설치

Linux iSCSI target Server & TAC

TINTI036

Website
www.tmaxtibero.com

안내서 정보
안내서 제목: Linux iSCSI target Server & TAC
발행일: 2025-12-16
소프트웨어 버전: Tibero 7.2.4.
안내서 버전: 1.1.0

제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0.0	2017.10.24	최초 제정	
1.1.0	2025.12.16.	기존 자료 최신화	

목차

1. 시작하기 전에

iSCSI target 서버를 storage 로 이용하는 2 node TAC를 구축합니다. 네트워크 기반의 공유 disk 볼륨을 구성함으로써, 'FC cable 절체'를 포함한 다양한 시나리오의 가용성 테스트를 진행할 수 있습니다. 사용되는 패키지 및 기능에 대해서

는 간략하게 설명하였고 명령어/스크립트 위주로 작성하였습니다. ip, hostname 등의 정보는 임의로 설정하였으며, 실제 구축 시에는 해당 내용을 변경하여 사용하시면 됩니다. 추가 IP 할당이 불가능해 host-only, NAT 스위치를 이용해 구성했습니다.(VMWARE 네트워크 설정 문서 참고)

2. 준비사항 (테스트 환경)

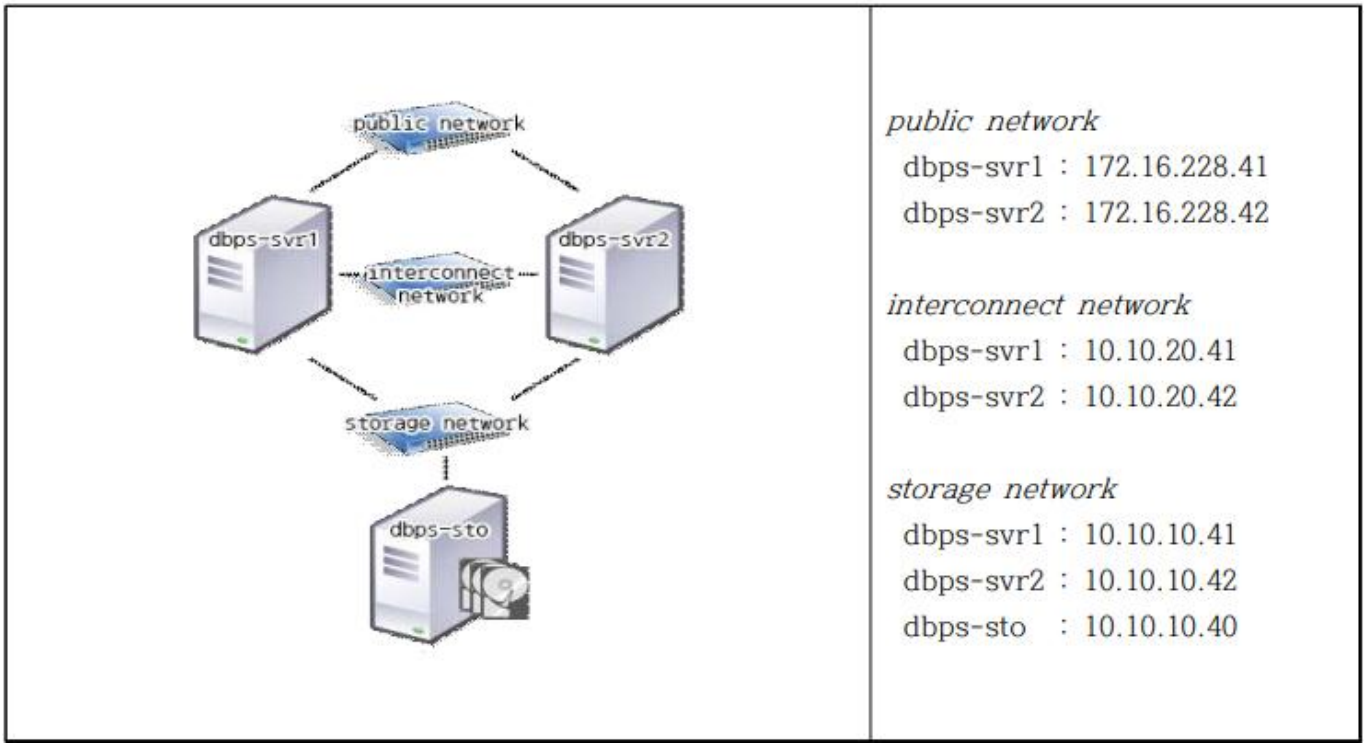
다음과 같은 사양의 장비, 소프트웨어 환경에서 진행했습니다. 본 문서는 가상 머신에서 테스트한 것을 바탕으로 작성되었으며, '4.가.1) Network 설정'의 스위치, '4.나.2) 가상머신 설정'의 사양과 같은 실제 물리 장비 구성으로 구축 가능합니다.

가. host server : CPU: Intel x86_64 Xeon E3-1231v3
Ram: DDR3-12800 32GB
OS: Ubuntu Linux 16.04.2

나. vmware version : VMware Workstation 12.5.7 build-5813279

다. guest server (linux) iso image : CentOS-7-x86_64-DVD-1708

3. 구성도

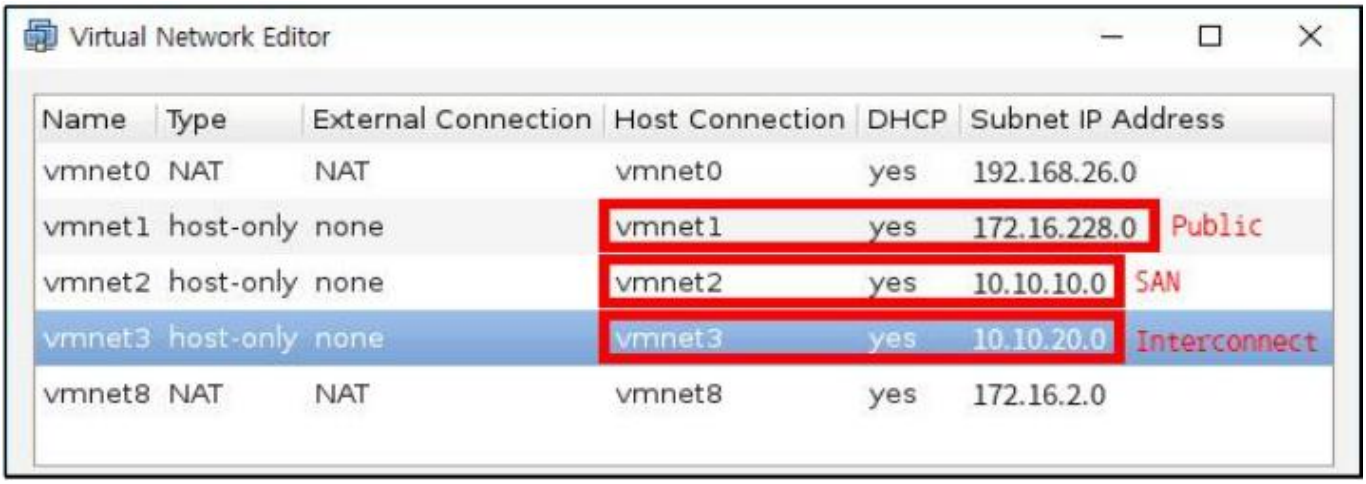


4. vmware 구성 및 Guest OS 설치

가. Network 설정

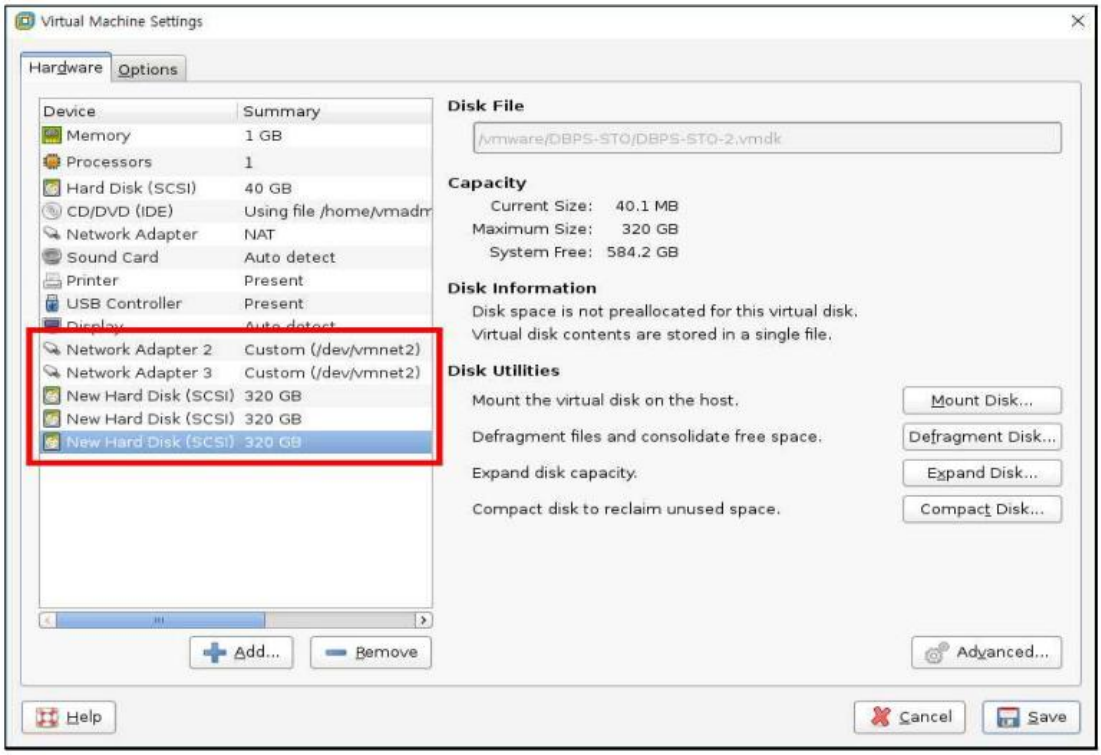
1) 가상 스위치 설정 다음과 같은 목적으로 가상 스위치를 설정합니다. (VMWARE 네트워크 설정 문서 참고) vmnet1 (public network), vmnet2 (storage network), vmnet3 (interconnect network)

