By Falko Timme
Published: 2007-04-15 13:24

Back Up (And Restore) LVM Partitions With LVM Snapshots

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 04/11/2007

This tutorial shows how you can create backups of LVM partitions with an LVM feature called LVM snapshots. An LVM snapshot is an exact copy of an LVM partition that has all the data from the LVM volume from the time the snapshot was created. The big advantage of LVM snapshots is that you don't have to worry about open files and database connections, and you don't have to interrupt/halt services on the live partition because a snapshot is usually created in fractions of a second, so your users won't notice any disruption, and your snapshot holds consistent data.

I will also show how to restore an LVM partition from a backup in an extra chapter at the end of this tutorial.

This document comes without warranty of any kind! I do not issue any guarantee that this will work for you!

1 Preliminary Note

I have tested this on a Debian Etch server with the IP address *192.168.0.100* and the hostname *server1.example.com*. It has two hard disks:

- */dev/sda* (10GB) that contains a small */boot* partition (non-LVM), a */* partition (LVM, a little less than 10GB), and a swap partition (LVM)
- */dev/sdb* (60GB), unused at the moment; will be used to create a 30GB */backups* partition (LVM) and for the snapshots of the */* partition (10GB - that's enough because the */* partition is a little less than 10GB).

I have created a **Debian Etch VMware image** that you can download and run in VMware Server or VMware Player (see **http://www.howtoforge.com/import_vmware_images** to learn how to do that). It has the same specifications as my test system from above. The root password is *howtoforge*. Using that VMware image, you can do the exact same steps than me in this tutorial to get used to using LVM snapshots.

To restore the / partition from your backup (covered in the last chapter of this tutorial) you need a Linux Live-CD that supports LVM, such as **Knoppix** or

the Debian Etch Netinstall CD which you can use as a rescue CD if you specify rescue at the boot prompt. I will use the Debian Etch Netinstall CD in this example (the list of mirrors is available here: **http://www.debian.org/CD/http-ftp/** - I downloaded this one: **http://ftp.de.debian.org/debian-cd/4.0_r0/i386/iso-cd/debian-40r0-i386-netinst.iso**).

To create a backup of the */* partition I will proceed as follows: I will create a snapshot of the */* partition, and afterwards I will create a backup of the snapshot (instead of the actual */* partition!) on the */backups* partition (of course, you can store that backup wherever you want - instead of creating an extra */backups* LVM partition, you could also use an external USB drive). The backup can be made using your preferred backup solution, e.g. with *tar* or *dd*. Afterwards, I'll destroy the snapshot because it isn't needed anymore and would use system resources.

You don't necessarily need a second HDD for the snapshots - you can use the first one provided you have enough free (unpartitioned) space left on it to create snapshots on it (you should use the same space for the snapshots that you use for the partition that you want to back up). And as mentioned before, you can use a USB drive for backing up the snapshots.

To learn more about LVM, you should read this tutorial: **http://www.howtoforge.com/linux_lvm**

2 Create The /backups LVM Partition

(If you'd like to store your backups somewhere else, e.g on an external USB drive, you don't have to do this.)

