşulub cədvəl yaradıb ora nə isə **insert** edək və **node1**-i işə salıb yoxlayaq görək informasiya ora sinxronizasiya edilib ya yox.

```
[oracle@node2 ~]$ sqlplus gosha@oradb      # gosha istifadəçi adı ilə bazaya
                                           qoşuluruq
```

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu Mar 14 20:07:54 2013
```

```
Copyright (c) 1982, 2011, Oracle. All rights reserved.
```

```
Enter password:
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> CREATE TABLE drbd( sutun INT);      # drbd adlı cədvəl yaratdıq
```

```
SQL> INSERT INTO drbd values ( 1 );      # Və həmin cədvələ bir neçə rəqəm
                                           insert elədik.
```

```
SQL> INSERT INTO drbd values ( 2 );
```

```
SQL> INSERT INTO drbd values ( 3 );
```

```
SQL> INSERT INTO drbd values ( 4 );
```

```
SQL> commit;                             # Və sonda Commit edirik ki,
                                           dəyişikliklərimiz saxlansın.
```

```
Commit complete.
```

Və sonda **node1**-i işə salıb yoxlayaq görək yaratdığımız məlumatlar orda yerləşir ya yox. Gördüyümüz kimi resurslarımız node1-ə miqrasiya edilib.

```
[root@node1 ~]# crm_mon -1
```

```
Last updated: Thu Mar 14 20:33:34 2013
```

```
Last change: Thu Mar 14 19:39:40 2013 via crm_resource on node2
```

```
Stack: classic openais (with plugin)
```

```
Current DC: node2 - partition with quorum
```

```
Version: 1.1.8-7.el6-394e906
```

```
2 Nodes configured, 2 expected votes
```

```
8 Resources configured.
```

```
Online: [ node1 node2 ]
```

```
Master/Slave Set: ms_drbd_oracle [p_drbd_oracle]
```

```
Masters: [ node1 ]
```

```
Slaves: [ node2 ]
```

```
Resource Group: g_oracle
```

```
p_fs_oracle      (ocf::heartbeat:Filesystem):      Started node1
```

```
p_ip_oracle      (ocf::heartbeat:IPaddr2):          Started node1
```

```
ora_alsnr        (ocf::heartbeat:oralsnr):          Started node1
```

```
ora_db           (ocf::heartbeat:oracle):          Started node1
```

```
Clone Set: cl_ping [p_ping]
Started: [ node1 node2 ]
```

Bazaya qoşulub cədvəlimiz və içində olan məlumatın **node1**-də olduğunu yoxlayaq. Gördüyümüz kimi məlumatlar burda da mövcuddur ☺.

```
[root@node1 ~]# su - oracle
```

```
[oracle@node1 ~]$ sqlplus gosha@oradb
```

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu Mar 14 20:36:14 2013
```

```
Copyright (c) 1982, 2011, Oracle. All rights reserved.
```

```
Enter password:
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> select * from drbd;
```

```
      SUTUN
```

```
-----
```

```
      1
```

```
      2
```

```
      3
```

```
      4
```

```
     10
```

Sınaq üçüb node-ları əlinizlə də **migrate** edə bilərsiniz. Oracle **Resource Group**-u **node2**-dən **node1**-ə miqrasiya edirik.

```
[root@node2 ~]# crm resource migrate g_oracle node1
```