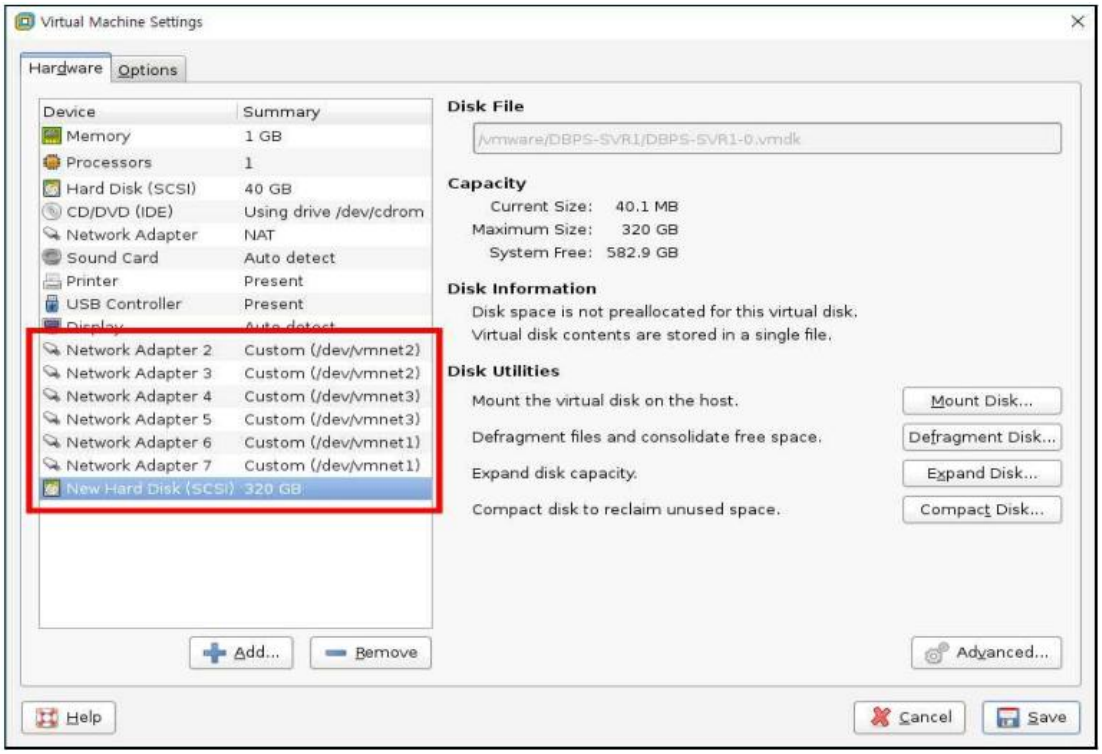
2) Virtual Network Editor 다음과 같이 vmnet1,vmnet2,vmnet3 로 설정된 host-only 가상 스위치를 설정합니다. public network (vmnet1)의 경우 추가 할당 가능한 ip가 있다면 bridge type으로 구성함으로써, port-forwarding 없이 host server 외부에서 접근이 가능합니다.



나. 가상 머신 설정

1) iSCSI storage server

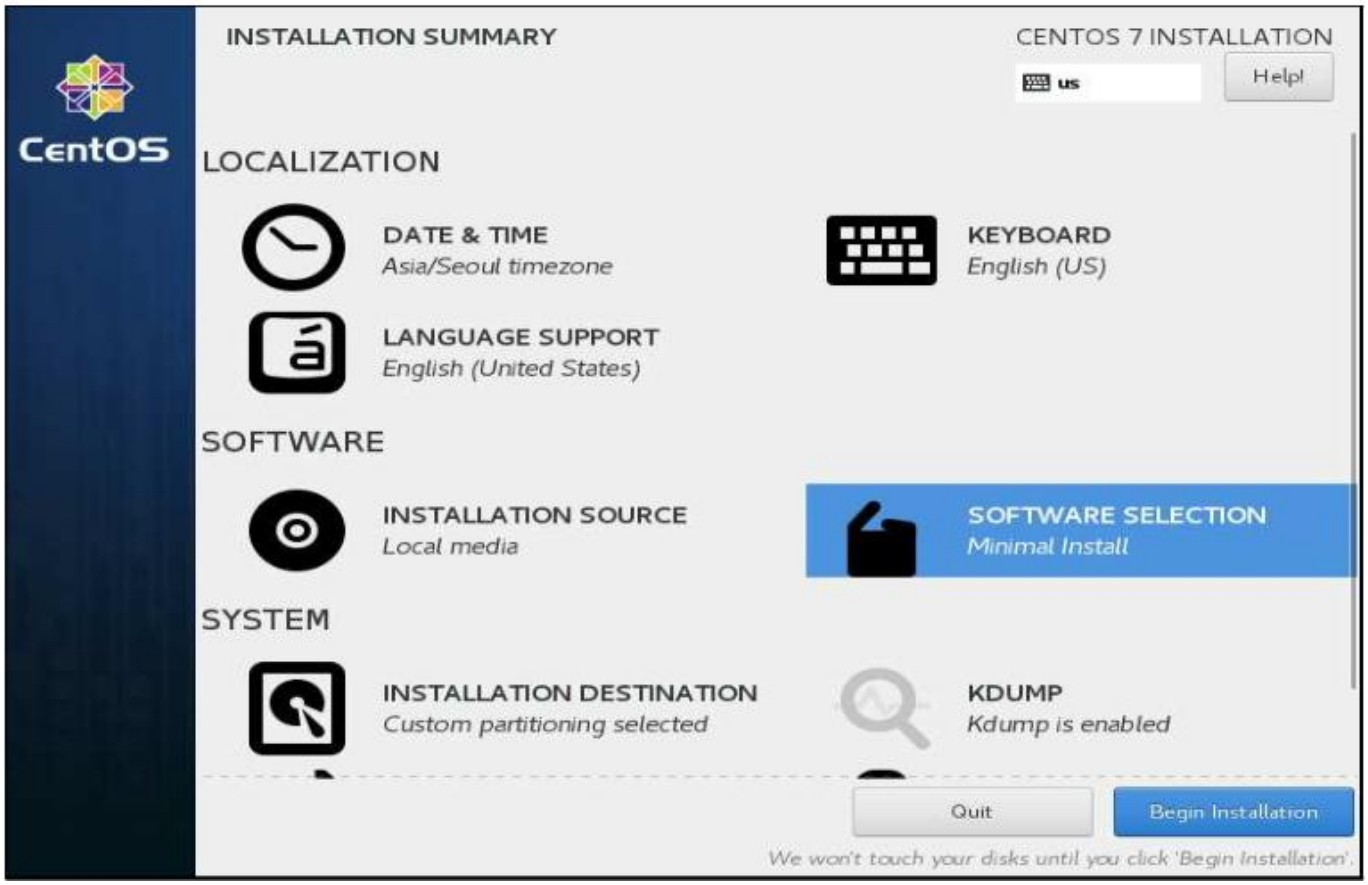




다. OS 설치

1) Minimal Install

불필요한 패키지가 설치되지 않도록 'Minimal Install'을 선택합니다.



5. Storage 서버 구축

가. network bonding 작업

'/etc/sysconfig/network-scripts/ifcfg-*', '/etc/sysconfig/network', '/etc/modprobe.d/bonding.conf' 파일을 생성/수정하여 bonding network를 설정할 수 있으나, CentOS 7 Minimal Install 설치 시 기본 설치되는 'nmcli' (network manager command-line interface)를 이용하여 구성하겠습니다

```
[root@dbps-sto ~]# nmcli d
DEVICE  TYPE      STATE         CONNECTION
ens33   ethernet  connected     ens33
ens34   ethernet  disconnected  --
ens35   ethernet  disconnected  --
lo      loopback  unmanaged     --
```

```
[root@dbps-sto ~]# nmcli connection modify ens34 connection.autoconnect no
[root@dbps-sto ~]# nmcli connection modify ens35 connection.autoconnect no
[root@dbps-sto ~]# nmcli connection add type bond con-name bond0 ifname bond0 mode active-backup
Connection 'bond0' (7bfe221e-646e-420f-8fff-0a0d86d90d3c) successfully added.
[root@dbps-sto ~]# nmcli d
DEVICE   TYPE        STATE                                  CONNECTION
ens33    ethernet    connected                              ens33
bond0    bond        connecting (getting IP configuration)  bond0
ens34    ethernet    disconnected                           --
ens35    ethernet    disconnected                           --
lo       loopback    unmanaged                              --
[root@dbps-sto ~]# nmcli connection add type bond-slave ifname ens34 master bond0
Connection 'bond-slave-ens34' (a574ae0e-91cc-4251-ae08-e4a280407869) successfully added.
[root@dbps-sto ~]# nmcli connection add type bond-slave ifname ens35 master bond0
Connection 'bond-slave-ens35' (9285c100-e32f-46db-9571-b9f5d2f37b4b) successfully added.
[root@dbps-sto ~]# nmcli connection modify bond0 ipv4.method manual connection.autoconnect yes ipv4.addres
10.10.10.40/24 ipv4.gateway 10.10.10.2 ipv4.dns 10.10.10.2
[root@dbps-sto ~]# nmcli connection modify bond0 +bond.options primary=ens34 +bond.options miimon=10
+bond.options updelay=0 +bond.options downdelay=0
[root@dbps-sto ~]# nmcli d
DEVICE   TYPE        STATE         CONNECTION
bond0    bond        connected     bond0
ens33    ethernet    connected     ens33
ens34    ethernet    connected     bond-slave-ens34
ens35    ethernet    connected     bond-slave-ens35
lo       loopback    unmanaged     --
[root@dbps-sto ~]# systemctl restart network
```

※bonding mode

#	mode	desc
0	balance-rr, Round Robin Load Balancing	패킷 송/수신에서 NIC를 번갈아가며 Load Balance.
1	active-backup	기본적으로 Primay NIC만을 이용하여 통신하며, 장애 시에 backup NIC를 사용
2	balance-xor, Load Balancing	source, target NIC의 MAC address를 xor 연산하여, 규칙에 따라 Load Balance

나. md 구성

/dev/md* (multi-path disk) 구성을 합니다. raid control hardware 장비를 이용해 물리적으로 구성 할 수도 있지만 mdadm 패키지를 이용하여 software 방식의 raid를 구성합니다. 서버의 scsi controler에 구성된 /dev/sd* (single-path disk)를 조합하여 software 방식의 raid5를 구성합니다.

```
[root@dbps-sto ~]# yum -y install mdadm
.
.
[root@dbps-sto ~]# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
```

```
mdadm: chunk size defaults to 512K
mdadm: size set to 335413248K
mdadm: automatically enabling write-intent bitmap on large array
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
#아래의 파일을 조회하였을 때, recovery = 100% 가 될 때까지 대기합니다. (100%가 되어야 md build가 완료된 것입니다.)
[root@dbps-sto ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sdd[3] sdc[1] sdb[0]
      670826496 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
      [>....................]  recovery =  0.4% (1600640/335413248) finish=24.3min speed=228662K/sec
      bitmap: 3/3 pages [12KB], 65536KB chunk
```

※RAID5

최소 3개 이상의, n 개의 disk를 이용하여 RAID5 md를 구성한 경우, 최대 n-1배의 읽기 속도, n-1배의 저장 공간을 가질 수 있습니다. 또한 분산 저장된 parity 정보를 바탕으로, disk 1개 장애 시 복구가 가능합니다.

