

# Alta Disponibilidade com Hypervisor Xen 4.0.1 no Debian Squeeze amd64.

## Introdução.

Este artigo irá abordar como deixar um ambiente de virtualização Xen sempre disponível, será utilizado softwares que permite fazer RAID pela rede – DRBD e outro que será responsável por monitorar servidores – Heartbeat.

## Motivação.

Na empresa onde trabalho não possuímos storage (por enquanto, e os investimentos neste ano já foram utilizados), então comecei a pesquisar soluções de alta disponibilidade para virtualização em ambiente que não possui storage, porém não encontrei nenhuma, ambas soluções até pagas, todas utilizam o storage para manter as VM's. Então foi ai que optei pelo XenSource pois com ele é possível fazer a integração de outras ferramentas de alta disponibilidade e ter o ambiente rodando totalmente personalizado atendendo as necessidades da empresa.

## Histórico.

Ao utilizar virtualização conseguimos dimensionar melhor os recursos dos servidores, gerando economia de investimento de máquinas e energia, porém tornamos a deixar o ambiente vulnerável a falhas de hardware, pois se uma máquina física der problemas as máquinas virtualizadas (domUs) que estão rodando no servidor irão ficar indisponíveis. É aí que entra a alta disponibilidade.

Nosso ambiente serão utilizadas 2 máquinas com tamanho de discos iguais, o DRBD exige pelo menos que a partição a ser replicada seja de tamanhos iguais.

Não iremos abordar a instalação do Sistema Operacional Debian Lenny amd64, apenas iremos informar que em ambos os servidores iremos ter uma partição no tamanho de 1 TB que não terá ponto de montagem, esta partição irá conter os dados a serem replicados pelo DRBD.

## 1 - Instalando o Xen

Iremos utilizar a versão do Xen empacotada pela própria distribuição, pois a mesma não é antiga e também é considerada estável pela comunidade do Debian.

```
# apt-get install xen-hypervisor-4.0-amd64 xen-utils-4.0 xen-linux-system-2.6.32-5-xen-amd64 xen-qemu-dm-4.0
```

### 1.2 - Configurando Memória disponível para dom0

Edite o arquivo `/etc/default/grub` deixando um valor reservador de memória RAM para a dom0. Este valor costuma ser baixo em torno de 1024 MB, pois em grande parte das vezes a dom0 só irá servir o Xen, mas neste caso iremos deixar 3048 MB por causa do Drbd e o Heartbeat. Deixe o `grub` desta forma abaixo:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_XEN_DEFAULT="dom0_mem=3048M"
GRUB_CMDLINE_LINUX=""
```

### 1.3 - Adicionando as entradas do kernel no Xen no GRUB

```
# update-grub
```

### 1.4 - Ativando o serviço necessário do Xen durante o Boot

```
update-rc.d xend defaults 20 21
```

Agora é só reiniciar a máquina, logar no terminal e passarmos para a configuração da bridge.

### 1.5 – Removendo parametro de status do Xen para domU.

Devemos editar o arquivo `/etc/default/xend` deixando os parametros abaixo com estes valores. Isto é para evitar problemas do Xen salvar o status das domU em tentar jogar para a outra máquina física.

```
XENDOMAINS_MIGRATE=""
```

```
XENDOMAINS_SAVE=
```

```
XENDOMAINS_RESTORE=false
```

### 1.6 - Configurando Interfaces de Rede.

Nesta situação, as máquinas que estamos utilizando possuem mais de 3 interfaces

de rede, então iremos utilizar, neste exemplo, todas as 3 interfaces de rede. A primeira interface será utilizada para termos acesso à máquina via rede. A segunda será utilizada pelo DRBD para sincronizar os dados com outro servidor via cabo crossover e também pelo Heartbeat para verificar o status do servidor slave (escravo). A terceira interface será a bridge para as VM's (domU), então não iremos definir nem uma configuração para ela neste arquivo, pois se definirmos a interface eth2 sem IP ela sempre dará erro durante o boot.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.50
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.5
    dns-nameservers 192.168.1.33
```

```
auto eth1
iface eth1 inet static
    address 10.1.1.50
    netmask 255.255.255.0
    network 10.1.1.0
    broadcast 10.1.1.255
```