Our current situation is as follows:

```
pvdisplay
```

```
server1:~# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda5
  VG Name               server1
  PV Size               9.76 GB / not usable 0
  Allocatable           yes (but full)
  PE Size (KByte)       4096
  Total PE              2498
  Free PE               0
```



```
  Allocated PE          2498
  PV UUID               vQIUga-221O-GIKj-81Ct-2ITT-bKPw-kKElpM
```

```
vgdisplay
```

```
server1:~# vgdisplay
  --- Volume group ---
  VG Name               server1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               9.76 GB
  PE Size               4.00 MB
  Total PE              2498
  Alloc PE / Size       2498 / 9.76 GB
  Free  PE / Size       0 / 0
  VG UUID               jkWyez-c0nT-LCaE-Bzvi-Q4oD-eD3Q-BKIOFC
```

```
lvdisplay
```

```
server1:~# lvdisplay
  --- Logical volume ---
```

```
  LV Name                /dev/server1/root
  VG Name                server1
  LV UUID                UK1rjH-LS3l-f7aO-240S-EwGw-0Uws-5ldhlW
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                9.30 GB
  Current LE             2382
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/server1/swap_1
  VG Name                server1
  LV UUID                2PASi6-fQV4-I8sJ-J0yq-Y9lH-SJ32-F9jHaj
  LV Write Access        read/write
  LV Status              available
  # open                 2
  LV Size                464.00 MB
  Current LE             116
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:1
```

```
fdisk -l
```

```
server1:~# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
```

```
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          31      248976   83  Linux
/dev/sda2              32        1305    10233405    5  Extended
/dev/sda5              32        1305    10233373+  8e  Linux LVM

Disk /dev/sdb: 64.4 GB, 64424509440 bytes
255 heads, 63 sectors/track, 7832 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdb doesn't contain a valid partition table

Disk /dev/dm-0: 9990 MB, 9990832128 bytes
255 heads, 63 sectors/track, 1214 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

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

Disk /dev/dm-1: 486 MB, 486539264 bytes
255 heads, 63 sectors/track, 59 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

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

So `/dev/sda` contains the logical volumes `/dev/server1/root` (`/` partition) and `/dev/server1/swap_1` (swap partition) plus a small `/boot` partition (non-LVM).

(BTW, `/dev/server1/root` is the same as `/dev/mapper/server1-root` on Debian Etch. The first is a symlink to the second; I will use both notations in this tutorial. The same goes for `/dev/server1/swap_1` and `/dev/mapper/server1-swap_1`.)

I will now create the partition `/dev/sdb1` and add it to the `server1` volume group, and afterwards I will create the volume `/dev/server1/backups` (which

will be 30GB instead of the full 60GB of */dev/sdb* so that we have enough space left for the snapshots) which I will mount on */backups*:

```
fdisk /dev/sdb
```

```
server1:~# fdisk /dev/sdb
  Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
  Building a new DOS disklabel. 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 7832.
  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):
Command action
   e   extended
   p   primary partition (1-4)

Partition number (1-4):
First cylinder (1-7832, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-7832, default 7832):
Using default value 7832

Command (m for help):
Selected partition 1
```



```
Hex code (type L to list codes):
Changed system type of partition 1 to 8e (Linux LVM)

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

Calling ioctl() to re-read partition table.
Syncing disks.
```

```
pvcreate /dev/sdb1

vgextend server1 /dev/sdb1

lvcreate --name backups --size 30G server1

mkfs.ext3 /dev/mapper/server1-backups

mkdir /backups
```

Now let's mount our */dev/server1/backups* volume on */backups*:

```
mount /dev/mapper/server1-backups /backups
```

To have that volume mounted automatically whenever you boot the system, you must edit */etc/fstab* and add a line like this to it:

```
vi /etc/fstab
```

```
[...]
/dev/mapper/server1-backups /backups              ext3    defaults,errors=remount-ro 0       1
```

Now our new situation looks like this:

```
pvdisplay
```

```
server1:~# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda5
  VG Name               server1
  PV Size               9.76 GB / not usable 0
  Allocatable           yes (but full)
  PE Size (KByte)       4096
  Total PE              2498
  Free PE               0
  Allocated PE          2498
  PV UUID               vQIUga-221O-GIKj-81Ct-2ITT-bKPw-kKElpM

  --- Physical volume ---
  PV Name               /dev/sdb1
  VG Name               server1
  PV Size               59.99 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              15358
  Free PE               7678
  Allocated PE          7680
  PV UUID               cvl1H5-cxRe-iyNg-m2mM-tjxM-AvER-rjqycO
```

```
vgdisplay
```

```
server1:~# vgdisplay
  --- Volume group ---
```

```
  VG Name               server1
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                3
  Open LV               3
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               69.75 GB
  PE Size               4.00 MB
  Total PE              17856
  Alloc PE / Size       10178 / 39.76 GB
  Free  PE / Size       7678 / 29.99 GB
  VG UUID               jkWyez-c0nT-LCaE-Bzvi-Q4oD-eD3Q-BKIOFC
```

```
lvdisplay
```

```
server1:~# lvdisplay
  --- Logical volume ---
  LV Name                /dev/server1/root
  VG Name                server1
  LV UUID                UK1rjH-LS3l-f7aO-240S-EwGw-0Uws-5ldhlW
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                9.30 GB
  Current LE             2382
```

```
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/server1/swap_1
  VG Name                server1
  LV UUID                2PASi6-fQV4-I8sJ-J0yq-Y9lH-SJ32-F9jHaj
  LV Write Access        read/write
  LV Status              available
  # open                 2
  LV Size                464.00 MB
  Current LE             116
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:1