다. targetcli disk target, lun 생성

```
[root@dbps-sto ~]# yum -y install targetcli
.
.
[root@dbps-sto ~]# targetcli
targetcli shell version 2.1.fb46
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> cd backstores/block
/backstores/block> create sto_disk1_target /dev/vg_sto/lvdisk1
Created block storage object sto_disk1_target using /dev/vg_sto/lvdisk1.
/backstores/block> cd /iscsi
/iscsi> create iqn.2017-10.dbps-sto.target:disk1
Created target iqn.2017-10.dbps-sto.target:disk1.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
/iscsi> cd iqn.2017-10.dbps-sto.target:disk1/tpg1/acls
# iqn.YYYY-MM."target_server_domain의-역순":initiator# 와 같은 규칙으로 iqn을 생성합니다.
# target server의 domain이 dbps-sto 인 경우 아래와 같습니다.
/iscsi/iqn.20...sk1/tpg1/acls> create iqn.2017-10.dbps-sto.target:initiator1
Created Node ACL for iqn.2017-10.dbps-sto.target:initiator1
# target server 의 domain이 tmaxsoft-dbps.com 인 경우 iqn.2017-10.com.tmaxsoft-dbps.target:initiator# 로 설정합니다.
# 다음과 같이 노드별로 iqn을 생성합니다.
/iscsi/iqn.20...sk1/tpg1/acls> create iqn.2017-10.dbps-sto.target:initiator2
Created Node ACL for iqn.2017-10.dbps-sto.target:initiator2
/iscsi/iqn.20...sk1/tpg1/acls> cd iqn.2017-10.dbps-sto.target:initiator1
/iscsi/iqn.20...get:initiator1> set auth userid=root
Parameter userid is now 'root'.
/iscsi/iqn.20...get:initiator1> set auth password=tibero
Parameter password is now 'tibero'.
/iscsi/iqn.20...get:initiator> cd ..
/iscsi/iqn.20...sk1/tpg1/acls> cd iqn.2017-10.dbps-sto.target:initiator2
```

```
Created Node ACL for iqn.2017-10.dbps-sto.target:initiator2
/iscsi/iqn.20...sk1/tpg1/acls> cd iqn.2017-10.dbps-sto.target:initiator1
/iscsi/iqn.20...get:initiator1> set auth userid=root
Parameter userid is now 'root'.
/iscsi/iqn.20...get:initiator1> set auth password=tibero
Parameter password is now 'tibero'.
/iscsi/iqn.20...get:initiator> cd ..
/iscsi/iqn.20...sk1/tpg1/acls> cd iqn.2017-10.dbps-sto.target:initiator2
/iscsi/iqn.20...get:initiator2> set auth userid=root
Parameter userid is now 'root'.
/iscsi/iqn.20...get:initiator2> set auth password=tibero
Parameter password is now 'tibero'.
/iscsi/iqn.20...get:initiator> cd /iscsi/iqn.2017-10.dbps-sto.target:disk1/tpg1/luns
/iscsi/iqn.20...sk1/tpg1/luns> create /backstores/block/sto_disk1_target
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2017-10.dbps-sto.target:initiator
/iscsi/iqn.20...sk1/tpg1/luns> cd /
/> saveconfig
Last 10 configs saved in /etc/target/backup.

Configuration saved to /etc/target/saveconfig.json
/> exit
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/target/backup.
Configuration saved to /etc/target/saveconfig.json
[root@dbps-sto ~]#
```

※storage 아키텍처 별 최대 대역폭

아키텍처	
iSCSI (GB Ethernet)	125 MB/s
Fiber Channel (2 GFC)	212.5 MB/s
SATA (SATAII)	300 MB/s
SCSI (U320)	320 MB/s
SAS	375 MB/s
Fiber Channel (4 GFC)	425 MB/s
Fiber Channel (8 GFC)	850 MB/s
iSCSI (10 GB Ethernet)	1250 MB/s

6. DB 서버 구축

가. network bonding 작업

```
[root@dbps-svr1 ~]# nmcli d
DEVICE  TYPE      STATE         CONNECTION
ens33   ethernet  connected     ens33
ens34   ethernet  disconnected  --
ens35   ethernet  disconnected  --
ens36   ethernet  disconnected  --
ens37   ethernet  disconnected  --
ens38   ethernet  disconnected  --
ens39   ethernet  disconnected  --
lo      loopback  unmanaged     --
```

```
[root@dbps-svr1 ~]# nmcli connection modify ens34 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection modify ens35 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection add type bond con-name bond0 ifname bond0 mode active-backup
Connection 'bond0' (bcffcf68-beeb-4cec-a88b-81e9abd05329) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens34 master bond0
Connection 'bond-slave-ens34' (273ead3d-c783-4575-ae81-bf6fe8e3e71a) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens35 master bond0
Connection 'bond-slave-ens35' (0656773c-db7a-4aff-bad6-4cd220e3311f) successfully added.
[root@dbps-svr1 ~]# nmcli connection modify bond0 ipv4.method manual connection.autoconnect yes ipv4.address
10.10.10.41/24 ipv4.gateway 10.10.10.2 ipv4.dns 10.10.10.2
[root@dbps-svr1 ~]# nmcli connection modify bond0 +bond.options primary=ens34 +bond.options miimon=100

+bond.options updelay=0 +bond.options downdelay=0
[root@dbps-svr1 ~]# nmcli connection modify ens36 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection modify ens37 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection add type bond con-name bond1 ifname bond1 mode active-backup
Connection 'bond1' (d0b5a80c-f648-49bb-9297-ea7d4617b6ed) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens36 master bond1
Connection 'bond-slave-ens36' (dfc745ec-62c5-45fb-ac5c-e1c9cb7cac65) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens37 master bond1
Connection 'bond-slave-ens37' (032db675-5164-4388-9219-e0224d604a85) successfully added.
[root@dbps-svr1 ~]# nmcli connection modify bond1 ipv4.method manual connection.autoconnect yes ipv4.address
10.10.20.41/24 ipv4.gateway 10.10.20.2 ipv4.dns 10.10.20.2
[root@dbps-svr1 ~]# nmcli connection modify bond1 +bond.options primary=ens36 +bond.options miimon=100
+bond.options updelay=0 +bond.options downdelay=0
[root@dbps-svr1 ~]# nmcli connection modify ens38 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection modify ens39 connection.autoconnect no
[root@dbps-svr1 ~]# nmcli connection add type bond con-name bond2 ifname bond2 mode active-backup
Connection 'bond2' (0264c9ef-8169-4787-ad92-2081b69aafb2) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens38 master bond2
Connection 'bond-slave-ens38' (2ab0c98d-c10c-46eb-b57a-693d954c14da) successfully added.
[root@dbps-svr1 ~]# nmcli connection add type  bond-slave ifname ens39 master bond2
Connection 'bond-slave-ens39' (5e936851-7eec-4df5-bdc2-860521413cad) successfully added.
[root@dbps-svr1 ~]# nmcli connection modify bond2 ipv4.method manual connection.autoconnect yes ipv4.address
172.16.228.41/24 ipv4.gateway 172.16.228.2 ipv4.dns 172.16.228.2
[root@dbps-svr1 ~]# nmcli connection modify bond2 +bond.options primary=ens38 +bond.options miimon=100
+bond.options updelay=0 +bond.options downdelay=0
[root@dbps-svr1 ~]# nmcli d
DEVICE   TYPE        STATE        CONNECTION
bond0    bond        connected    bond0
bond1    bond        connected    bond1
bond2    bond        connected    bond2
ens33    ethernet    connected    ens33
ens34    ethernet    connected    bond-slave-ens34
ens35    ethernet    connected    bond-slave-ens35
ens36    ethernet    connected    bond-slave-ens36
ens37    ethernet    connected    bond-slave-ens37
ens38    ethernet    connected    bond-slave-ens38
ens39    ethernet    connected    bond-slave-ens39
lo       loopback    unmanaged    --
[root@dbps-svr1 ~]#
[root@dbps-svr1 ~]# systemctl restart network
```

나. disk 작업

1) local disk volume 작업

```
[root@dbps-svr1 ~]# cat /proc/partitions
major minor  #blocks  name

   8        0   41943040 sda
   8        1    1048576 sda1
   8        2   40893440 sda2
   8       16  335544320 sdb
  11        0    1048575 sr0
 253        0   32501760 dm-0
 253        1    8388608 dm-1

[root@dbps-svr1 ~]# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x68d6ad19.

Command (m for help): n

Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-671088639, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-671088639, default 671088639):
Using default value 671088639
Partition 1 of type Linux and of size 320 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@dbps-svr1 ~]# vgcreate vgdb /dev/sdb1
  Physical volume "/dev/sdb1" successfully created.
  Volume group "vgdb" successfully created
[root@dbps-svr1 ~]# lvcreate -L 40G -n bin vgdb /dev/sdb1
  Logical volume "bin" created.
[root@dbps-svr1 ~]# lvcreate -L 80G -n log vgdb /dev/sdb1
  Logical volume "log" created.
[root@dbps-svr1 ~]# lvcreate -L 40G -n dba vgdb /dev/sdb1
  Logical volume "dba" created.
[root@dbps-svr1 ~]# lvcreate -L 120G -n arch vgdb /dev/sdb1
  Logical volume "arch" created.
```

```
[root@dbps-svr1 ~]# mkfs.ext4 /dev/vgdb/bin
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@dbps-svr1 ~]# mkfs.ext4 /dev/vgdb/log
.

[root@dbps-svr1 ~]# mkfs.ext4 /dev/vgdb/dba
.

[root@dbps-svr1 ~]# mkfs.ext4 /dev/vgdb/arch
.

[root@dbps-svr1 ~]# mkdir /tibero
[root@dbps-svr1 ~]# mkdir /tiblog
[root@dbps-svr1 ~]# mkdir /tibdba
[root@dbps-svr1 ~]# mkdir /tibarch
[root@dbps-svr1 ~]# mount /dev/vgdb/bin /tibero
[root@dbps-svr1 ~]# mount /dev/vgdb/log /tiblog
[root@dbps-svr1 ~]# mount /dev/vgdb/dba /tibdba

[root@dbps-svr1 ~]# mount /dev/vgdb/arch /tibarch
# 재기동 후에도 local disk volume에 대해서는 자동으로 mount 되도록 다음 파일을 설정합니다.
[root@dbps-svr1 ~]# vi /etc/fstab
[root@dbps-svr1 ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Tue Oct 24 09:39:26 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/vglocal-root /                       xfs     defaults        0 0
UUID=1a292047-de7d-4a0e-817d-dd60d1e0d158 /boot                   xfs     defaults        0 0
/dev/mapper/vglocal-swap swap                    swap    defaults        0 0
/dev/vgdb/bin           /tibero                 ext4    defaults    0       0
/dev/vgdb/log           /tiblog                 ext4    defaults    0       0
/dev/vgdb/dba           /tibdba                 ext4    defaults    0       0
/dev/vgdb/arch          /tibarch                ext4    defaults    0       0
[root@dbps-svr1 ~]#
```

2) iscsi client 작업

```
[root@dbps-svr1 ~]# yum -y install iscsid
.
[root@dbps-svr1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

##storage server ip
10.10.10.40             target.dbps-sto

##db server ip
#interconnect
10.10.20.41             finsvr1_net
10.10.20.42             finsvr2_net
#public
172.16.228.41           finsvr1_pub             finsvr1
172.16.228.42           finsvr2_pub             finsvr2
#virtual
172.16.228.141          finsvr1_vip
172.16.228.142          finsvr2_vip

[root@dbps-svr1 ~]# iscsiadm -m discovery -t st -p target.dbps-sto 10.10.10.40:3260,1
iqn.2017-10.dbps-sto.target:disk1
# 앞서 dbps-sto 서버에서 설정한 auth userid,password를 다음 파일(/etc/iscsid/iscsid.conf)에 설정합니다.
# authentication by the target(s), uncomment the following lines:
node.session.auth.username = root
node.session.auth.password = tibero
[root@dbps-svr1 ~]# vi /etc/iscsi/iscsid.conf
# 앞서 dbps-sto 서버에서 생성한 initiator 의 이름을 다음 파일(/etc/iscsi/initiatorname.iscsi)에 설정합니다.
[root@dbps-svr1 ~]# vi /etc/iscsi/initiatorname.iscsi
[root@dbps-svr1 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2017-10.dbps-sto.target:initiator1 (node2의 경우 iqn.2017-10.dbps-sto.target:initiator2)
[root@dbps-svr1 ~]# systemctl restart iscsid
[root@dbps-svr1 ~]# iscsiadm -m discovery -t st -p target.dbps-sto 10.10.10.40:3260,1 iqn.2017-10.dbps-sto.target:disk1
10.10.10.40:3260,1 iqn.2017-10.dbps-sto.target:disk1
[root@dbps-svr1 ~]# iscsiadm -m node -T iqn.2017-10.dbps-sto.target:disk1 -p target.dbps-sto -l
Logging in to [iface: default, target: iqn.2017-10.dbps-sto.target:disk1, portal: 10.10.10.40,3260] (multiple)
Login to [iface: default, target: iqn.2017-10.dbps-sto.target:disk1, portal: 10.10.10.40,3260] successful.
[root@dbps-svr1 ~]# cat /proc/partitions
major minor  #blocks  name

   8        0   41943040 sda
   8        1    1048576 sda1
   8        2   40893440 sda2
   8       16  335544320 sdb

   8       17  335543296 sdb1
  11        0    1048575 sr0
 253        0   32501760 dm-0
 253        1    8388608 dm-1
 253        2   41943040 dm-2
 253        3   83886080 dm-3
 253        4   41943040 dm-4
 253        5  125829120 dm-5
   8       32  670822400 sdc
```

