Resumo da matéria teórica

1. O que é o DRBD?

DRBD - **D**istributed **R**eplicated **B**lock **D**evice – é um sistema de replicação de dados entre vários servidores. Consiste na replicação de volumes lógicos - /dev/drbdX.

- **Primary/ Secondary:** quando são feitas escritas nos volumes lógicos do nodo **primário**, as alterações são transferidas para o nodo **secundário**. O filesystem encontra-se apenas montado num dos nodos, ou seja, os dados apenas podem ser acedidos a partir de um nodo.

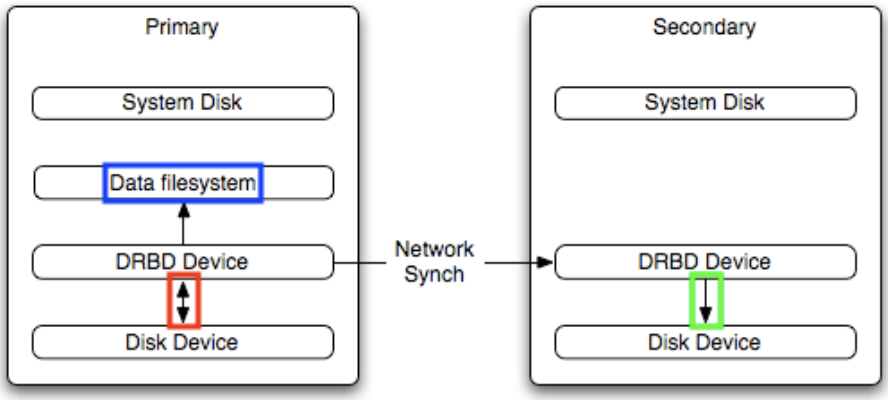

Figura 1 - arquitetura primary – secondary.

- **Dual-Primary**: ambos os nodos escrevem no mesmo volume lógico, por isso o filesystem encontra-se montado em ambos os nodos simultaneamente.

Nota: ao implementar uma arquitetura **dual-primary** é necessário **controlar possíveis escritas concorrentes** por isso já existe software que implementa sistemas de cluster com concorrência controlada.

2. O que é o protocolo iSCSI?

iSCSI - **I**nternet **S**mall **C**omputer **S**ystems **I**nterface – é um protocolo IP para **conectar entre si instalações de armazenamento de dados**.

Permite o controlo de dispositivos de armazenamento através de comandos SCSI - **S**mall **C**omputer **S**ystems **I**nterface – pela camada TCP/IP (pela rede).

Nota: SCSI – é um conjunto de padrões (comandos, protocolos e interfaces elétricas, óticas e lógicas) usados para conectar e transferir dados entre dispositivos periféricos (usb, vga, hdd, etc) e o computador.

Desta forma quem utilizar o protocolo iSCSI conecta-se a um dispositivo remoto (na rede) da mesma maneira que a um dispositivo periférico local (ex: pen usb) tendo **total abstração da conexão remota** (análogo a um DAO).

A arquitetura deste protocolo consiste em **iSCSI clients** e **iSCSI targets**:

- **iSCSI client** ou **initiator** estabelece a ligação com os respetivos targets, emulando que os dispositivos de armazenamento (targets) se encontram conectados localmente na máquina.

Nota: existem 2 tipos de initiators, podem ser implementados em **software** ou **hardware**, o mais comum é ser implementado em software tal como fizemos no guião 4.

- **iSCSI target** é um recurso de armazenamento localizado num servidor iSCSI.

Nota: 1 único servidor iSCSI pode implementar N targets e por isso servir vários iSCSI clients ou até, o mesmo, para diferentes propósitos. (ex: pode servir para conter várias bases de dados para diferentes aplicações)