Com as interfaces configuradas, reinicie a máquina e execute o comando ifconfig para ver se as interfaces de rede estarão configuradas.

```
# ifconfig
```

```
eth0    Link encap:Ethernet  Endereço de HW a4:ba:db:28:74:98
        inet end.: 192.168.1.50  Bcast:192.168.1.255  Masc:255.255.255.0
        endereço inet6: fe80::a6ba:dbff:fe28:7498/64  Escopo:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
        RX packets:212879 errors:0 dropped:0 overruns:0 frame:0

        TX packets:1532 errors:0 dropped:0 overruns:0 carrier:0
        colisões:0 txqueuelen:1000
```

RX bytes:20560507 (19.6 MiB) TX bytes:287063 (280.3 KiB)  
IRQ:17 Memória:d2000000-d2012100

eth1 Link encap:Ethernet Endereço de HW a4:ba:db:28:74:9a  
inet end.: 10.1.1.50 Bcast:10.1.1.255 Masc:255.255.255.0  
endereço inet6: fe80::a6ba:dbff:fe28:749a/64 Escopo:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Métrica:1  
RX packets:53388 errors:0 dropped:0 overruns:0 frame:0  
TX packets:55435 errors:0 dropped:0 overruns:0 carrier:0  
colisões:0 txqueuelen:1000  
RX bytes:37267833 (35.5 MiB) TX bytes:41958064 (40.0 MiB)  
IRQ:18 Memória:d4000000-d4012100

### 1.7 - Configuração do Bridge.

Primeiro passo, depois de efetuar a configuração das interfaces de rede, teremos que editar o arquivo de configuração do Xen /etc/xen/xend-config.sxp. Localize a linha abaixo, e insira para ficar da forma abaixo.

```
(vif-script vif-bridge)
(network-script network-xen-multi-bridge)
```

### 1.8 – Inicializando Interfaces para a Bridge

Para que as domUs consigam utilizar as bridges da dom0, o primeiro é fazer com que o xen utilize o script chamado network-xen-multi-bridge que iremos criar dentro do diretório /etc/xen/scripts com o conteúdo abaixo.

```
#!/bin/sh

#network-xen-multi-bridge

#Exit if anything goes wrong

set -e

#First arg is operation.

OP=$1

shift

script=/etc/xen/scripts/network-bridge

case ${OP} in

start)
```

```

$script start vifnum=2 netdev=eth2 bridge=xenbr2
;;

stop)

$script stop vifnum=2 netdev=eth2 bridge=xenbr2

status)

$script status vifnum=2 netdev=eth2 bridge=xenbr2

;;

*)

echo 'Unknown command: ' ${OP}

echo 'Valid commands are: start, stop, status'

exit 1

esac

```

## 2 - Configurando Servidor Slave Xen.

Instale o Debian Lenny deixando uma partição física sem ponto de montagem no tamanho de 1 TB, esta será utilizada pelo DRBD para armazenar os volumes LVM e replicar as informações. Em seguida, instalar o Xen 4.0.1 e configurar conforme foi descrito nos passo 1 até 1.8, porém iremos só alterar IP da interface de rede eth0 para 192.168.1.58, da eth1 10.1.1.51 e deixar a eth2 para ser utilizada como bridge pelas domUs. A configuração das redes ficarão desta forma abaixo: (saída do comando ifconfig)

```

eth0  Link encap:Ethernet  Endereço de HW 00:08:54:18:9b:be
      inet end.: 192.168.1.58  Bcast:192.168.1.255  Masc:255.255.255.0
      endereço inet6: fe80::208:54ff:fe18:9bbe/64  Escopo:Link
      UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
      RX packets:3041424 errors:0 dropped:0 overruns:0 frame:0
      TX packets:10025 errors:0 dropped:0 overruns:0 carrier:0
      colisões:0 txqueuelen:1000
      RX bytes:318162339 (303.4 MiB)  TX bytes:1635587 (1.5 MiB)
      IRQ:20  Endereço de E/S:0xec00

```

```

eth1  Link encap:Ethernet  Endereço de HW 00:0a:5e:54:1b:ed
      inet end.: 10.1.1.51  Bcast:10.1.1.255  Masc:255.255.255.0
      endereço inet6: fe80::20a:5eff:fe54:1bed/64  Escopo:Link
      UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1