  --- Logical volume ---
  LV Name                /dev/server1/backups
  VG Name                server1
  LV UUID                sXq2Xe-y2CE-Ycko-rCoE-M5kl-E1vH-KQRoP6
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                30.00 GB
  Current LE             7680
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:2
```

3 Create An LVM Snapshot Of /

Now it's time to create the snapshot of the */dev/server1/root* volume. We will call the snapshot *rootsnapshot*:

```
lvcreate -L10G -s -n rootsnapshot /dev/server1/root
```

The output of

```
lvdisplay
```

should look like this:

```
server1:~# lvdisplay
  --- Logical volume ---
  LV Name                /dev/server1/root
  VG Name                server1
  LV UUID                UK1rjH-LS3l-f7aO-240S-EwGw-0Uws-5ldhlW
  LV Write Access        read/write
  LV snapshot status     source of
                         /dev/server1/rootsnapshot [active]
  LV Status              available
  # open                 1
  LV Size                9.30 GB
  Current LE             2382
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/server1/swap_1
```



```
  VG Name                server1
  LV UUID                2PASi6-fQV4-I8sJ-J0yq-Y9lH-SJ32-F9jHaj
  LV Write Access        read/write
  LV Status              available
  # open                 2
  LV Size                464.00 MB
  Current LE             116
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:1

  --- Logical volume ---
  LV Name                /dev/server1/backups
  VG Name                server1
  LV UUID                sXq2Xe-y2CE-Ycko-rCoE-M5kl-E1vH-KQRoP6
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                30.00 GB
  Current LE             7680
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:2

  --- Logical volume ---
  LV Name                /dev/server1/rootsnapshot
  VG Name                server1
  LV UUID                9zR5X5-OhM5-xUI0-OolP-vLjG-pexO-nk36oz
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/server1/root
  LV Status              available
```

```
  # open                 1
  LV Size                9.30 GB
  Current LE             2382
  COW-table size         10.00 GB
  COW-table LE           2560
  Allocated to snapshot  0.01%
  Snapshot chunk size    8.00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:5
```

We want to mount */dev/server1/rootsnapshot* on */mnt/server1/rootsnapshot*, so we have to create that directory first:

```
mkdir -p /mnt/server1/rootsnapshot
```

Then we mount our snapshot:

```
mount /dev/server1/rootsnapshot /mnt/server1/rootsnapshot
```

Then we run

```
ls -l /mnt/server1/rootsnapshot/
```

This should show all directories and files that we know from our */* partition:

```
server1:~# ls -l /mnt/server1/rootsnapshot/
total 132
drwxr-xr-x  2 root root  4096 2007-04-10 21:02 backups
drwxr-xr-x  2 root root  4096 2007-04-10 20:35 bin
```

```
drwxr-xr-x  2 root root  4096 2007-04-10 20:25 boot
lrwxrwxrwx  1 root root    11 2007-04-10 20:25 cdrom -> media/cdrom
drwxr-xr-x 13 root root 40960 2007-04-10 20:36 dev
drwxr-xr-x 57 root root  4096 2007-04-10 21:09 etc
drwxr-xr-x  3 root root  4096 2007-04-10 20:36 home
drwxr-xr-x  2 root root  4096 2007-04-10 20:26 initrd
lrwxrwxrwx  1 root root    28 2007-04-10 20:29 initrd.img -> boot/initrd.img-2.6.18-4-486
drwxr-xr-x 13 root root  4096 2007-04-10 20:34 lib
drwx------  2 root root 16384 2007-04-10 20:25 lost+found
drwxr-xr-x  4 root root  4096 2007-04-10 20:25 media
drwxr-xr-x  2 root root  4096 2006-10-28 16:06 mnt
drwxr-xr-x  2 root root  4096 2007-04-10 20:26 opt
drwxr-xr-x  2 root root  4096 2006-10-28 16:06 proc
drwxr-xr-x  3 root root  4096 2007-04-10 20:42 root
drwxr-xr-x  2 root root  4096 2007-04-10 20:36 sbin
drwxr-xr-x  2 root root  4096 2007-03-07 23:56 selinux
drwxr-xr-x  2 root root  4096 2007-04-10 20:26 srv
drwxr-xr-x  2 root root  4096 2007-01-30 23:27 sys
drwxrwxrwt  2 root root  4096 2007-04-10 21:09 tmp
drwxr-xr-x 10 root root  4096 2007-04-10 20:26 usr
drwxr-xr-x 13 root root  4096 2007-04-10 20:26 var
lrwxrwxrwx  1 root root    25 2007-04-10 20:29 vmlinuz -> boot/vmlinuz-2.6.18-4-486
```

So our snapshot has successfullly been created!

Now we can create a backup of the snapshot on the `/backups` partition using our preferred backup solution. For example, if you like to do a file-based backup, you can do it like this:

```
tar -pczf /backups/root.tar.gz /mnt/server1/rootsnapshot
```

And if you like to do a bitwise backup (i.e. an image), you can do it like this:

```
dd if=/dev/server1/rootsnapshot of=/backups/root.dd
```

```
server1:~# dd if=/dev/server1/rootsnapshot of=/backups/root.dd
19513344+0 records in
19513344+0 records out
9990832128 bytes (10 GB) copied, 320.059 seconds, 31.2 MB/s
```

You could also use both ways to be prepared for whatever might happen to your */dev/server1/root* volume. In this case, you should have two backups afterwards:

```
ls -l /backups/
```

```
server1:~# ls -l /backups/
total 9947076
drwx------ 2 root root      16384 2007-04-10 21:04 lost+found
-rw-r--r-- 1 root root 9990832128 2007-04-10 21:28 root.dd
-rw-r--r-- 1 root root  184994590 2007-04-10 21:18 root.tar.gz
```

Afterwards, we unmount and remove the snapshot to prevent it from consuming system resources:

```
umount /mnt/server1/rootsnapshot

lvremove /dev/server1/rootsnapshot
```

That's it, you've just made your first backup from an LVM snapshot.

4 Restore A Backup

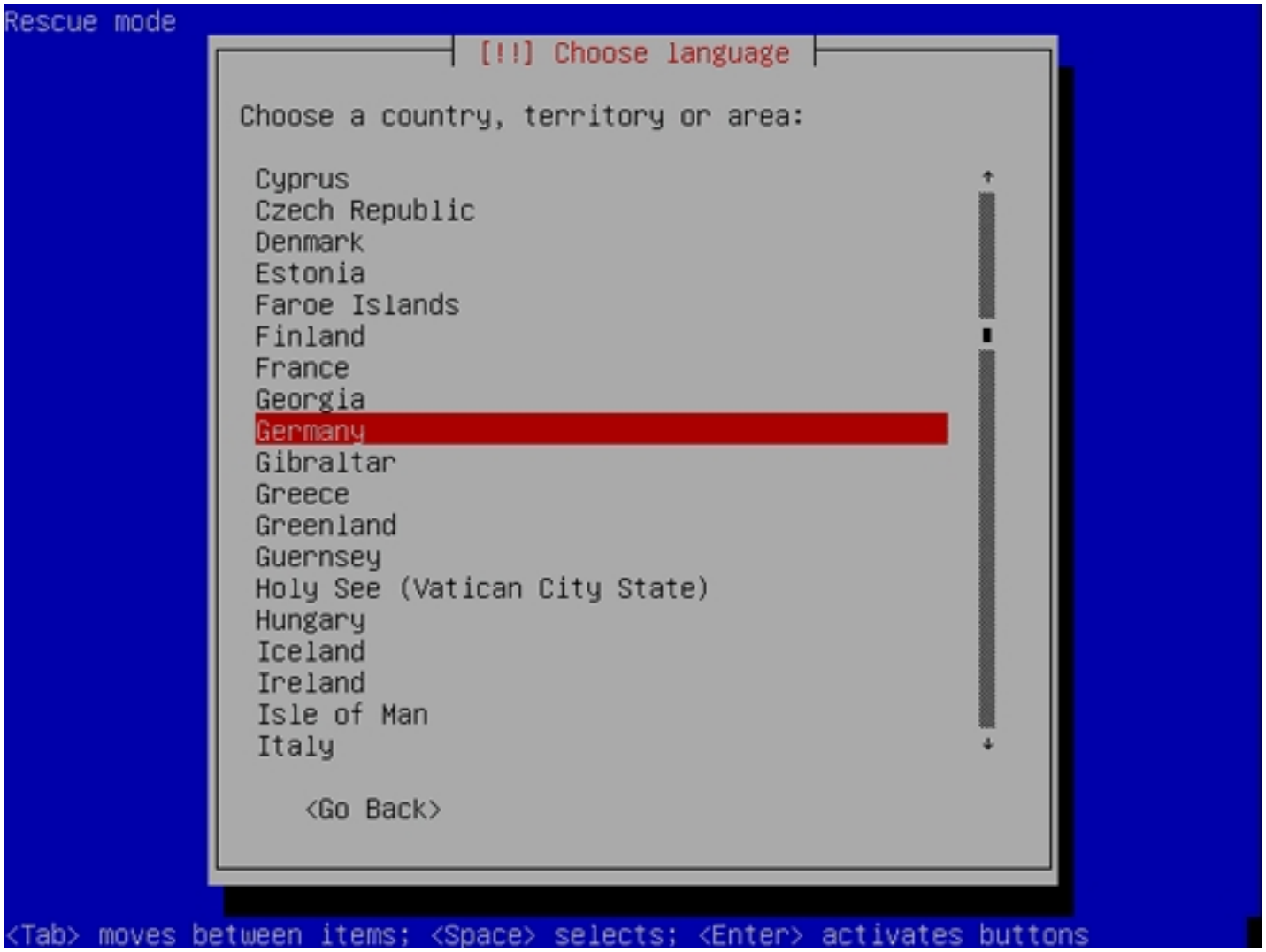
This chapter is about restoring the */dev/server1/root* volume from the *dd* image