By Falko Timme
Published: 2009-04-28 18:16

Xen Live Migration Of An LVM-Based Virtual Machine With iSCSI On Debian Lenny

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 04/16/2009

This guide explains how you can do a live migration of an LVM-based virtual machine (domU) from one Xen host to the other. I will use iSCSI to provide shared storage for the virtual machines in this tutorial. Both Xen hosts and the iSCSI target are running on Debian Lenny in this article.

I do not issue any guarantee that this will work for you!

1 Preliminary Note

I'm using the following systems here:

- Xen host 1 : *server.example.com*, IP address: *192.168.0.100*
- Xen host 2 : *server2.example.com*, IP address: *192.168.0.101*
- iSCSI target (shared storage): *iscsi.example.com*, IP address: *192.168.0.102*
- virtual machine: *vm1.example.com*, IP address: *192.168.0.103*

I will use LVM on the shared storage so that I can create/use LVM-based Xen guests.

The two Xen hosts and the iSCSI target should have the following lines in */etc/hosts* (unless you have a DNS server that resolves the hostnames):

```
vi /etc/hosts
```

```
127.0.0.1       localhost.localdomain   localhost
192.168.0.100   server1.example.com     server1
192.168.0.101   server2.example.com     server2
192.168.0.102   iscsi.example.com       iscsi
192.168.0.103   vm1.example.com         vm1
[...]
```

2 Xen Setup

server1/server2:

The two Xen hosts should be set up according to chapter two of this tutorial: **Virtualization With Xen On Debian Lenny (AMD64)**

To allow live migration of virtual machines, we must enable the following settings in */etc/xen/xend-config.sxp*...

```
vi /etc/xen/xend-config.sxp
```

```
[...]
(xend-relocation-server yes)
[...]
(xend-relocation-port 8002)
[...]
(xend-relocation-address '')
[...]
(xend-relocation-hosts-allow '')
[...]
```

... and restart Xen:

```
/etc/init.d/xend restart
```

3 Setting Up The iSCSI Target (Shared Storage)

iscsi.example.com:

Now we set up the target. The target will provide shared storage for *server1* and *server2*, i.e., the virtual Xen machines will be stored on the shared storage.

```
aptitude install iscsitarget iscsitarget-modules-`uname -r`
```

Open */etc/default/iscsitarget*...

```
vi /etc/default/iscsitarget
```

... and set *ISCSITARGET_ENABLE* to *true*:

```
ISCSITARGET_ENABLE=true
```

We can use unused logical volumes, image files, hard drives (e.g. */dev/sdb*), hard drive partitions (e.g. */dev/sdb1*) or RAID devices (e.g. */dev/md0*) for the storage. In this example I will create a logical volume of 20GB named *storage_lun1* in the volume group *vg0*:

```
lvcreate -L20G -n storage_lun1 vg0
```

(If you want to use an image file, you can create it as follows:

```
mkdir /storage
```

```
dd if=/dev/zero of=/storage/lun1.img bs=1024k count=20000
```

This creates the image file */storage/lun1.img* with a size of 20GB.

)

Next we edit */etc/ietd.conf*...

```
vi /etc/ietd.conf
```

... and comment out everything in that file. At the end we add the following stanza:

```
[...]
Target iqn.2001-04.com.example:storage.lun1
        IncomingUser someuser secret
        OutgoingUser
        Lun 0 Path=/dev/vg0/storage_lun1,Type=fileio
        Alias LUN1
        #MaxConnections  6
```

The target name must be a globally unique name, the iSCSI standard defines the "iSCSI Qualified Name" as follows: *iqn.yyyy-mm.<reversed domain name>[:identifier]*; *yyyy-mm* is the date at which the domain is valid; the identifier is freely selectable. The *IncomingUser* line contains a username and a password so that only the initiators (clients) that provide this username and password can log in and use the storage device; if you don't need authentication, don't specify a username and password in the *IncomingUser* line. In the *Lun* line, we must specify the full path to the storage device (e.g. */dev/vg0/storage_lun1*, */storage/lun1.img*, */dev/sdb*, etc.).