```

```
RX packets:975805 errors:0 dropped:0 overruns:0 frame:0
TX packets:724838 errors:0 dropped:0 overruns:0 carrier:0
colisões:0 txqueuelen:1000
RX bytes:1014632449 (967.6 MiB) TX bytes:289400244 (275.9 MiB)
IRQ:18
```

```
lo    Link encap:Loopback Local
      inet end.: 127.0.0.1  Masc:255.0.0.0
      endereço inet6: ::1/128 Escopo:Máquina
      UP LOOPBACKRUNNING MTU:16436 Métrica:1
      RX packets:125 errors:0 dropped:0 overruns:0 frame:0
      TX packets:125 errors:0 dropped:0 overruns:0 carrier:0
      colisões:0 txqueuelen:0
      RX bytes:10904 (10.6 KiB) TX bytes:10904 (10.6 KiB)
```

### 3 – Instalando o DRBD 8

O DRBD é um software que permite fazer um RAID pela rede. Iremos utilizar a versão que esta nos repositórios do Debian Squeeze, Faça este procedimento abaixo nas 2 máquinas:

```
# apt-get install drbd8-utils
```

#### 3.1 - Configurando o DRBD

Primeiro passo é verificar se o DRBD foi compilado com sucesso. Para isto, digite “modprobe drbd” nos 2 servidores. Caso não retorne nem um erro, o drbd foi inicializado. Em seguida, editar o arquivo /etc/hosts das 2 máquinas. Neste exemplo, um irá chamar pitanga e a outra será teste2, desta forma abaixo:

Servidor Pitanga

```
/etc/hosts
127.0.0.1    localhost
192.168.1.50 pitanga.agrovale.com.br pitanga
10.1.1.51    teste2.agrovale.com.br teste2
```

Servidor Teste2

```
/etc/hosts
127.0.0.1    localhost
192.168.1.58 teste2.agrovale.com.br teste2
10.1.1.50    pitanga.agrovale.com.br pitanga
```

Teste a resolução de nomes. No Servidor Pitanga digite “ping teste2” e no Servidor teste2 digite “ping pitanga”. Ambos devem responder ao ping, conforme os IP's que foram configurados.

Iremos configurar o /etc/drbd.conf. Este deverá ser idêntico nas 2 máquinas:

```
global {
usage-count yes;
}

# Velocidade de transferência (utilize em torno de 40% a 60% da sua banda total)
common {
    syncer { rate 100M; }
}

# Nome do resource em questão (será utilizado como referência nos comandos
posteriores, iremos usar o nome xen)
resource xen {

protocol C; # Método de replicação. Neste caso, replicação síncrona.
handlers {

}

startup {

# Tempo que o DRBD irá tentar conectar com o segundo host quando o servidor é
inicializado.
wfc-timeout 15;
degr-wfc-timeout 20; # 2 minutes.
# Define que o host será primário quando o servidor for inicializado.
become-primary-on both;

}

disk {
    on-io-error detach;
}

net {
sndbuf-size 512k;
timeout 60; # 6 seconds (unit = 0.1 seconds)
connect-int 10; # 10 seconds (unit = 1 second)
ping-int 10; # 10 seconds (unit = 1 second)
ping-timeout 5; # 500 ms (unit = 0.1 seconds)
max-buffers 2048;

# Informa que será utilizado o DRBD primário nas duas máquinas (válido somente para
últimas versões do DRBD)
allow-two-primaries;

after-sb-0pri discard-zero-changes;
```

```
after-sb-0pri discard-least-changes;
after-sb-1pri discard-secondary;
}
```

```
syncer {
rate 100M;
al-extents 257;
}
```

```
on pitanga {
device /dev/drbd0;
disk /dev/sda7;
address 10.1.1.50:7788;
meta-disk internal;
}
```

```
on teste2 {
device /dev/drbd0;
disk /dev/sda7;
address 10.1.1.51:7788;
meta-disk internal;
```

```
}
```

```
}
```

Meta-disk – disco temporário utilizado pelo drbd para armazenamento.

Lembrando que a partição sda7 não poderá estar montada em nenhuma das máquinas.

Caso esteja, desmonte-a com o comando “umount /dev/sda7” e, caso necessário, edite o arquivo /etc/fstab removendo a entrada /dev/sda7.

### **3.2 - Zerando partições antes de aplicar o DRBD.**

Serv. Pitanga

```
# dd if=/dev/zero of=/dev/sda7 bs=1M count=128
```

Serv. Teste2

```
# dd if=/dev/zero of=/dev/sda7 bs=1M count=128
```

### **3.3 – Criando meta data do resource para DRBD.**

Onde xen é o nome do nosso dispositivo, que no arquivo de configuração do drbd está como resource xen. Execute este comando nos 2 servidores.