we've created in the previous chapter. Normally you can restore a backup from the same running system if the volume that you want to restore doesn't contain system-critical files. But because the */dev/server1/root*

volume is the system partition of our machine, we must use a rescue system or Live-CD to restore the backup. The rescue system/Live-CD must support LVM.

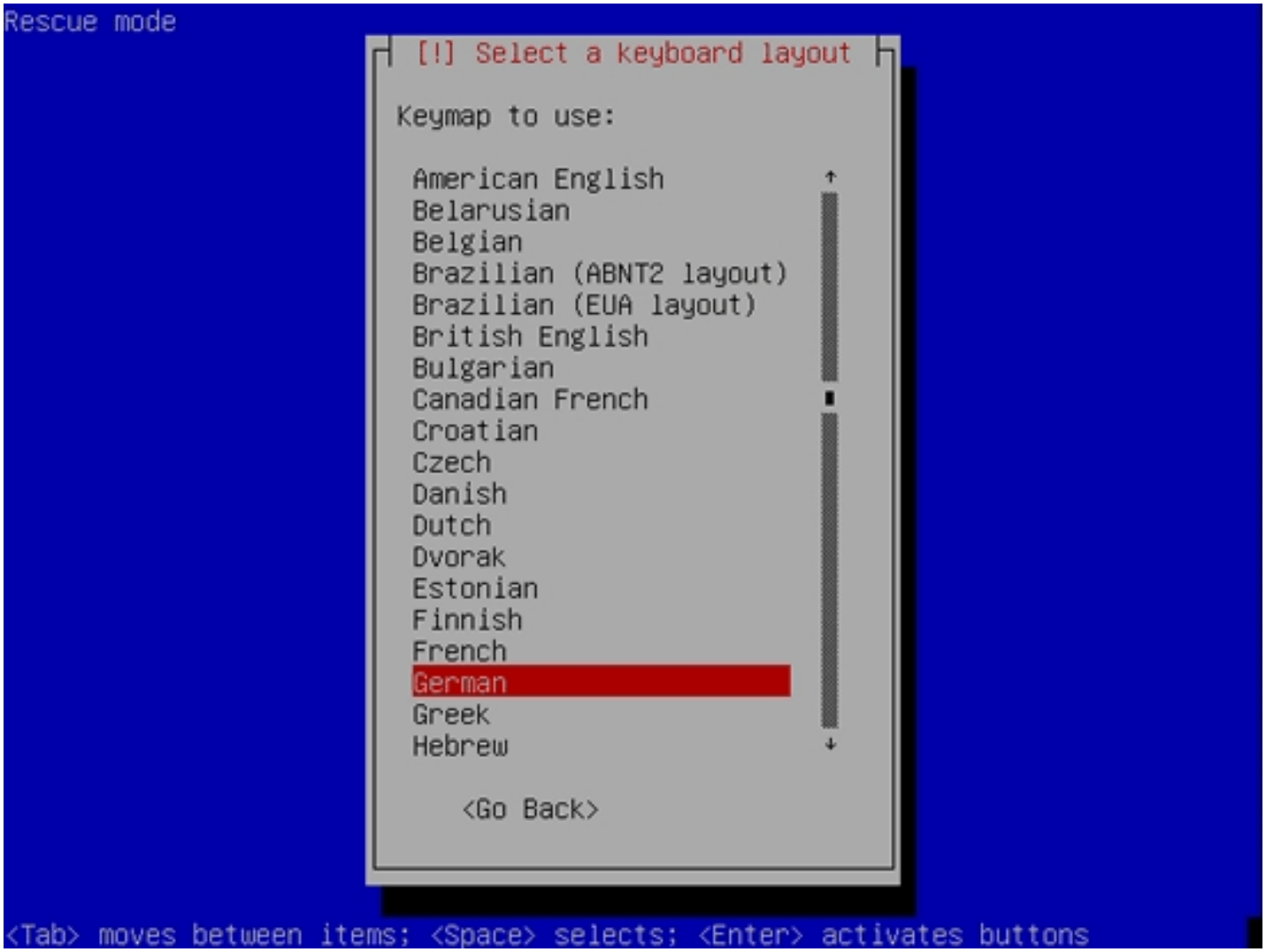