Now we tell the target that we want to allow connections to the device *iqn.2001-04.com.example:storage.lun1* from the IP address *192.168.0.100* (*server1.example.com*) and *192.168.0.101* (*server2.example.com*)...


```
vi /etc/initiators.allow
```

```
[...]
iqn.2001-04.com.example:storage.lun1 192.168.0.100, 192.168.0.101
```

... and start the target:

```
/etc/init.d/iscsitarget start
```

4 Making The Shared Storage Available On server1 And server2

server1/server2:

On *server1* and *server2*, we install the initiator:

```
aptitude install open-iscsi
```

Next we open */etc/iscsi/iscsid.conf*...

```
vi /etc/iscsi/iscsid.conf
```

... and set *node.startup* to *automatic*:

```
[...]
node.startup = automatic
[...]
```

Then we restart the initiator:

```
/etc/init.d/open-iscsi restart
```

Now we connect to the target (*iscsi.example.com*) and check what storage devices it has to offer:

```
iscsiadm -m discovery -t st -p 192.168.0.102
```

```
server1:~# iscsiadm -m discovery -t st -p 192.168.0.102
  192.168.0.102:3260,1 iqn.2001-04.com.example:storage.lun1
server1:~#
```

```
iscsiadm -m node
```

```
server1:~# iscsiadm -m node
  192.168.0.102:3260,1 iqn.2001-04.com.example:storage.lun1
server1:~#
```

The settings for the storage device *iqn.2001-04.com.example:storage.lun1* on *192.168.0.102:3260,1* are stored in the file */etc/iscsi/nodes/iqn.2001-04.com.example:storage.lun1/192.168.0.102,3260,1/default*. We need to set the username and password for the target in that file; instead of editing that file manually, we can use the *iscsiadm* command to do this for us:

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --op=update --name node.session.auth.authmethod --value=CHAP

iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --op=update --name node.session.auth.username --value=someuser
```



```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --op=update --name node.session.auth.password
--value=secret
```

Now we can log in by running...

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --login
```

In the output of

```
fdisk -l
```

you should now find a new hard drive; that's our iSCSI storage device (in this example, it's named */dev/sdf* on *server1* and */dev/sdc* on *server2*):

server1 output:

```
server1:~# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00023cd1

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          62      497983+  83  Linux
/dev/sda2              63       60801   487886017+   5  Extended
/dev/sda5              63       60801   487885986   8e  Linux LVM

Disk /dev/dm-0: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Disk identifier: 0x00000000

Disk /dev/dm-0 doesn't contain a valid partition table

Disk /dev/dm-1: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-1 doesn't contain a valid partition table
```

Disk /dev/sdf: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x00000000

Disk /dev/sdf doesn't contain a valid partition table
server1:~#

server2 output:

```
server2:~# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00036268

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          62      497983+  83  Linux
/dev/sda2              63       60801   487886017+   5  Extended
/dev/sda5              63       60801   487885986   8e  Linux LVM
```

```
Disk /dev/dm-0: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-0 doesn't contain a valid partition table

Disk /dev/dm-1: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-1 doesn't contain a valid partition table
```

Disk /dev/sdc: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x00000000

Disk /dev/sdc doesn't contain a valid partition table
server2:~#

To use that device, we must format it (I want to create/use LVM-based virtual machines, therefore I format it as LVM (type `8e`)):