Servidor Pitanga

```
# drbdadm create-md xen
```

Servidor Teste2

```
# drbdadm create-md xen
```



### **3.4 – Anexando Dispositivo ao Meta Dados criado.**

Servidor Pitanga  
# drbdadm attach xen

Servidor Teste2  
# drbdadm attach xen

### **Definindo e Inicializando sincronização do dispositivo.**

Servidor Pitanga  
# drbdadm syncer xen

Servidor Teste2  
# drbdadm syncer xen

### **Conectando os nodes.**

Servidor Pitanga  
# drbdadm connect xen

Servidor Teste2  
# drbdadm connect xen

### **Força a partição do Servidor Pitanga como Primário,**

Irá definir a sincronização, pois o drbd necessita mesmo que seja 2 primários, a partição de um dos servidores será primaria:

# drbdadm -- --overwrite-data-of-peer primary xen

Reiniciando serviço DRBD.

Servidor Pitanga  
# /etc/init.d/drbd restart

Servidor Teste2  
# /etc/init.d/drbd restart

### **3.5 – Verificando a Sincronização dos discos.**

Em ambos os servidores, digite o comando abaixo, que deverá retornar o resultado próximo ao demonstrado. Aguarde a sincronização terminar. O tempo irá variar de acordo com o tamanho da partição e velocidade de discos e rede. Observe, em negrito, a palavra **Inconsistent**. Isto ocorre pois os servidores não terminaram de sincronizar.

```
# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: bb447522fc9a87d0069b7e14f0234911ebdab0f7 build by phil@fat-tyre, 2010-
01-23 16:40:33
0: cs:SyncSource st:Primary/ Primary ds:UpToDate/Inconsistent C r---
ns:898320 nr:0 dw:0 dr:909728 al:0 bm:54 lo:0 pe:15 ua:357 ap:0
[==>.....] sync'ed: 18.5% (3892/4769)M
finish: 0:00:39 speed: 99,760 (99,760) K/sec
resync: used:2/61 hits:56431 misses:56 starving:0 dirty:0 changed:56
act_log: used:0/257 hits:0 misses:0 starving:0 dirty:0 changed:0
```

#### 4 – Formatando o disco virtual.

Escolha um dos servidores, no caso iremos escolher o Pitanga para formatar o disco virtual. Este procedimento é necessário apenas em uma das máquinas. Estamos utilizando o EXT4 como sistema de arquivos. Fica a critério qual utilizar qualquer um, porem tive alguns problemas com o XFS de corromper o filesystem do DRBD.

```
# mkfs.ext4 /dev/drbd0
```

##### 4.1 – Realizando Testes

Em ambos os servidores, crie um ponto de montagem para realizar teste de sincronização dos dados:

```
# mkdir /mnt/drbd
```

No Servidor pitanga monte a partição virtual.

```
# mount /dev/drbd0 /mnt/drbd
```

Crie um arquivo chamado teste1.

```
# touch sincronismo /mnt/drbd/
```

Confirme de o arquivo sincronismo foi criado

```
# ls /mnt/drbd
```

Desmonte

```
# umount /mnt/drbd
```

Verifique se existe o arquivo chamado sincronismo na partição montada

```
# ls /mnt/drbd
```

Após conferir que a sincronização está funcionando, desmonte a partição no servidor Teste2. Obs. A partição virtual só poderá ser montada para conferência. A mesma deverá permanecer desmontada em ambos os servidores

## 5 – Instalando o Heartbeat.

O Heartbeat é o software responsável em verificar e monitorar os serviços dos servidores que fazem parte do cluster. Este software mantém uma comunicação direta entre os 2 servidores, e, caso algum não responda, o mesmo irá iniciar os serviços que foram configurados para serem inicializados.

Servidor Pitanga

```
# apt-get install heartbeat
```

Servidor Teste2

```
# apt-get install heartbeat
```

### 5.1 – Configurando Heartbeat

Caso não seja criado os arquivos de configuração ha.cf, authkeys e haresources dentro de /etc/ha.d, será necessário copiar e descompactar estes arquivos que estão localizados dentro de /usr/share/doc/heartbeat para dentro de /etc/ha.d