3) iscsi storage volume 작업

```
[root@dbps-svr1 ~]# cat /proc/partitions
major minor  #blocks  name

   8        0   41943040 sda
   8        1    1048576 sda1
   8        2   40893440 sda2
   8       16  335544320 sdb
   8       17  335543296 sdb1
  11        0    1048575 sr0
 253        0   32501760 dm-0
 253        1    8388608 dm-1
 253        2   41943040 dm-2
 253        3   83886080 dm-3
 253        4   41943040 dm-4
 253        5  125829120 dm-5
   8       32  670822400 sdc
[root@dbps-svr1 ~]# fdisk /dev/sdc
.
.
[root@dbps-svr1 ~]# fdisk -l /dev/sdc1

Disk /dev/sdc1: 686.9 GB, 686921089024 bytes, 1341642752 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 524288 bytes / 1048576 bytes

[root@dbps-svr1 ~]# vgdisplay
  --- Volume group ---
  VG Name               vgsto
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               639.74 GiB
  PE Size               4.00 MiB
  Total PE              163774
  Alloc PE / Size       0 / 0
  Free  PE / Size       163774 / 639.74 GiB
  VG UUID               SXQOeB-Qbid-sWh7-9aOy-tkcr-c2Re-JTAAbb

  --- Volume group ---
  VG Name               vglocal
  .
  .
  --- Volume group ---
```

```
[root@dbps-svr1 ~]# cat lvcreate_vgsto.sh
.
lvcreate -L 1028M -n lv_cmfile1          vgsto    /dev/sdc1
.
lvcreate -L 516M -n lv_redo001           vgsto    /dev/sdc1
.
.
.
lvcreate -L 10244M -n lv_data029          vgsto    /dev/sdc1
lvcreate -L 10244M -n lv_data030          vgsto    /dev/sdc1

[root@dbps-svr1 ~]# sh lvcreate_vgsto.sh
  Logical volume "lv_cmfile1" created.
  Logical volume "lv_cmfile2" created.
  Logical volume "lv_ctlfile1" created.
  Logical volume "lv_ctlfile2" created.
  Logical volume "lv_redo001" created.
  Logical volume "lv_redo002" created.
  Logical volume "lv_redo011" created.
  Logical volume "lv_redo012" created.
  Logical volume "lv_redo021" created.
  Logical volume "lv_redo022" created.
  Logical volume "lv_redo131" created.
  Logical volume "lv_redo132" created.
  Logical volume "lv_redo141" created.

  -- 이하 생략
```

가) lvcreate_vgsto.sh

```
[root@dbps-svr2 ~]# cat lvcreate_vgsto.sh
lvcreate -L 1028M -n lv_cmfile1          vgsto      /dev/sdc1
lvcreate -L 1028M -n lv_cmfile2          vgsto      /dev/sdc1
lvcreate -L 1028M -n lv_ctlfile1         vgsto      /dev/sdc1
lvcreate -L 1028M -n lv_ctlfile2         vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo001           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo002           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo011           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo012           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo021           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo022           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo131           vgsto      /dev/sdc1
lvcreate -L 516M -n lv_redo132           vgsto      /dev/sdc1

-- 이하 생략
```

4) raw device mapping 및 hard link 설정

가) raw device mapping구성합니다.

6.나.3) volume 작업에서 생성한 lv (block device)를 character device 로 mapping 합니다.

```
[root@dbps-svr1 ~]# cat raw.sh
raw /dev/raw/raw1 /dev/vgsto/lv_cmfile1
raw /dev/raw/raw2 /dev/vgsto/lv_cmfile2
raw /dev/raw/raw3 /dev/vgsto/lv_ctlfile1
raw /dev/raw/raw4 /dev/vgsto/lv_ctlfile2
raw /dev/raw/raw5 /dev/vgsto/lv_redo001
raw /dev/raw/raw6 /dev/vgsto/lv_redo002
raw /dev/raw/raw7 /dev/vgsto/lv_redo011
raw /dev/raw/raw8 /dev/vgsto/lv_redo012
raw /dev/raw/raw9 /dev/vgsto/lv_redo021
raw /dev/raw/raw10 /dev/vgsto/lv_redo022
raw /dev/raw/raw11 /dev/vgsto/lv_redo131
raw /dev/raw/raw12 /dev/vgsto/lv_redo132
```

-- 이하 생략

```
[root@dbps-svr1 ~]# sh raw.sh
/dev/raw/raw1:  bound to major 253, minor 6
/dev/raw/raw2:  bound to major 253, minor 7
/dev/raw/raw3:  bound to major 253, minor 8
/dev/raw/raw4:  bound to major 253, minor 9
/dev/raw/raw5:  bound to major 253, minor 10
/dev/raw/raw6:  bound to major 253, minor 11
/dev/raw/raw7:  bound to major 253, minor 12
/dev/raw/raw8:  bound to major 253, minor 13
/dev/raw/raw9:  bound to major 253, minor 14
/dev/raw/raw10:  bound to major 253, minor 15
/dev/raw/raw11:  bound to major 253, minor 16
/dev/raw/raw12:  bound to major 253, minor 17
/dev/raw/raw13:  bound to major 253, minor 18
/dev/raw/raw14:  bound to major 253, minor 19
/dev/raw/raw15:  bound to major 253, minor 20
/dev/raw/raw16:  bound to major 253, minor 21
/dev/raw/raw17:  bound to major 253, minor 22
```

-- 이하 생략

```
# raw map 확인
[root@dbps-svr1 ~]# raw -qa
/dev/raw/raw1:  bound to major 253, minor 6
/dev/raw/raw2:  bound to major 253, minor 7
/dev/raw/raw3:  bound to major 253, minor 8
/dev/raw/raw4:  bound to major 253, minor 9
/dev/raw/raw5:  bound to major 253, minor 10
/dev/raw/raw6:  bound to major 253, minor 11
/dev/raw/raw7:  bound to major 253, minor 12
/dev/raw/raw8:  bound to major 253, minor 13
/dev/raw/raw9:  bound to major 253, minor 14
/dev/raw/raw10:  bound to major 253, minor 15
/dev/raw/raw11:  bound to major 253, minor 16
/dev/raw/raw12:  bound to major 253, minor 17
/dev/raw/raw13:  bound to major 253, minor 18
/dev/raw/raw14:  bound to major 253, minor 19
```

-- 이하 생략

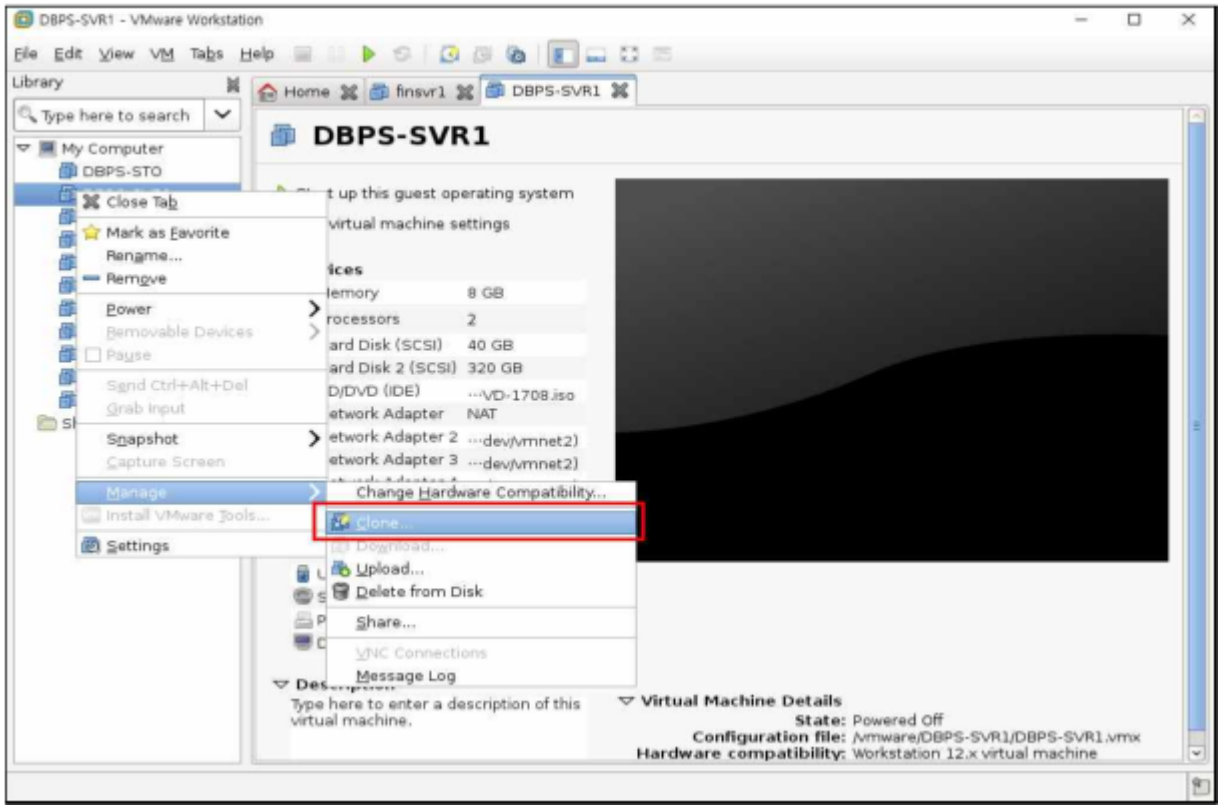
나) hard link

6.나.4.가) raw-device-mapping 에서 /dev/raw/raw# 경로로 mapping 된 raw device를 /dev/VG_NAME/. 경로에 식별이 가능하도록 hard link 설정을 합니다. 앞선 raw 명령에서 binding한 raw device#와 target lv (block device)가 정확하게 mapping 될 수 있도록 주의해야 하며, LV_NAME 에 raw device를 의미하는 알파벳 'r'을 붙여 raw device의 hard link를 생성합니다.