server1:

```
fdisk /dev/sdf
```

```
server1:~# fdisk /dev/sdf
  Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
  Building a new DOS disklabel with disk identifier 0x353f5965.
  Changes will remain in memory only, until you decide to write them.
```

```
  After that, of course, the previous content won't be recoverable.


  The number of cylinders for this disk is set to 20480.
  There is nothing wrong with that, but this is larger than 1024,
  and could in certain setups cause problems with:
  1) software that runs at boot time (e.g., old versions of LILO)
  2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): <-- n
Command action
   e   extended
   p   primary partition (1-4)
<-- p
Partition number (1-4): <-- 1
First cylinder (1-20480, default 1): <-- ENTER
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-20480, default 20480): <-- ENTER
Using default value 20480

Command (m for help): <-- t
Selected partition 1
Hex code (type L to list codes): <-- L

 0  Empty           1e  Hidden W95 FAT1 80  Old Minix       be  Solaris boot
 1  FAT12           24  NEC DOS         81  Minix / old Lin bf  Solaris
 2  XENIX root      39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 5  Extended        41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 6  FAT16           42  SFS             86  NTFS volume set da  Non-FS data
```

```
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .
 8  AIX             4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e1  DOS access
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          e4  SpeedStor
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  EFI GPT
10  OPUS            55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b
12  Compaq diagnost 5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f4  SpeedStor
16  Hidden FAT16    63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fd  Linux raid auto
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fe  LANstep
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid ff  BBT
1c  Hidden W95 FAT3 75  PC/IX
Hex code (type L to list codes): <-- 8e
Changed system type of partition 1 to 8e (Linux LVM)

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

Calling ioctl() to re-read partition table.
Syncing disks.
server1:~#
```

Afterwards, the output of

```
fdisk -l
```

should look as follows:

```
server1:~# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00023cd1

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          62      497983+  83  Linux
/dev/sda2              63       60801   487886017+   5  Extended
/dev/sda5              63       60801   487885986   8e  Linux LVM

Disk /dev/dm-0: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-0 doesn't contain a valid partition table

Disk /dev/dm-1: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-1 doesn't contain a valid partition table

Disk /dev/sdf: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x353f5965

   Device Boot      Start         End      Blocks   Id  System
/dev/sdf1               1       20480    20971504   8e  Linux LVM
```

```
server1:~#
```

Since this is a shared storage, /dev/sdc on server2 should now also contain an LVM partition, /dev/sdc1:

server2:

```
fdisk -l
```

```
server2:~# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00036268

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          62      497983+  83  Linux
/dev/sda2              63       60801   487886017+   5  Extended
/dev/sda5              63       60801   487885986   8e  Linux LVM

Disk /dev/dm-0: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/dm-0 doesn't contain a valid partition table

Disk /dev/dm-1: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000
```

```
Disk /dev/dm-1 doesn't contain a valid partition table

Disk /dev/sdc: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x353f5965

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1       20480    20971504   8e  Linux LVM
server2:~#
```

Now I initialize `/dev/sdf1` on `server1` for LVM usage and create the volume group `vg_xen` on it:

server1:

```
pvcreate /dev/sdf1
```

```
vgcreate vg_xen /dev/sdf1
```

In order to make the new volume group available on `server2`, we must first log out of iSCSI and then back in:

server2:

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --logout

iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.102:3260" --login
```

Then run...

```
vgscan
```



```
server2:~# vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "vg_xen" using metadata type lvm2
  Found volume group "vg0" using metadata type lvm2
server2:~#
```

... and...

```
vgchange -a y
```

```
server2:~# vgchange -a y
  0 logical volume(s) in volume group "vg_xen" now active
  2 logical volume(s) in volume group "vg0" now active
server2:~#
```

... to activate the *vg_xen* volume group on *server2*.

5 Creating Virtual Machines

We will use **xen-tools** to create virtual machines. xen-tools make it very easy to create virtual machines - please read this tutorial to learn more: **http://www.howtoforge.com/xen_tools_xen_shell_argo**.

Now we edit */etc/xen-tools/xen-tools.conf*. This file contains the default values that are used by the *xen-create-image* script unless you specify other values on the command line. I changed the following values and left the rest untouched:

server1/server2:

```
vi /etc/xen-tools/xen-tools.conf
```

```
[...]
```

```
lvm = vg_xen
[...]
dist   = lenny     # Default distribution to install.
[...]
gateway   = 192.168.0.1
netmask   = 255.255.255.0
broadcast = 192.168.0.255
[...]
passwd = 1
[...]
kernel      = /boot/vmlinuz-`uname -r`
initrd      = /boot/initrd.img-`uname -r`
[...]
mirror = http://ftp.de.debian.org/debian/
[...]
serial_device = hvc0
[...]
disk_device = xvda
[...]
```

Make sure that you uncomment the *lvm* line and fill in the name of the volume group on the shared storage (*vg_xen*). At the same time make sure that the *dir* line is commented out!

dist specifies the distribution to be installed in the virtual machines (Debian Lenny) (there's a comment in the file that explains what distributions are currently supported).

The *passwd = 1* line makes that you can specify a root password when you create a new guest domain.

In the *mirror* line specify a Debian mirror close to you.

Make sure you specify a gateway, netmask, and broadcast address. If you don't, and you don't specify a gateway and netmask on the command line when using *xen-create-image*, your guest domains won't have networking even if you specified an IP address!

It is very important that you add the line *serial_device = hvc0* because otherwise your virtual machines might not boot properly!

Now let's create our first guest domain, *vm1.example.com*, with the IP address *192.168.0.103*:

server1:

```
xen-create-image --hostname=vm1.example.com --size=4Gb --swap=256Mb --ip=192.168.0.103 --memory=128Mb --arch=amd64 --role=udev
```

```
server1:~# xen-create-image --hostname=vm1.example.com --size=4Gb --swap=256Mb --ip=192.168.0.103 --memory=128Mb --arch=amd64
 --role=udev

General Information
--------------------
Hostname       :  vm1.example.com
Distribution   :  lenny
Partitions     :  swap            256Mb (swap)
                  /               4Gb   (ext3)
Image type     :  full
Memory size    :  128Mb
Kernel path    :  /boot/vmlinuz-2.6.26-1-xen-amd64
Initrd path    :  /boot/initrd.img-2.6.26-1-xen-amd64

Networking Information
----------------------
IP Address 1   : 192.168.0.103 [MAC: 00:16:3E:4D:61:B6]
Netmask        : 255.255.255.0
Broadcast      : 192.168.0.255
Gateway        : 192.168.0.1


Creating swap on /dev/vg_xen/vm1.example.com-swap
Done
```



```
Creating ext3 filesystem on /dev/vg_xen/vm1.example.com-disk
Done
Installation method: debootstrap
Done

Running hooks
Done

Role: udev
        File: /etc/xen-tools/role.d/udev
Role script completed.

Creating Xen configuration file
Done
Setting up root password
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
All done


Logfile produced at:
         /var/log/xen-tools/vm1.example.com.log
server1:~#
```

As you see, the command has created two new logical volumes in the *vg_xen* volume group, */dev/vg_xen/vm1.example.com-disk* and */dev/vg_xen/vm1.example.com-swap*.

There should now be a configuration file for the *vm1.example.com* Xen guest in the */etc/xen* directory, *vm1.example.com.cfg*. Because we want to migrate the Xen guest from *server1* to *server2* later on, we must copy that configuration file to *server2*:

```
scp /etc/xen/vm1.example.com.cfg root@server2.example.com:/etc/xen/
```


Now we can start *vm1.example.com*:

```
xm create /etc/xen/vm1.example.com.cfg
```

5.1 Moving Existing Virtual Machines To The vg_xen Volume Group

If you want to do live migration for existing virtual machines that are not stored on the iSCSI shared storage, you must move them to the *vg_xen* volume group first. You can do this with *dd*, no matter if the guests are image- or LVM-based. This tutorial should give you the idea how to do this: **Xen: How to Convert An Image-Based Guest To An LVM-Based Guest**

6 Live Migration Of vm1.example.com From server1 To server2

To check if the live migration is really done "live", i.e. without interruption of the guest, you can log into *vm1.example.com* (e.g. with SSH) and ping another server:

vm1.example.com:

```
ping google.com
```

This will ping google.com until you press *CTRL + C*. The pinging should continue even during the live migration.

server1:

```
xm list
```

should show that *vm1.example.com* is currently running on *server1*:

```
server1:~# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  3628     2     r-----    115.6
```