Este arquivo será idêntico nos dois servidores:

```
/etc/ha.d/ha.cf
```

```
# Arquivos de debug
```

```
debugfile /var/log/ha-debug
```

```
# Arquivos de Log
```

```
logfile /var/log/ha-log
```

```
logfacility local0
```

```
# Frequência, em segundos, da verificação das máquinas
```

```
keepalive 2
```

```
#Tempo mínimo para declarar a outra máquina como morta (não utilizar muito baixo)
```

```
deadtime 07
```

```
warnetime 10
```

```
# Interface dedicada para o heartbeat (e a mesma utilizada pelo DRBD)
```

```
bcast eth1 # Linux
```

```
# O Servidor Pitanga retornado de uma queda, ira ter os serviços assumidos
```

```
auto_failback on
```

```
# informe os nomes dos computadores que formam a replicação (deve ser igual a saída do comando uname -n)
```

```
node pitanga
node teste2
```

## 5.2 – Informando os serviços a serem inicializados pelo Heartbeat

O arquivo responsável por iniciar os serviços é o `/etc/ha.d/haresources`. Iremos indicar o nome do script que irá subir o Xen, e também o arquivo de configuração da VM.

Este arquivo será idêntico nos dois servidores.

```
pitanga xendomains::vm1.cfg
```

```
pitanga xendomains::vm2.cfg
```

## 5.3 – Definindo configuração da chave autenticação.

Edite o arquivo `/etc/ha.d/authkeys` e defina a permissão de somente leitura e escrita para o dono do arquivo:

```
# chmod 600
```

Estes procedimentos serão idênticos nos dois servidores.

```
auth 1
```

```
1 crc
```

## 5.4 – Criando Script xendomains para o Heartbeat.

Salve o conteúdo abaixo em um arquivo chamado `xendomains` dentro de `/etc/ha.d/resources.d` de ambos os servidores. O Arquivo `vm1.cfg` e `vm2.cfg`, que são os arquivos responsáveis pela configuração das domUs, estarão em um diretório dentro de `/home` chamado `VM`. Os scripts de inicialização `xendomains` irão utilizar este diretório para carregar as configurações das Vms:

```
#!/bin/bash
XM="/usr/sbin/xm"
CONFPATH="/home/VM/"

RES="$1"
CMD="$2"
NOME_BASE=`basename $RES .cfg`
isrunning=false
case "$CMD" in
    start)
        $XM create -f $CONFPATH$RES
        ;;
    stop)
        exec $XM shutdown --halt --wait $NOME_BASE
        ;;
    status)
```

```

$XM list | awk '{print $1}' | grep $RES > /dev/null
if [ $? -eq 0 ]
then
    echo running
else
    echo stopped
fi
;;
*)
    echo "Usage: xendomain [filename] {start|stop|status}"
    exit 1
;;
esac

exit 0

```

## 6 – Criando Volumes LVM

Antes de iniciarmos o processo de criação dos volumes LVM, teremos que informar ao LVM que será criado em cima de um dispositivo virtual que será o do DRBD. Toda a criação dos PVs, VGs terá como base o disco virtual. É bom lembrar que não iremos utilizar a partição física /dev/sda7 para mais nada. Tudo será criado pelo LVM no dispositivo /dev/drbd0 que, por sua vez, esta sobre o dispositivo /dev/sda7.

Localize a linha abaixo lvm.conf e deixe da forma abaixo nos dois servidores:

```

/etc/lvm/lvm.conf
antes
#filter = [ "r|/dev/sda7|", "r|/dev/disk|", "r|/dev/block|", "a/*|" ]

depois
filter = [ "a|drbd.*|", "r|.*)" ]

write_cache_state = 0

```

### 6.1 - Criando o PV

Estes procedimentos abaixo devem ser realizados somente em um dos servidores. Escolhemos o pitanga pois ele está com a partição primaria do DRBD. Obs. Tudo relacionado ao LVM terá que ser realizado em um dos servidores, no caso iremos usar o pitanga para isto. Recomendamos que seja apenas em um, pois, depois de pronto, será necessário enviar os arquivos do LVM para o servidor slave, tendo que sincronizar volumes criados no master e no slave e descobrir qual tem que ser sincronizado ou não.

```
# pvcreate /dev/drbd0
```

#### Conferindo se o PV foi criado

```
# pvscan
```

#### Criando Grupo de Volume

```
# vgcreate VGxen /dev/drbd0
```

### **Verificando se o grupo foi criado**

```
# vgs
```

### **Criando volume**

No exemplo abaixo iremos criar 2 volumes. O parâmetro -L informa o tamanho que este volume irá conter, -n define o nome do volume e, por fim, VGxen será o grupo que foi criado pelo passo anterior:

```
# lvcreate -L 1G -n debian_vm1.disk1 VGxen  
  (Partição Swap)  
# lvcreate -L 100G -n debian_vm1.disk2 VGxen  
  (Partição dados VM1)
```

```
# lvcreate -L 1G -n debian_vm2.disk1 VGxen  
  (Partição Swap)  
# lvcreate -L 100G -n debian_vm2.disk2 VGxen  
  (Partição dados VM2)
```