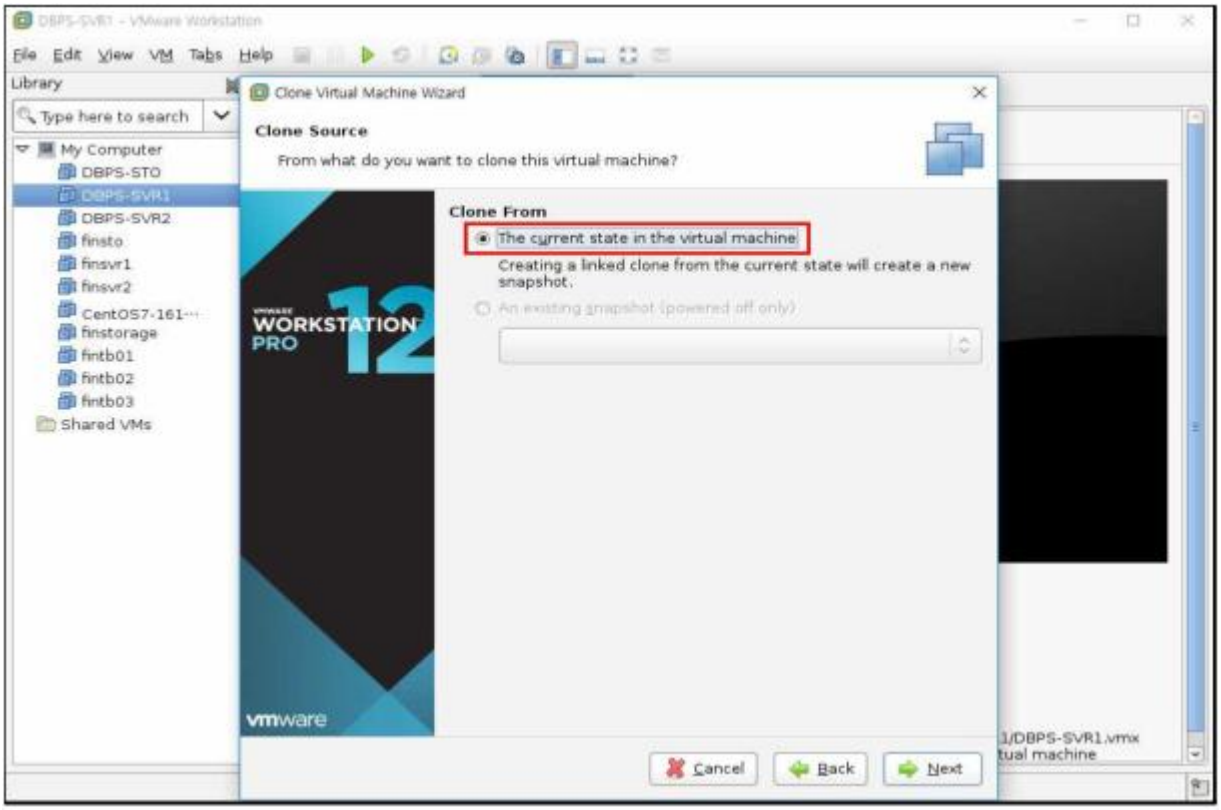
```
[root@dbps-svr1 ~]# cat link.sh
ln -Pf /dev/raw/raw1   /dev/vgsto/rlv_cmfile1
ln -Pf /dev/raw/raw2   /dev/vgsto/rlv_cmfile2
ln -Pf /dev/raw/raw3   /dev/vgsto/rlv_ctlfile1
ln -Pf /dev/raw/raw4   /dev/vgsto/rlv_ctlfile2
ln -Pf /dev/raw/raw5   /dev/vgsto/rlv_redo001
ln -Pf /dev/raw/raw6   /dev/vgsto/rlv_redo002
ln -Pf /dev/raw/raw7   /dev/vgsto/rlv_redo011
ln -Pf /dev/raw/raw8   /dev/vgsto/rlv_redo012
ln -Pf /dev/raw/raw9   /dev/vgsto/rlv_redo021
ln -Pf /dev/raw/raw10   /dev/vgsto/rlv_redo022
ln -Pf /dev/raw/raw11   /dev/vgsto/rlv_redo131
ln -Pf /dev/raw/raw12   /dev/vgsto/rlv_redo132
ln -Pf /dev/raw/raw13   /dev/vgsto/rlv_redo141
ln -Pf /dev/raw/raw14   /dev/vgsto/rlv_redo142
ln -Pf /dev/raw/raw15   /dev/vgsto/rlv_redo151
ln -Pf /dev/raw/raw16   /dev/vgsto/rlv_redo152
ln -Pf /dev/raw/raw17   /dev/vgsto/rlv_redo261
ln -Pf /dev/raw/raw18   /dev/vgsto/rlv_redo262
ln -Pf /dev/raw/raw19   /dev/vgsto/rlv_redo271

-- 중략

ln -Pf /dev/raw/raw62   /dev/vgsto/rlv_data028
ln -Pf /dev/raw/raw63   /dev/vgsto/rlv_data029
ln -Pf /dev/raw/raw64   /dev/vgsto/rlv_data030
[root@dbps-svr1 ~]# sh link.sh
```

다) 서버 복제

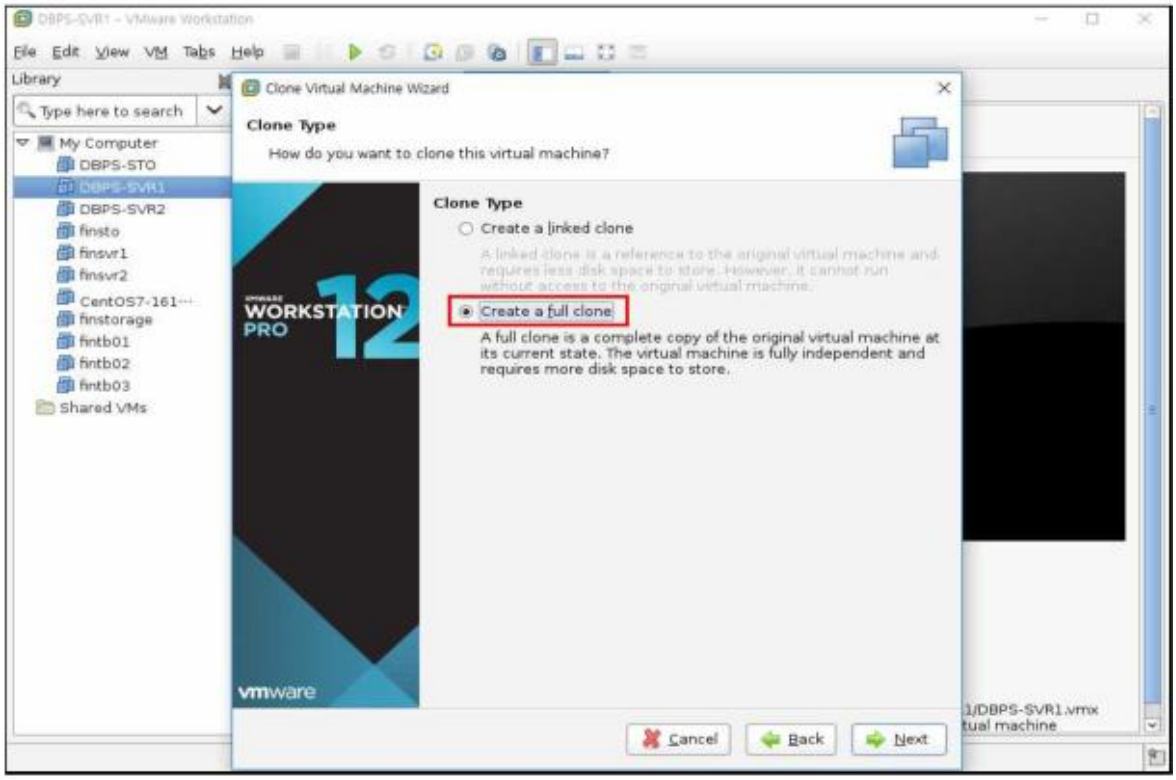
실제 물리 서버 구성인 경우 6. DB 서버 구축 의 절차를 한 번 더 진행하면 됩니다. 테스트는 vmware에서 진행되었기 때문에 편의상 dbps-svr1 가상머신을 복제하도록 하겠습니다. vmware의 clone 기능의 경우 복제 대상의 가상머신이 종료된 상태에서 가능합니다. 다음과 같은 절차로 복제 를 진행합니다.



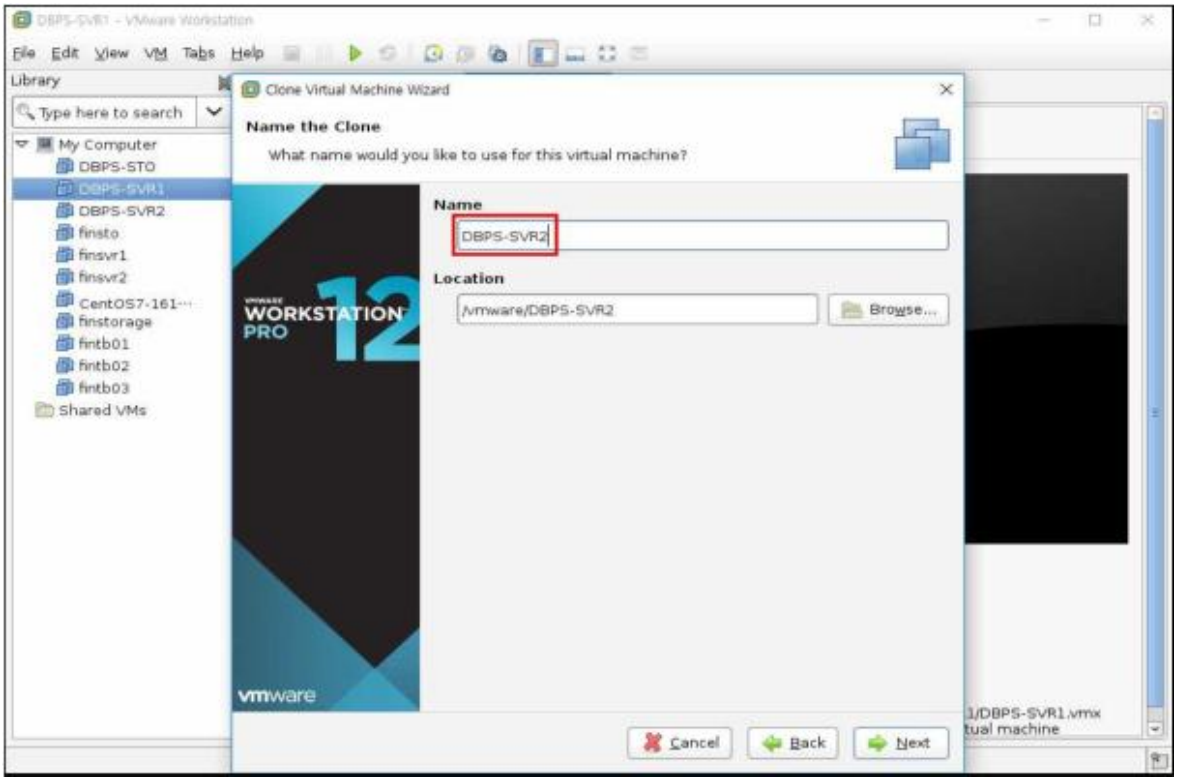
2) The Current state in the virtual machine 선택



3) Create a full clone 선택



4) Name 변경. 기존 가상 머신에서 host id number만 변경



5) 복제 이후 node 2 머신만 기동한 뒤, '/etc/sysconfig/network-script/ifcfg-*' 파일의 IPADDR 항목을 변경하고, '/etc/iscsi/initiatorname.iscsi' 파일의 initiatorName을 노드#에 맞게 수정합니다.