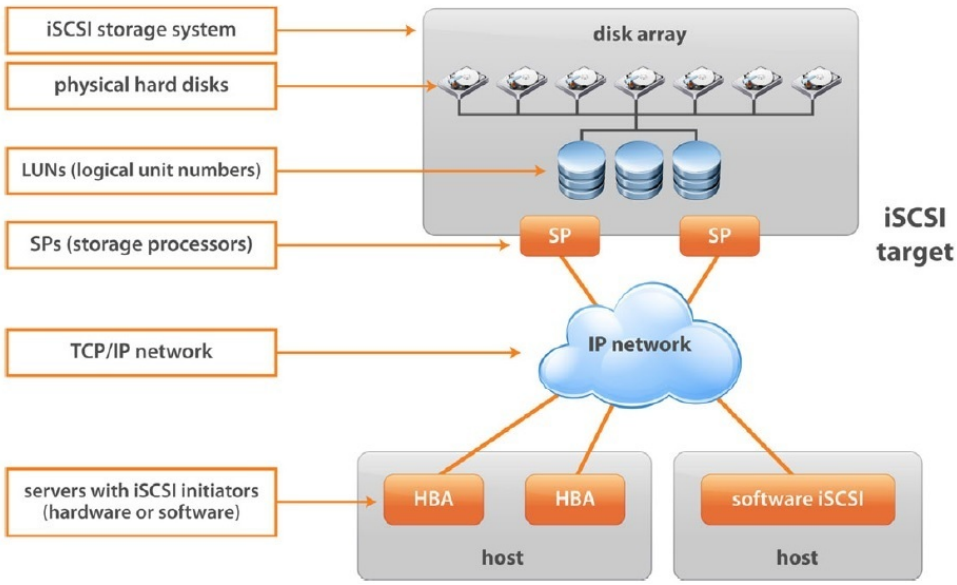

Figura 2 - arquitetura iSCSI.

Guião 4:

1. instalar o repositório elrepo em ambas as máquinas:
rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm
yum update
yum install drbd90-utils kmod-drbd90 drbd90-utils-sysvinit

2. atribuir nome (drbd1 e drbd2) com o nmtui.

3. definir um IP estático nas placas de rede usando o nmtui:
10.0.0.1/24 na drbd1
10.0.0.2/24 na drbd2

4. acrescentar as seguintes linhas ao ficheiro /etc/hosts:
10.0.0.1 drbd1
10.0.0.2 drbd2

5. reiniciar as placa de rede:
systemctl restart network
systemctl restart NetworkManager

6. varificar se as máquinas têm conectividade entre si:
ping 10.0.0.2 na drbd1
ping 10.0.0.1 na drbd2
caso não tenham, fazer reboot ou verfificar se os IPs estão bem configurados.

7. editar o ficheiro de configuração /etc/drbd.d/global_common.conf

```
global {
   usage-count no;
}

common {
   options {
      auto-promote yes;
   }
   net {
      protocol C;
   }
}
```

8. criar o ficheiro de configuração /etc/drbd.d/d1.res para configurar o volume lógico /dev/drbd1.

```
resource d1 {
        net {
                protocol C;
                allow-two-primaries yes;
                after-sb-0pri discard-least-changes;
                after-sb-1pri discard-secondary;
                after-sb-2pri call-pri-lost-after-sb;
        }
        on drbd1 {
                device    /dev/drbd1;
                disk      /dev/sdb;
                address   10.0.0.1:7789;
                meta-disk internal;
        }
        on drbd2 {
                device    /dev/drbd1;
                disk      /dev/sdb;
                address   10.0.0.2:7789;
                meta-disk internal;
        }
}
```

NOTA: guardar cópia das VMs aqui porque os próximos passos são suscetíveis de dar cagada

9. criar o volume lógico /dev/drbd1 em ambas as máquinas:
drbdadm create-md d1

10. inicialzar os volumes em ambas as máquinas:
drbdadm up d1

11. na máquina drbd1:
drbdadm primary **--force** d1
drbdadm status

NOTA: esperar que sincronização acabe

12. na drbd1 criar um filesystem para o volume lógico /dev/drbd1 e montá-lo:
mkfs.xfs /dev/drbd1
mkdir /mnt/lv1
mount /dev/drbd1 /mnt/lv1

13. na máquina drbd2:
drbdadm primary d1
drbdadm status

NOTA: esperar que sincronização acabe

14. na drbd2 criar um filesystem para o volume lógico /dev/drbd1 e montá-lo:
mkfs.xfs /dev/drbd1
mkdir /mnt/lv1
mount /dev/drbd1 /mnt/lv1

NOTA: neste momento temos uma arquitetura dual-primary com o mesmo volume lógico montado em ambas as máquinas ao mesmo tempo

15. Instalar o targetcli.
yum install targetcli

16. Ativar automaticamente o serviço no boot.
systemctl enable target

17. configurar o targetcli **apenas** na drbd1:
targetcli
cd backstores/block
create d1 /dev/drbd1
cd /iscsi
create
cd iqn…/tpg1/luns/
create /backstores/block/d1
cd /iscsi/iqn…/tpg1/
set attribute authentication=0 demo_mode_write_protect=0 generate_node_acls=1
cache_dynamic_acls=1
exit

18. copiar o /etc/target/saveconfig.json da máquina drbd1 para a drbd2
na drbd1 correr:
scp /etc/target/saveconfig.json root@<IP>:/etc/target/saveconfig.json

19. editar o /etc/target/saveconfig.json na drbd2 alterar o wwn do target
(alterar de drbd1 para drbd2)

20. guardar cópias do /etc/target/saveconfig.json:
cp /etc/target/saveconfig.json /etc/target/saveconfig.json.bak

21. instalar nettols em ambas as máquinas:
yum install net-tools.x86_64

CONFIGURAR CLIENT:
… seguir os passos do guião do prof