### **Formatando Volumes**

```
# mkswap /dev/VGxen/debian_vm1.disk1 (Formatando Partição Swap para VM1)  
# mkswap /dev/VGxen/debian_vm2.disk1 (Formatando Partição Swap para VM2)
```

Escolhemos EXT4 como sistema de arquivos para as domUs.

```
# mkfs.ext4 /dev/VGxen/debian_vm1.disk2 (Formatando Partição Dados para VM1)  
# mkfs.ext4 /dev/VGxen/debian_vm2.disk2 (Formatando Partição Dados para VM1)
```

## **7 - Instalando Debootstrap**

Instale o pacote debootstrap nos 2 servidores, com isto irá baixar um Debian básico dos repositórios do Debian:

```
# apt-get install debootstrap
```

### **7.1 – Instalando sistema Debian Básico**

No Servidor Pitanga, crie um ponto de montagem para que a partição de dados da VM1 e VM2 seja disponibilizada para instalação do Debian básico:

```
# mkdir /mnt/vm1  
# mkdir /mnt/vm2
```

```
# mount dev/VGxen/debian_vm1.disk2 /mnt/vm1  
# mount dev/VGxen/debian_vm2.disk2 /mnt/vm2
```

```
# debootstrap --arch amd64 lenny /mnt/vm1 http://ftp.br.debian.org/debian  
# debootstrap --arch amd64 lenny /mnt/vm2 http://ftp.br.debian.org/debian
```

Obs.: Caso queira, pode trocar o "--arch amd64" por "--arch i386" por exemplo.

Este processo deverá demorar um pouco dado que todos os pacotes serão baixados da Internet.

## 7.2 – Configuração pós instalação.

Estes arquivos abaixo a serem editados estão localizados dentro do ponto de montagem, tanto dentro de /mnt/vm1 como /mnt/vm2

### /etc/fstab

Deixe o /mnt/vm1/etc/fstab e /mnt/vm2/etc/fstab desta forma abaixo, lembrando que utilizamos XFS como sistema de arquivos:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options>    <dump> <pass>
proc            /proc          proc defaults    0      0
/dev/xvda1      none           swap  sw           0      1
/dev/xvda2      /              ext4  defaults     0      2
```

### /etc/inittab

Insira a linha em negrito no /mnt/vm1/etc/inittab e /mnt/vm2/etc/inittab abaixo destas linhas.

```
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
co:2345:respawn:/sbin/getty 38400 xvc0
```

### /etc/securetty

Confira se /mnt/vm1/etc/securetty e /mnt/vm2/etc/securetty existe uma linha como esta abaixo. É comum que já esta inserida, caso não esteja, insira:

```
# Oldstyle Xen console
```

```
xvc0
```

### /etc/hosts

Nos arquivos /mnt/vm1/etc/hosts e /mnt/vm2/etc/hosts inclua o nome e domínio das máquinas e IP's:

```
/mnt/vm1/etc/hosts
127.0.0.1          localhost
192.168.1.253      vm1.agrovale.com.br  vm1
```

```
/mnt/vm1/etc/hosts
```

```
127.0.0.1      localhost
192.168.1.254  vm2.agrovale.com.br  vm2
```

### **/etc/hostname**

Defina o nome das VM's:

```
/mnt/vm1/etc/hostname
vm1
```

```
/mnt/vm2/etc/hostname
vm2
```

### **/etc/network/interfaces**

Informe os IP's das máquinas:

```
/mnt/vm1/etc/network/interfaces
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
```

```
auto eth0
    iface eth0 inet static
    address 192.168.1.253
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.5
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.1.33
```

```
/mnt/vm2/etc/network/interfaces
```

```
# The primary network interface
```

```
auto eth0
    iface eth0 inet static
    address 192.168.1.254
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.5
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.1.33
```

### **Copiando Módulos paras vms**



Agora, com todos os arquivos configurados, copie os módulos do kernel para o local apropriado no disco da máquina virtual:

```
# cp -a /lib/modules/2.6.32-5-xen-amd64 /mnt/vm1/lib/modules
# cp -a /lib/modules/2.6.32-5-xen-amd64 /mnt/vm2/lib/modules
```

Terminada a configuração inicial para os servidores virtuais, neste ponto teremos que criar um diretório em cada servidor, dentro de /home chamado VM. Este irá armazenar os arquivos de configuração vm1.cfg e o vm2.cfg.

Servidor Pitanga

```
# mkdir /home/VM
# vim /home/VM/vm1.cfg
# Configuracao DomU para utilizar kernel do lenny
import commands
krn_vers = commands.getoutput('uname -r')
builder = 'linux'
name = 'vm1'
disk = [ 'phy:/dev/VGxen/debian_vm1.disk1,xvda1,w',
'phy:/dev/VGxen/debian_vm1.disk2,xvda2,w' ]
vif = [ 'mac=00:17:3a:0a:cf:aa, bridge=xenbr2' ]
vcpus = 2
memory = 1024
kernel = '/boot/vmlinuz-' + krn_vers
ramdisk = '/boot/initrd.img-' + krn_vers
root = '/dev/xvda2 ro'
on_poweroff = 'destroy'
on_reboot = 'restart'
on_crash = 'restart'
extra = 'xencons=xvc console=xvc0 video=tty'
```

```
# vim /home/VM/vm1.cfg
# Configuracao DomU para utilizar kernel do lenny
import commands
krn_vers = commands.getoutput('uname -r')
builder = 'linux'
name = 'vm2'
disk = [ 'phy:/dev/VGxen/debian_vm2.disk1,xvda1,w',
'phy:/dev/VGxen/debian_vm2.disk2,xvda2,w' ]
vif = [ 'mac=00:17:3b:0b:cf:bb, bridge=xenbr2' ]
vcpus = 2
memory = 1024
kernel = '/boot/vmlinuz-' + krn_vers
ramdisk = '/boot/initrd.img-' + krn_vers
root = '/dev/xvda2 ro'
on_poweroff = 'destroy'
on_reboot = 'restart'
```

```
on_crash = 'restart'
extra = 'xencons=xvc console=xvc0 video=tty'
```

Realize o procedimento acima no servidor Teste2.

### 7.3 - Testando os servidores virtuais Xen (domU)

No Servidor Pitanga execute estes comando abaixo. para testar se as DomU irão ser inicializadas:

```
# xm create /home/VM/vm1.cfg
# xm create /home/VM/vm2.cfg
```

Listando maquinas virtuais

```
# xm list
```

Acessando Console da vm1 e vm2

```
# xm console vm1
# xm console vm2
```

Saindo do Console das domU, aperte  
Ctrl + Shift + }.

Desligar as máquinas virtuais domU's:

```
# xm shutdown -H vm1
# xm shutdown -H vm2
```

Detalhe importante: para que seja possível subir os servidores virtuais, o serviço xend tem que estar inicializado durante o boot do dom0 e o xendomains não estar para iniciar no boot.

## 8 – Adequando Script de Inicialização do DRBD

Devemos fazer uma alteração do script de inicialização do DRBD nos 2 servidores. Esta adequação é necessária pois sempre devemos ter os volumes LVM ativados quando o DRBD for iniciado ou desativado. Não encontramos outra maneira de como fazer o DRBD ativar e desativar o volume LVM a não ser desta forma. No script só terá a inserção das linha abaixo em negrito nos determinados lugares:

```
/etc/init.d/drbd
case "$1" in
```

```
start)
```

```
# Just in case drbdadm want to display any errors in the configuration
# file, or we need to ask the user about registering this installation
# at http://usage.drbd.org, we call drbdadm here without any IO
```

```

# redirection.
$DRBDADM sh-nop
log_daemon_msg "Starting DRBD resources"
assure_module_is_loaded
adjust_with_progress

# make sure udev has time to create the device files
for RESOURCE in ` $DRBDADM sh-resources `; do
    for DEVICE in ` $DRBDADM sh-dev $RESOURCE `; do
        UDEV_TIMEOUT_LOCAL=$UDEV_TIMEOUT
        while [ ! -e $DEVICE ] && [ $UDEV_TIMEOUT_LOCAL -gt 0 ] ; do
            sleep 1
            UDEV_TIMEOUT_LOCAL=$(( $UDEV_TIMEOUT_LOCAL-1 ))
        done
    done
done

[ -d /var/lock/subsys ] && touch /var/lock/subsys/drbd # for RedHat
$DRBDADM wait-con-int # User interruptible version of wait-connect all
$DRBDADM sh-b-pri all # Become primary if configured
log_end_msg 0
vgchange -a y # Ativa volume LVM, tem que ser no final do start
;;

```

stop)