6) DB Server 의 경우, systemctl enable iscsid 설정으로 iSCSI target server에 자동 로그인은 되지만 raw device mapping 및 Hard link 설정은 자동으로 되지 않습니다. network 기동 및 iSCSI target server login 이후에 mapping이 되어야 하므로, 재기동 이후, 수동으 로 다음 스크립트를 실행합니다.

```
[root@dbps-svr1 ~]# cat raw_mapping.sh
raw /dev/raw/raw1 /dev/vgsto/lv_cmfile1
raw /dev/raw/raw2 /dev/vgsto/lv_cmfile2
raw /dev/raw/raw3 /dev/vgsto/lv_ctlfile1
raw /dev/raw/raw4 /dev/vgsto/lv_ctlfile2
raw /dev/raw/raw5 /dev/vgsto/lv_redo001
raw /dev/raw/raw6 /dev/vgsto/lv_redo002
raw /dev/raw/raw7 /dev/vgsto/lv_redo011
raw /dev/raw/raw8 /dev/vgsto/lv_redo012
raw /dev/raw/raw9 /dev/vgsto/lv_redo021
raw /dev/raw/raw10 /dev/vgsto/lv_redo022
raw /dev/raw/raw11 /dev/vgsto/lv_redo131
raw /dev/raw/raw12 /dev/vgsto/lv_redo132

-- 중략

# tibero 프로세스가 실행되는 사용자에게 권한 부여
chown tibero:dba /dev/vgsto/rlv_*
[root@dbps-svr1 ~]# sh raw_mapping.sh
```

7. TAC 설치

7.1 설치 전 준비사항

7.1.1 바이너리 업로드 및 압축 해제

- 본 테스트에서는 Technet에서 배포중인 Tibero7.2.4 바이너리를 사용한다.

```
$ ls
tibero7-bin-FS02_PS04-linux64_4.18-303667-20251013015219.tar.gz
$ tar -zxf tibero7-bin-FS02_PS04-linux64_4.18-303667-20251013015219.tar.gz
```

7.1.2 라이선스 설정

- Technet을 통해 받은 데모라이선스 파일을 $TB_HOME/license/license.xml 위치 및 파일 명으로 지정한다.

```
$ ls *.xml
Tibero_demo_test01.xml      <-  발급 받은 데모 라이선스 파일
$ mv Tibero_demo_test01.xml $TB_HOME/license/license.xml
$ ls $TB_HOME/license/license.xml  <- 파일 명을 "license.xml" 으로 변경해야 함
/home/tibero/tibero7/license/license.xml
```

7.1.3 OS 커널 파라미터 설정

- 테스트 환경이 아닌 타 OS는 "Tibero_7_Installation-Guide_v7.2.4.pdf"공식 문서의 "2.5 운영체제별 파라미터 설정"을 참고

- 붉은색 글씨는 제거

---

vi /etc/sysctl.conf

kernel.shmmax = 1073741824 //TOTAL_SHM_SIZE 이상 (byte)
kernel.sem = 10000 32000 10000 10000

순서대로 SEMMSL, SEMMNS, SEMOPM, SEMMNI 최소 설정값은 아래와 같다.
kernel.sem
– SEMMSL : 2 / 권장값 : (Tibero 전체 Thread 수) x 2
– SEMMNS : (Tibero 전체 Thread 수) x 2
– SEMOPM : 2(=SEMMSL) / 권장값 : (Tibero 전체 Thread 수) x 2
– SEMMNI : (Tibero 전체 Thread 수)
max값을 높게 설정해도 큰 문제가 없기에 여유롭게 설정하는 것을 권장한다.

kernel.shmall = 262144 //ceil(shmmax/PAGE_SIZE)
kernel.shmmni = 4096
fs.file-max = 67108864
(nofile 파라미터) x (WTHR_PROC_CNT + PEP_PROC_CNT) 또는 67108864 으로 설정

fs.nr_open = nofile 파라미터 이상으로 설정해야만 한다.
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range=1024 65000
net.core.rmem_default = 262144, OS 기존 설정값 중 최댓값
net.core.wmem_default = 262144, OS 기존 설정값 중 최댓값
net.core.rmem_max= 67108864, OS 기존 설정값, TCP_RCVBUF_SIZE, _INC_TCP_RCVBUF_SIZE 중 최댓값
net.core.wmem_max= 67108864, OS 기존 설정값, TCP_SNDBUF_SIZE, _INC_TCP_SNDBUF_SIZE 중 최댓값

/etc/security/limits.conf

tibero soft nofile 65536
tibero hard nofile 65536
tibero soft nproc 65536
tibero hard nproc 65536

nofile = (WTHR_PER_PROC * ((total data files in db) + 15)) + (tbsvr processcount + 5) + 100
또는 크게 3,000,000 으로 설정
nproc = MAX_SESSION_COUNT+10000 이상으로 설정
nofile <= fs.nr_open <= fs.file-max

7.2 설치

7.2.1 설정 내역

- 해당 문서는 아래와 같은 정보로 TAC를 구성한다
  사전에 아래의 정보를 고객사에 전달하여 설치 정보를 전달 받고 방화벽에 inbound, outbound 해제 여부를 확인 후 방문한다면 설치 대기시간을 최소화 할 수 있다.

	항 목	NODE1	NODE2	default	비고
OS	Hostname	test01	test02		각 서버의 hostname
	User	tibero	tibero		Tibero가 설치될 OS 유저
TAC	SID	TEST1	TEST2		Tibero Instance Name
	DB_NAME	TEST			Tibero Database Name
	LISTENER_PORT	8629	8629	8629	Tibero Listener port
	LOCAL_CLUSTER_ADDR	10.10.10.1	10.10.10.2		Tibero node간 통신 IP
	LOCAL_CLUSTER_PORT	11029	11029	11029	Tibero node간 통신 port
	CM_PORT	11039	11039	11039	cmrctl 명령어를 실행시 Tibero instance의 상태 체크를 위해 tibero에 접속하는 포트
CM	SID	CM0	CM1		TBCM SID
	CM Inter-Connect PORT	11019	11019	11019	Inter-connect Port (tbcm node간 통신 port) (cmrctl add 명령시 사용 port)
	CM_UI_PORT	11039	11039	11039	cmrctl 명령어를 실행시 tbcm의 상태 체크를 위해 tbcm에 접속하는 포트

■ 각 port 용도 및 설정 방법 설명
- LISTENER_PORT : DB listener port. $TB_SID.tip 파일에 설정
- LOCAL_CLUSTER_PORT : Tibero node간 통신을 위한 Inter-Connect port. $TB_SID.tip 파일에 설정
- CM_PORT : Tibero instance 에서 tbcm 에 접속하기 위한 port. $TB_SID.tip 파일에 설정
- CM Inter-Connect PORT : 각 서버의 CM간 통신을 위한 Inter-Connect port. cmrctl 에서 private network 를 추가할 때 설정
- CM_UI_PORT : tbcm 에서 접속을 허용하는 용도로 사용하는 port. $CM_SID.tip 파일에 설정

※ $TB_SID.tip 의 CM_PORT 와 $CM_SID.tip 의 CM_UI_PORT 는 동일하게 설정해야 한다

7.2.2 1번 Node 환경 변수 설정

- 환경변수의 경우 기존 Tibero를 설치할 때의 변수에 CM 관련 환경변수와 편의성을 위한 alias 만 추가된다.

```
### Tibero7 ENV ###
export TB_HOME=/home/tibero/tibero7
export TB_SID=test1                                  ## In case of NODE2, it will be test2
export PATH=$TB_HOME/bin:$TB_HOME/client/bin:$PATH

### Linux, Solaris ###
```

```
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib:$LD_LIBRARY_PATH
### AIX ###
export LIBPATH=$TB_HOME/lib:$TB_HOME/client/lib:$LIBPATH

### Tibero Cluster Manager ENV ###
export CM_HOME=$TB_HOME
export CM_SID=cm0                                   ## In case of NODE2, it will be cm1
### Alias For Tibero ###
alias tbhome='cd $TB_HOME'
alias tbcfg='cd $TB_HOME/config'
alias tbcfgv='vi $TB_HOME/config/$TB_SID.tip'
alias tbcli='cd $TB_HOME/client/config'
alias tbcliv='vi $TB_HOME/client/config/tbdsn.tbr'
alias tblog='cd $TB_HOME/instance/$TB_SID/log'
alias tbi='cd /idiska/expert/tbinary'
alias tm='cd /idiska/expert/tbinary/monitor;./monitor;cd -'
alias ll='ls -alF'
### Alias For tbcm ###
alias cmcfgv='vi $CM_HOME/config/$CM_SID.tip'
alias cmlog='cd $CM_HOME/instance/$CM_SID/log'
alias show='cmrctl show'
```

7.2.3 2번 Node 환경 변수 설정

- 환경변수의 경우 기존 Tibero 를 설치할 때의 변수에 CM 관련 환경변수와 편의성을 위한 alias 만 추가된다.

```
### Tibero7 ENV ###
export TB_HOME=/home/tibero/tibero7
export TB_SID=test2                                 ## In case of NODE1, it will be test1
export PATH=$TB_HOME/bin:$TB_HOME/client/bin:$PATH

### Linux, Solaris ###
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib:$LD_LIBRARY_PATH
### AIX ###
export LIBPATH=$TB_HOME/lib:$TB_HOME/client/lib:$LIBPATH

### Tibero Cluster Manager ENV ###
export CM_HOME=$TB_HOME
export CM_SID=cm1                                   ## In case of NODE1, it will be cm0
### Alias For Tibero ###
```

```
alias tbhome='cd $TB_HOME'
alias tbcfg='cd $TB_HOME/config'
alias tbcfgv='vi $TB_HOME/config/$TB_SID.tip'
alias tbcli='cd $TB_HOME/client/config'
alias tbcliv='vi $TB_HOME/client/config/tbdsn.tbr'
alias tblog='cd $TB_HOME/instance/$TB_SID/log'
alias tbi='cd /idiska/expert/tbinary'
alias tm='cd /idiska/expert/tbinary/monitor;./monitor;cd -'
alias ll='ls -alF'
### Alias For tbcm ###
alias cmcfgv='vi $CM_HOME/config/$CM_SID.tip'
alias cmlog='cd $CM_HOME/instance/$CM_SID/log'
alias show='cmrctl show'
```

7.2.4 1번 Node의 $TB_SID.tip 파일 생성 및 수정

- 환경변수의 경우 기존 Tibero 를 설치할 때의 변수에 CM 관련 환경변수와 편의성을 위한 alias 만 추가된다.

```
$ sh $TB_HOME/config/gen_tip.sh
Using TB_SID "TEST1"
/home/tibero/tibero7/config/TEST1.tip generated
/home/tibero/tibero7/config/psm_commands generated
/home/tibero/tibero7/client/config/tbdsn.tbr generated.
Running client/config/gen_esql_cfg.sh
Done.
$ vi /home/tibero/tibero7/config/TEST1.tip

DB_NAME=TEST
LISTENER_PORT=8629
CONTROL_FILES="/tbdata/DB_TEST/ControlFile1/c1.ctl","/tbdata/DB_TEST/ControlFile2/c2.ctl"
DB_CREATE_FILE_DEST=/tbdata/DB_TEST/DataFile/

MAX_SESSION_COUNT=20

TOTAL_SHM_SIZE=2G
MEMORY_TARGET=4G
THREAD=0
UNDO_TABLESPACE=UNDO0

CLUSTER_DATABASE=Y
LOCAL_CLUSTER_ADDR=10.10.10.1
```

```
LOCAL_CLUSTER_PORT=11029
CM_PORT=11039
```

7.2.5 2번 Node의 $TB_SID.tip 파일 생성 및 수정

- 환경변수의 경우 기존 Tibero 를 설치할 때의 변수에 CM 관련 환경변수와 편의성을 위한 alias 만 추가된다.

```
$ sh $TB_HOME/config/gen_tip.sh
Using TB_SID "TEST2"
/home/tibero/tibero7/config/TEST2.tip generated
/home/tibero/tibero7/config/psm_commands generated
/home/tibero/tibero7/client/config/tbdsn.tbr generated.
Running client/config/gen_esql_cfg.sh
Done.
$ vi /home/tibero/tibero7/config/TEST2.tip

DB_NAME=TEST
LISTENER_PORT=8629
CONTROL_FILES="/tbdata/DB_TEST/ControlFile1/c1.ctl","/tbdata/DB_TEST/ControlFile2/c2.ctl"
DB_CREATE_FILE_DEST=/tbdata/DB_TEST/DataFile/

MAX_SESSION_COUNT=20

TOTAL_SHM_SIZE=2G
MEMORY_TARGET=4G

THREAD=1
UNDO_TABLESPACE=UNDO1

CLUSTER_DATABASE=Y
LOCAL_CLUSTER_ADDR=10.10.10.2
LOCAL_CLUSTER_PORT=11029
CM_PORT=11039
```

7.2.6 CM tip파일 생성

- $TB_HOME/config/cm.template 파일을 참조하여 동일한 위치에 $CM_SID.tip 형태로 설정 파일을 생성한다.