```
vm1.example.com                          1   128     1     -b----      2.4
server1:~#
```

Before we migrate the virtual machine to *server2*, we must make sure that */dev/vg_xen/vm1.example.com-disk* and */dev/vg_xen/vm1.example.com-swap* are available on *server2*:

server2:

```
lvdisplay
```

```
server2:/etc/xen# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg_xen/vm1.example.com-swap
  VG Name                vg_xen
  LV UUID                ubgqAl-YSmJ-BiVl-YLKc-t4Np-VPl2-WG5eFx
  LV Write Access        read/write
  LV Status              NOT available
  # open                 1
  LV Size                256.00 MB
  Current LE             64
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:3

  --- Logical volume ---
  LV Name                /dev/vg_xen/vm1.example.com-disk
  VG Name                vg_xen
  LV UUID                4zNxf2-Pt16-cQO6-sqmt-kfo9-uSQY-55WN76
  LV Write Access        read/write
  LV Status              NOT available
```

```
  # open                 1
  LV Size                4.00 GB
  Current LE             1024
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:2

  --- Logical volume ---
  LV Name                /dev/vg0/root
  VG Name                vg0
  LV UUID                aQrAHn-ZqyG-kTQN-eYE9-2QBQ-IZMW-ERRvqv
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                100.00 GB
  Current LE             25600
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/vg0/swap_1
  VG Name                vg0
  LV UUID                9gXmOT-KP9j-21yw-gJPS-lurt-QlNK-WAL8we
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                1.00 GB
  Current LE             256
```

```
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:1

server2:/etc/xen#
```

As you see, the command shows NOT available for both volumes, so we must make them available:

```
lvscan

lvchange -a y /dev/vg_xen/vm1.example.com-disk

lvchange -a y /dev/vg_xen/vm1.example.com-swap
```

Now they should be available:

```
lvdisplay
```

```
server2:/etc/xen# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg_xen/vm1.example.com-swap
  VG Name                vg_xen
  LV UUID                ubgqAl-YSmJ-BiVl-YLKc-t4Np-VPl2-WG5eFx
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                256.00 MB
  Current LE             64
  Segments               1
```

```
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:3

  --- Logical volume ---
  LV Name                /dev/vg_xen/vm1.example.com-disk
  VG Name                vg_xen
  LV UUID                4zNxf2-Pt16-cQO6-sqmt-kfo9-uSQY-55WN76
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                4.00 GB
  Current LE             1024
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:2

  --- Logical volume ---
  LV Name                /dev/vg0/root
  VG Name                vg0
  LV UUID                aQrAHn-ZqyG-kTQN-eYE9-2QBQ-IZMW-ERRvqv
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                100.00 GB
  Current LE             25600
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
```



```
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/vg0/swap_1
  VG Name                vg0
  LV UUID                9gXmOT-KP9j-21yw-gJPS-lurt-QlNK-WAL8we
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                1.00 GB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:1

server2:/etc/xen#
```

```
xm list
```

should not list *vm1.example.com* yet on *server2*:

```
server2:~# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  3633     2     r-----     16.2
server2:~#
```

Now we can start the live migration:

server1:


```
xm migrate --live vm1.example.com server2.example.com
```

During the migration, the pings on *vm1.example.com* should continue which means that the guest is running even during the migration process.

Afterwards, take a look at

```
xm list
```

```
server1:~# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  3626     2     r-----    118.2
server1:~#
```

As you see, *vm1.example.com* isn't listed anymore on *server1*.

Let's check on *server2*:

server2:

```
xm list
```

```
server2:~# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  3633     2     r-----     19.4
vm1.example.com                              1   128     1     --p---      0.0
server2:~#
```

If everything went well, *vm1.example.com* should now be running on *server2*.

7 Links

- Xen: **http://www.xensource.com/xen/**
- Open-iSCSI: **http://www.open-iscsi.org/**
- iSCSI Enterprise Target: **http://iscsitarget.sourceforge.net/**
- Debian: **http://www.debian.org/**