```

vgchange -a n # Desativa volume LVM, tem que ser no inicio do stop.
$DRBDADM sh-nop
log_daemon_msg "Stopping all DRBD resources"
if [ -e $PROC_DRBD ] ; then
    # bypass drbdadm and drbd config file and everything,
    # to avoid leaving devices around that are not referenced by
    # the current config file, or in case the current config file
    # does not parse for some reason.
    for d in /dev/drbd* ; do
        [ -L "$d" ] && continue
        [ -b "$d" ] || continue
        M=$(umount "$d" 2>&1)
        case $M in
            *" not mounted") ;;
            *) echo "$M" >&2 ;;
        esac
        $DRBDSETUP "$d" down
    done
    $RMMOD drbd
fi

```

Obs.: Com esta modificação reinicie o DRBD nas duas máquinas

```
# /etc/init.d/drbd restart
```

Em seguida, inicialize o Heartbeat na máquina Pitanga, logo, inicialize o Heartbeat na máquina Teste2

```
#/etc/init.d/heartbeat start
```

## **9 – Serviços para serem inicializados durante o boot dos Servidores dom0**

Coloque o DRBD e o HEARTBEAT para serem inicializados automaticamente durante o boot em ambos os servidores.

## **10 – Arquivos de LVM sendo enviado para Servidor Slave.**

Teremos que enviar o conteúdo do /etc/lvm/archive do servidor pitanga para o servidor slave teste2, isto se deve ao fato que criamos todos volumes LVM em apenas em um dos servidores. Este procedimento deverá ser realizado sempre que criar um novo volume LVM no servidor pitanga.

```
# rsync -Cravzp --delete -e 'ssh -p PORTA_SSH' /etc/lvm/archive/  
root@IP_SERVIDOR_TESTE2:/etc/lvm/archive
```

Caso não tenha instalado o rsync, instale-o nos 2 servidores.

## **11 – Realizando Testes**

Acesse o Servidor Pitanga e pare o serviço Heartbeat, com isto o servidor Teste2 irá subir a vm1 e vm2

```
# /etc/init.d/heartbeat stop
```

Depois de verificado se os serviços foram migrados para o Servidor Teste2, inicialize o Heartbeat da máquina Pitanga para que os serviços voltem para ela, e, posteriormente, para o Heartbeat da máquina Teste2, constatando que os serviços continuaram rodando no servidor Pitanga.

## **Conclusão**

Esta solução esta em pleno funcionamento sem nem um problema, e o que você poderia aproveitar e utilizar o segundo servidor (TESTE2) para rodar algumas domU. Assim aproveitando os recursos da segunda maquina o TESTE2 para ela não ficar apenas aguardando entrar em produção quando o servidor pitanga cair.

Lembrando que você deve tomar muito cuidado para não ter 2 servidores domU rodando simultaneamente tanto no Servidor Pitanga como no Servidor Teste2, pois se isto ocorrer você terá problema de danificar o filesystem das mesmas. Então e melhor você sempre deixar estas domU que irá rodar no Servidor Teste2 para serem inicializadas

manualmente para não correr o risco o heartbeat subi-las em outro servidor.

Esta solução só foi possível realiza-las com ajuda do material encontrado na Internet, e também ajuda dos colegas citado abaixo.

### **Agradecimentos.**

Aldrey Galdino  
Diogo Oliveira  
Marcos Pitanga  
Marco Sinharelli  
Renato Pacheco  
Rodrigo Braga

### **Referencias:**

<http://wiki.xen-br.org/Xen-3.4.1-install-all>

<http://wiki.xen-br.org/Vm-debian-3.4.x>

<http://www.thepolux.com.br/index.php/linux/111-drbd-heartbeat-samba>

<http://www.vivaolinux.com.br/artigo/Instalacao-configuracao-utilizacao-e-testes-do-DRBD-0.7/?pagina=3>

<http://www.vivaolinux.com.br/artigo/Alta-disponibilidade-com-Debian-Lenny-+-Heartbeat-+-DRBD8-+-OCFS2-+-MONIT-+-LVS?pagina=6>

<http://gcharriere.com/blog/?p=1>

[http://www.linuxtoday.com/high\\_performance/2009022602535OSSVNT](http://www.linuxtoday.com/high_performance/2009022602535OSSVNT)