※ $TB_SID.tip 의 CM_PORT 와 $CM_SID.tip 의 CM_UI_PORT 는 동일하게 설정해야 한다.

1번 노드

```
vi $CM_HOME/config/$CM_SID.tip
```

```
CM_NAME=CM0
CM_UI_PORT=11039
CM_RESOURCE_FILE=/tbdata/DB_TEST/CM/CM0/CM0_Res.crf

CM_LOG_DEST=/tbdata/DB_TEST/CM/CM0/LOG
CM_GUARD_LOG_DEST=/tbdata/DB_TEST/CM/CM0/G_LOG
```

2번 노드

```
vi $CM_HOME/config/$CM_SID.tip

CM_NAME=CM1
CM_UI_PORT=11039
CM_RESOURCE_FILE=/tbdata/DB_TEST/CM/CM1/CM1_Res.crf

CM_LOG_DEST=/tbdata/DB_TEST/CM/CM1/LOG
CM_GUARD_LOG_DEST=/tbdata/DB_TEST/CM/CM1/G_LOG
```

7.2.7 tbdsn.tbr 수정

Node 1

```
$vi $TB_HOME/client/config/tbdsn.tbr

TEST1=(
    (INSTANCE=(HOST=localhost)
            (PORT=8629)
            (DB_NAME=TEST)
    )
)

TEST=(
    (INSTANCE=(HOST=192.168.0.141)
            (PORT=8629)
            (DB_NAME=TEST)
    )
    (INSTANCE=(HOST=192.168.0.142)
            (PORT=8629)
            (DB_NAME=TEST)
    )
)
```

Node 2

```
$vi $TB_HOME/client/config/tbdsn.tbr

TEST2=(
    (INSTANCE=(HOST=localhost)
            (PORT=8629)
            (DB_NAME=TEST)
    )
)

TEST=(
    (INSTANCE=(HOST=192.168.0.141)
            (PORT=8629)
            (DB_NAME=TEST)
    )
    (INSTANCE=(HOST=192.168.0.142)
            (PORT=8629)
            (DB_NAME=TEST)
    )
)
```

7.2.8 Node 1번 – tbcm 기동

- $TB_HOME

```
$ tbcm -b
CM Guard daemon started up.

TBCM 7.1.1 (Build 303667)

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Tibero cluster manager started up.
Local node name is (CM0:11039).
```

7.2.9 Node 1번 – Network 등록

- public network 추가시 public IP가 설정된 이더넷 이름을 등록해야한다. 이는 ifconfig -a 를 통해 확인한다.

```
$ ifconfig -a
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.141  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7656:3cff:fe2a:729b  prefixlen 64  scopeid 0x20<link>
        ether 74:56:3c:2a:72:9b  txqueuelen 1000  (Ethernet)
        RX packets 144070427  bytes 30858206257 (28.7 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 98371456  bytes 14494596654 (13.4 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Private Network add command

$ cmrctl add network --name <network_name> --nettype <public/private> --ipaddr <inter-connect IP> --portno <CM 통신 port>

- Private Network add ex

$ cmrctl add network --name **test_inter0** --nettype private --ipaddr 10.10.10.1 --portno 11019

- Public Network add command

$ cmrctl add network --name <network_name> --nettype <public/private> --ifname <NIC_NAME>

- Public Network add ex

$ cmrctl add network --name **test_pub0** --nettype public --ifname enp4s0

- 결과 확인

$ cmrctl show

Resource List of Node CM0

```
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON   network       test_pub0         UP (public) enp4s0
================================================================================
```

7.2.10 Node 1번 – Cluster 등록

- public network

- Cluster add command

$ cmrctl add cluster --name <cluster_name> --incnet <private_network_name> --pubnet <public_network_name> --cfile "<cfile_name1>"

- Cluster add ex

$cmrctl add cluster --name test_cls --incnet **test_inter0** --pubnet **test_pub0** --cfile "/tbdata/DB_TEST/CM/CM0/CM0File.cm, /tbdata/DB_TEST/CM/CM1/CM1File.cm"

※ cluster file(cfile)은 위의 예시와 같이 이중화 구성을 권장한다.

- 결과 확인

$ cmrctl show

Resource List of Node CM0

```
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON   network       test_pub0         UP (public) enp4s0
     COMMON   cluster        test_cls      DOWN inc: test_inter0, pub: test_pub0
================================================================================
```

- cluster start

$ cmrctl start cluster --name test_cls

```
- 결과 확인
$ cmrctl show
Resource List of Node CM0
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON  network       test_pub0        UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter0, pub: test_pub0
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
================================================================================
```

7.2.11 Node 1번 – DB 등록

```
- Service add command
$ cmrctl add service --name <database_name> --cname <cluster_name>

- Service add ex
$ cmrctl add service --name TEST --cname test_cls

- Database add command
$ cmrctl add db --name <instance_name> --svcname <database_name> --dbhome <tibero_home_directory>

- Database add ex
$ cmrctl add db --name TEST1 --svcname TEST --dbhome $TB_HOME

- 결과 확인
$ cmrctl show
Resource List of Node CM0
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON  network       test_pub0        UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter0, pub: test_pub0
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls   service            TEST      DOWN Database, Active Cluster (auto-restart: OFF)
```

```
   test_cls       db          TEST1      DOWN TEST, /home/tibero/tibero7, failed retry cnt: 0
========================================================================
```

7.2.12 Node 1번 – Database 생성 및 기동

```
- tbboot nomount mode command
$ tbboot nomount 또는 cmrctl start db --name <instance_name> --option "-t <boot_mode>"

- tbboot nomount mode ex
$ tbboot nomount 또는 cmrctl start db --name TEST1 --option "-t nomount"

- 결과 확인
$ cmrctl show
Resource List of Node CM0
========================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON   network      test_pub0         UP (public) enp4s0
     COMMON   cluster        test_cls        UP inc: test_inter0, pub: test_pub0
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls   service            TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls        db           TEST1 UP(NMNT) TEST, /home/tibero/tibero7, failed retry cnt: 0
========================================================================

- Database 생성
$ tbsql sys/tibero
SQL> create database "TEST"
  user sys identified by tibero
  NOARCHIVELOG
  MAXLOGFILES 255
  MAXLOGMEMBERS 8
  MAXDATAFILES 4096
  MAXARCHIVELOG 500
  MAXBACKUPSET 500
  MAXLOGHISTORY 500
  CHARACTER SET UTF8
  NATIONAL CHARACTER SET UTF16
```

```
  logfile
     group 0 ('/tbdata/DB_TEST/Redo1/Redo001.redo', '/tbdata/DB_TEST/Redo2/Redo002.redo') size 100M,
     group 1 ('/tbdata/DB_TEST/Redo1/Redo011.redo', '/tbdata/DB_TEST/Redo2/Redo012.redo') size 100M,
     group 2 ('/tbdata/DB_TEST/Redo1/Redo021.redo', '/tbdata/DB_TEST/Redo2/Redo022.redo') size 100M,
     group 3 ('/tbdata/DB_TEST/Redo1/Redo031.redo', '/tbdata/DB_TEST/Redo2/Redo032.redo') size 100M,
     group 4 ('/tbdata/DB_TEST/Redo1/Redo041.redo', '/tbdata/DB_TEST/Redo2/Redo042.redo') size 100M
  datafile '/tbdata/DB_TEST/DataFile/system001.dtf' size 3G
    autoextend on next 32M maxsize unlimited
  SYSSUB
    datafile '/tbdata/DB_TEST/DataFile/syssub001.dtf' size 3G
    autoextend on next 32M maxsize unlimited
  default temporary tablespace TEMP
    tempfile '/tbdata/DB_TEST/DataFile/temp001.dtf' size 3G
    autoextend on next 32M maxsize unlimited
    extent management local autoallocate
  undo tablespace UNDO0
    datafile '/tbdata/DB_TEST/DataFile/undo001.dtf' size 3G
    autoextend on next 32M maxsize unlimited
    extent management local autoallocate
  default tablespace USR
    datafile  '/tbdata/DB_TEST/DataFile/usr001.dtf' size 3G
    autoextend on next 32M maxsize unlimited
    extent management local autoallocate
;

- tbboot normal mode command
$ tbboot 또는 cmrctl start db --name <instance_name>

- tbboot normal mode command
$ tbboot 또는 cmrctl start db --name TEST1

- 결과 확인
$ cmrctl show
Resource List of Node CM0
================================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON  network      test_pub0         UP (public) enp4s0
     COMMON  cluster       test_cls        UP inc: test_inter0, pub: test_pub0
```

```
   test_cls     file     test_cls:0       UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls     file     test_cls:1       UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls  service          TEST       UP Database, Active Cluster (auto-restart: OFF)
   test_cls       db         TEST1 UP(NRML) TEST, /home/tibero/tibero7, failed retry cnt: 0
================================================================================
```

7.2.13 Node 1번 – Node 2번에서 사용할 Undo Tablespace 및 Redo log 추가

```
- Undo tablespace create
SQL> create undo tablespace UNDO1
        datafile '/tbdata/DB_TEST/DataFile/undo101.dtf' size 3G
        extent management local AUTOALLOCATE;

- Redo Log add
SQL> alter database add logfile thread 1 group 5
('/tbdata/DB_TEST/Redo1/Redo151.redo', '/tbdata/DB_TEST/Redo2/Redo152.redo' ) size 100M;

 alter database add logfile thread 1 group 6
 ('/tbdata/DB_TEST/Redo1/Redo161.redo', '/tbdata/DB_TEST/Redo2/Redo162.redo' ) size 100M;
 alter database add logfile thread 1 group 7
 ('/tbdata/DB_TEST/Redo1/Redo171.redo', '/tbdata/DB_TEST/Redo2/Redo172.redo' ) size 100M;
 alter database add logfile thread 1 group 8
 ('/tbdata/DB_TEST/Redo1/Redo181.redo', '/tbdata/DB_TEST/Redo2/Redo182.redo' ) size 100M;
 alter database add logfile thread 1 group 9
 ('/tbdata/DB_TEST/Redo1/Redo191.redo', '/tbdata/DB_TEST/Redo2/Redo192.redo' ) size 100M;

- thread enable 처리
SQL> alter database enable public thread 1;
```

7.2.14 Node 1번 – System.sh 수행

```
$ sh $TB_HOME/scripts/system_install.sh -p1 tibero -p2 syscat
Creating internal system tables...
Creating additional system index...
Creating client policy table ...
Setting case insensitive password...
Create system tables related to dbms_redact...
```

```
Creating text packages table ...
Inserting for text packages table ...
Creating the role DBA...
```

7.2.15 Node 2번 – tbcm 기동

```
$ tbcm -b
CM Guard daemon started up.

TBCM 7.1.1 (Build 303667)

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Tibero cluster manager started up.
Local node name is (CM1:11039).
```

7.2.16 Node 2번 – Network 등록

- Node1에서 수행했던 방식과 동일

```
$ifconfig -a
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.142  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7656:3cff:fe94:de86  prefixlen 64  scopeid 0x20<link>
        ether 74:56:3c:94:de:86  txqueuelen 1000  (Ethernet)
        RX packets 174188677  bytes 39457095902 (36.7 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77396259  bytes 11089180430 (10.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Private Network add command

$ cmrctl add network --name <network_name> --nettype <public/private> --ipaddr <inter-connect IP> --portno <CM 통신 port>

- Private Network add ex

$ cmrctl add network --name **test_inter1** --nettype private --ipaddr 10.10.10.2 --portno 11019

- Public Network add command

$ cmrctl add network --name <network_name> --nettype <public/private> --ifname <NIC_NAME>

- Public Network add ex

$ cmrctl add network --name **test_pub1** --nettype public --ifname enp4s0

- 결과 확인

$ cmrctl show

Resource List of Node CM1

```
================================================================================
  CLUSTER      TYPE         NAME        STATUS           DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON   network      test_pub1         UP (public) enp4s0
================================================================================
```

7.2.17 Node 2번 – Cluster 등록

- Cluster add command

$ cmrctl add cluster --name <cluster_name> --incnet <private_network_name> --pubnet <public_network_name> --cfile "<cfile_name1>"

※ cluster_name 은 1번 Node에서 등록했던 name과 동일해야 cluster의 정보를 읽을 수 있다.

- Cluster add ex

$cmrctl add cluster --name test_cls --incnet **test_inter1** --pubnet **test_pub1** --cfile "/tbdata/DB_TEST/CM/CM0/CM0File.cm, /tbdata/DB_TEST/CM/CM1/CM1File.cm"

※ cluster file(cfile)은 위의 예시와 같이 이중화 구성을 권장한다.

- 결과 확인

$ cmrctl show

Resource List of Node CM1

```
================================================================================
  CLUSTER      TYPE         NAME        STATUS           DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON   network      test_pub1         UP (public) enp4s0
     COMMON   cluster       test_cls      DOWN inc: test_inter1, pub: test_pub1
================================================================================
```

- cluster start

$ cmrctl start cluster --name test_cls

- 결과 확인

$ cmrctl show

```
Resource List of Node CM1
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
------------ -------- --------------- -------- ------------------------
     COMMON   network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON   network       test_pub1         UP (public) enp4s0
     COMMON   cluster        test_cls        UP inc: test_inter1, pub: test_pub1
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls   service            TEST        UP Database, Active Cluster (auto-restart: OFF)
================================================================================
```

※ 1번 Node에서 등록했던 service 항목이 있어야 정상적으로 처리된 것이다.
만약 service 항목이 보이지 않는다면 설정을 확인 후 cluster를 재등록 해야 한다.

7.2.18 Node 2번 – DB 등록 및 기동

- Database add command

```
$ cmrctl add db --name <instance_name> --svcname <database_name> --dbhome <tibero_home_directory>
```

- Database add ex

```
$ cmrctl add db --name TEST2 --svcname TEST --dbhome $TB_HOME
```

- 결과 확인

```
$ cmrctl show
Resource List of Node CM1
================================================================================
  CLUSTER      TYPE         NAME         STATUS            DETAIL
------------ -------- --------------- -------- ------------------------
     COMMON   network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON   network       test_pub1         UP (public) enp4s0
     COMMON   cluster        test_cls        UP inc: test_inter1, pub: test_pub1
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls   service            TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls        db           TEST2      DOWN TEST, /home/tibero/tibero7, failed retry cnt: 0
================================================================================
```

※ 1번 Node에서 service 를 등록하여 공유하기 때문에 2번 Node에서는 db 만 추가한다

- tbboot nomount mode command

$ tbboot nomount 또는 cmrctl start db --name <instance_name> --option "-t <boot_mode>"

- tbboot nomount mode ex
$ tbboot nomount 또는 cmrctl start db --name TEST2

- 결과 확인
$ cmrctl show
Resource List of Node CM1

```
==========================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON   network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON   network      test_pub1         UP (public) enp4s0
     COMMON   cluster        test_cls        UP inc: test_inter1, pub: test_pub1
   test_cls      file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls      file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls   service            TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls        db           TEST2 UP(NRML) TEST, /home/tibero/tibero7, failed retry cnt: 0
==========================================================================
```

7.3 VIP설정

- Tibero 의 VIP는 DB instance 의 boot / down 에 따라 UP / DOWN / FAIL-OVER 가 일어난다.
  본 문서의 VIP 설정은 TAC 가 설치 후 boot 되어 있는 상태에서 VIP 를 추가 설정하는 상황에 대해 설명한다.

7.3.1 VIP Resource 등록 전 작업

- VIP는 new cm이 OS의 root 유저로 실행해야 추가할 수 있다. 위에서 설정한 new cm 은 root 가 아닌 계정으로 실행되어 조건에 충족하지 않는다. 이에 따라 DB의 모든 instance down 후 tbcm을 root 유저로 실행 후 VIP를 추가해야 한다.

DB instance down command
$ cmrctl stop service --name <database_name>
- DB instance down ex
$ cmrctl stop service --name TEST
- new cm down
$ tbcm –d (각 노드 모두 실행)
- root 유저 접속
$ su –

- Tibero 환경변수 적용

$. /home/tibero/.bash_profile

- 환경변수 적용 확인

$ env

- new cm boot

$ tbcm –b (각 노드 모두 실행)

7.3.2 Node 1번 – VIP Resource 등록

- vip add command

$ cmrctl add vip --name <vip_name> --node <cm_sid> --svcname <database_name> --ipaddr <virtual_IP >

- vip add ex

$ cmrctl add vip --name test_vip0 --node CM0 --svcname TEST --ipaddr 192.168.0.143

- 결과 확인

$ cmrctl show

```
Resource List of Node CM0
================================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON  network       test_pub0        UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter0, pub: test_pub0
   test_cls     file     test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls     file     test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls  service           TEST      DOWN Database, Active Cluster (auto-restart: OFF)
   test_cls       db          TEST1      DOWN TEST, /home/tibero/tibero7, failed retry cnt: 0
   test_cls      vip       test_vip0      DOWN TEST, 192.168.0.143/Not Determined/Not Determined (1)
                                        failed retry cnt: 0
================================================================================
```

7.3.3 Node 1번 – VIP 확인

- 3.3 항목에서 설명한 바와 같이 VIP 는 DB instance boot 시 같이 UP 이 되므로 instance를 boot한다.

- tibero OS user 접속

$ su - tibero

- tbboot normal mode command

$ tbboot 또는 cmrctl start db --name <instance_name>

- tbboot normal mode command

$ tbboot 또는 cmrctl start db --name TEST1

- NIC에 정상적으로 VIP가 추가되었는지 ifconfig를 통해 확인 한다.

$ifconfig -a

```
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.141  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7656:3cff:fe2a:729b  prefixlen 64  scopeid 0x20<link>
        ether 74:56:3c:2a:72:9b  txqueuelen 1000  (Ethernet)
        RX packets 150256326  bytes 37418404734 (34.8 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 103525147  bytes 18438866518 (17.1 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp4s0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.143  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 74:56:3c:2a:72:9b  txqueuelen 1000  (Ethernet)
```

- 결과 확인

$ cmrctl show

```
Resource List of Node CM0
================================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter0        UP (private) 10.10.10.1/11019
     COMMON  network       test_pub0         UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter0, pub: test_pub0
   test_cls     file      test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls     file      test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls  service            TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls       db           TEST1 UP(NRML) TEST, /home/tibero/tibero7, failed retry cnt: 0
   test_cls      vip       test_vip0        UP TEST, 192.168.0.143/255.255.255.0/192.168.0.255 (1)
                                       failed retry cnt: 0
================================================================================
```

7.3.4 Node 2번 – VIP Resource 등록

- vip add command

$ cmrctl add vip --name <vip_name> --node <cm_sid> --svcname <database_name> --ipaddr <virtual_IP >

- vip add ex

$ cmrctl add vip --name test_vip1 --node CM1 --svcname TEST --ipaddr 192.168.0.144

- 결과 확인

$ cmrctl show

```
Resource List of Node CM1
=============================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON  network      test_pub1         UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter1, pub: test_pub1
   test_cls     file     test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls     file     test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls  service           TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls       db          TEST2      DOWN TEST, /home/tibero/tibero7, failed retry cnt: 0
   test_cls      vip       test_vip0     UP(R) TEST, 192.168.0.143/Not Determined/Not Determined (1)
                                         failed retry cnt: 0
   test_cls      vip       test_vip1      DOWN TEST, 192.168.0.144/Not Determined/Not Determined (2)
                                         failed retry cnt: 0
=============================================================================
```

7.3.5 Node 1번 – VIP 확인

- 3.3 항목에서 설명한 바와 같이 VIP 는 DB instance boot 시 같이 UP 이 되므로 instance를 boot한다.

- tibero OS user 접속

$ su - tibero

- tbboot normal mode command

$ tbboot 또는 cmrctl start db --name <instance_name>

- tbboot normal mode command

$ tbboot 또는 cmrctl start db --name TEST2

- NIC에 정상적으로 VIP가 추가되었는지 ifconfig를 통해 확인 한다.

$ifconfig -a

```
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.142  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7656:3cff:fe94:de86  prefixlen 64  scopeid 0x20<link>
        ether 74:56:3c:94:de:86  txqueuelen 1000  (Ethernet)
        RX packets 176915903  bytes 43185676897 (40.2 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 78943181  bytes 11423168274 (10.6 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp4s0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.144  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 74:56:3c:94:de:86  txqueuelen 1000  (Ethernet)
```

- 결과 확인

```
$ cmrctl show
Resource List of Node CM1
================================================================================
  CLUSTER      TYPE         NAME        STATUS            DETAIL
----------- -------- -------------- -------- ------------------------
     COMMON  network     test_inter1        UP (private) 10.10.10.2/11019
     COMMON  network       test_pub1         UP (public) enp4s0
     COMMON  cluster        test_cls        UP inc: test_inter1, pub: test_pub1
   test_cls     file     test_cls:0        UP /tbdata/DB_TEST/CM/CM0/CM0File.cm
   test_cls     file     test_cls:1        UP /tbdata/DB_TEST/CM/CM1/CM1File.cm
   test_cls  service           TEST        UP Database, Active Cluster (auto-restart: OFF)
   test_cls       db          TEST2 UP(NRML) TEST, /home/tibero/tibero7, failed retry cnt: 0
   test_cls      vip       test_vip0     UP(R) TEST, 192.168.0.143/Not Determined/Not Determined (1)
                                        failed retry cnt: 0
   test_cls      vip       test_vip1        UP TEST, 192.168.0.144/255.255.255.0/192.168.0.255 (2)
                                        failed retry cnt: 0
================================================================================
```