To restore the `/dev/server1/root` volume, I boot the system from the Debian Etch Netinstall CD and type in `rescue` at the boot prompt:

Select your language:

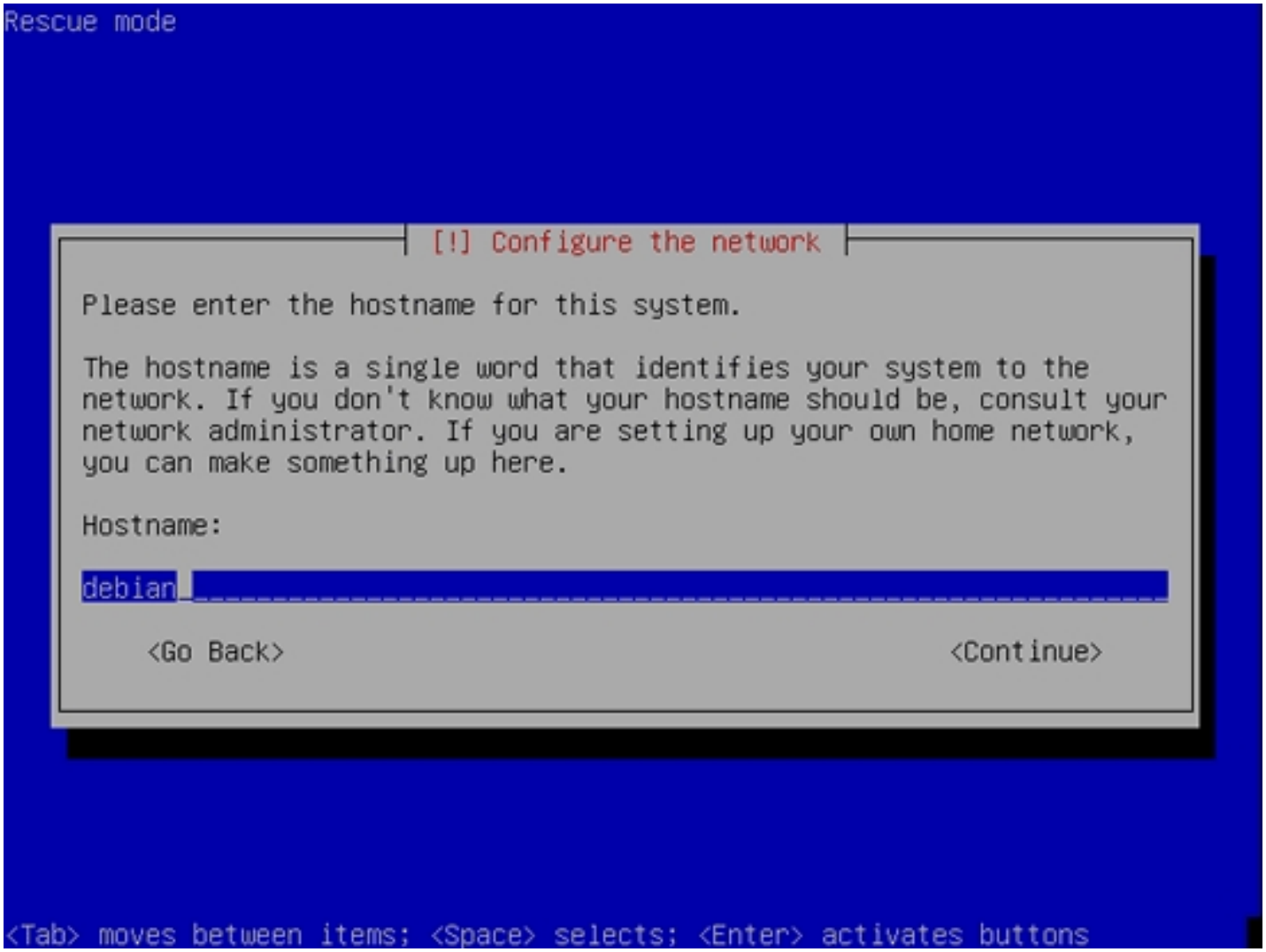
Choose your country:



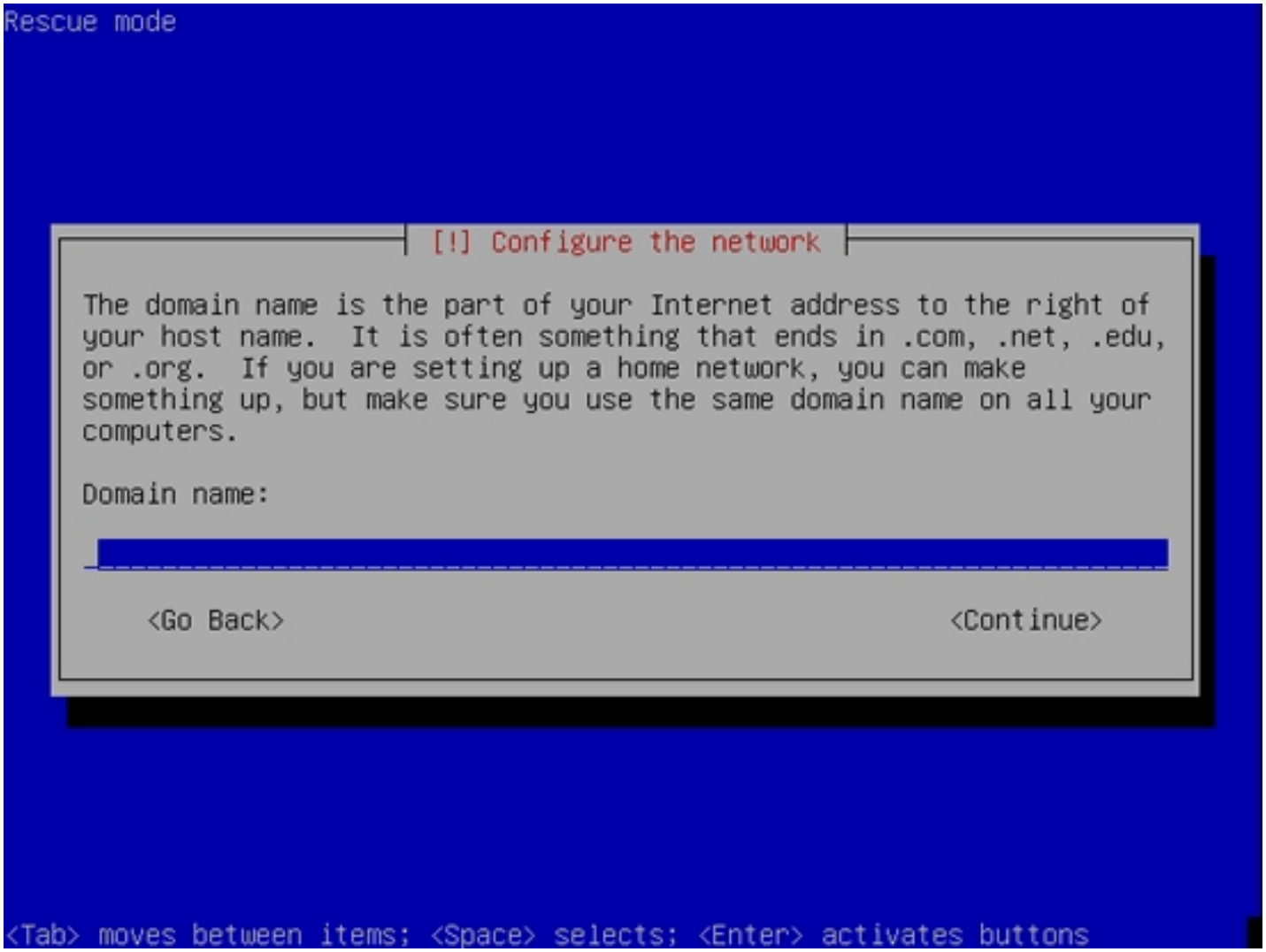
Choose your keyboard layout:



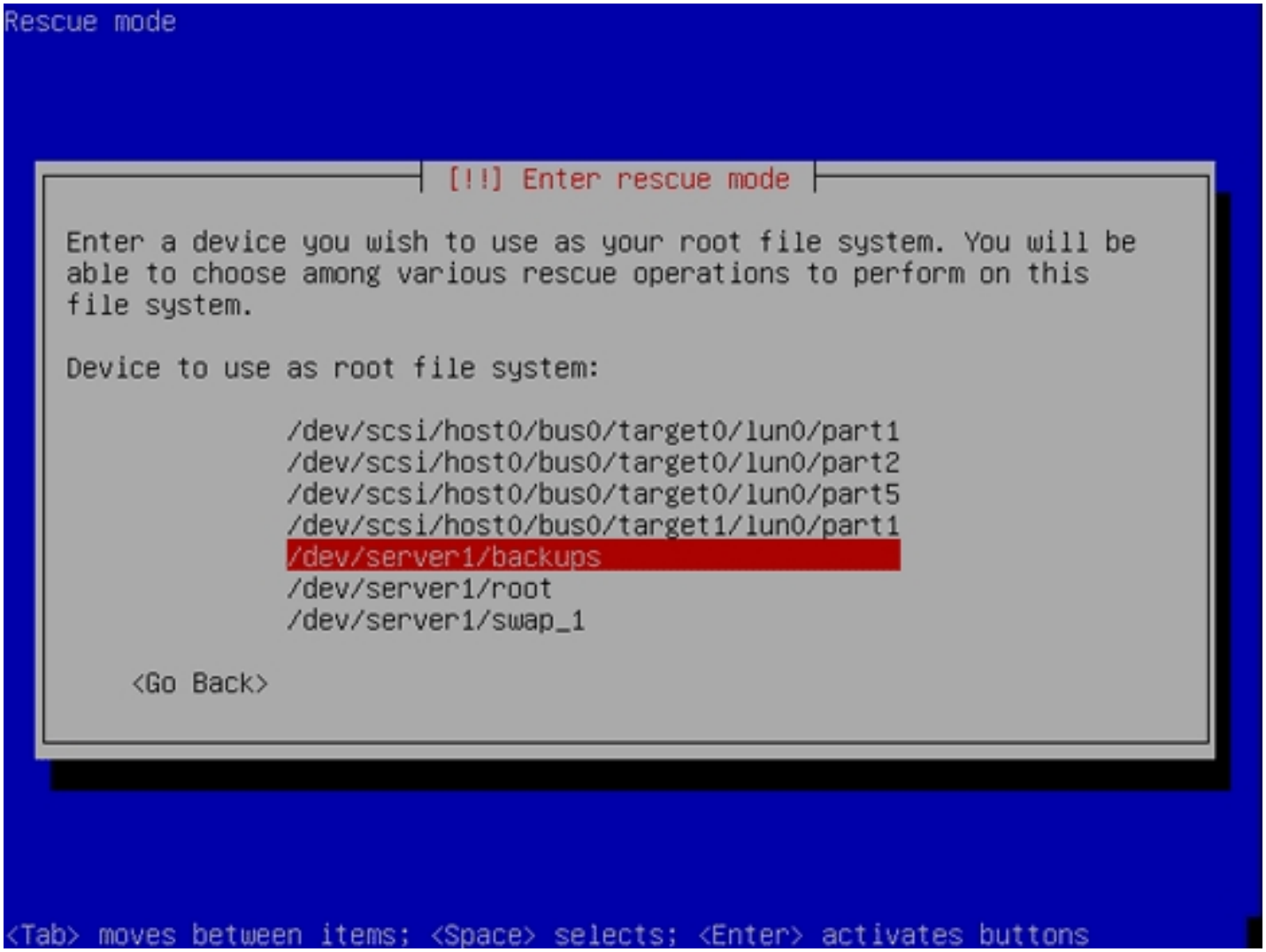
You can accept the default hostname:



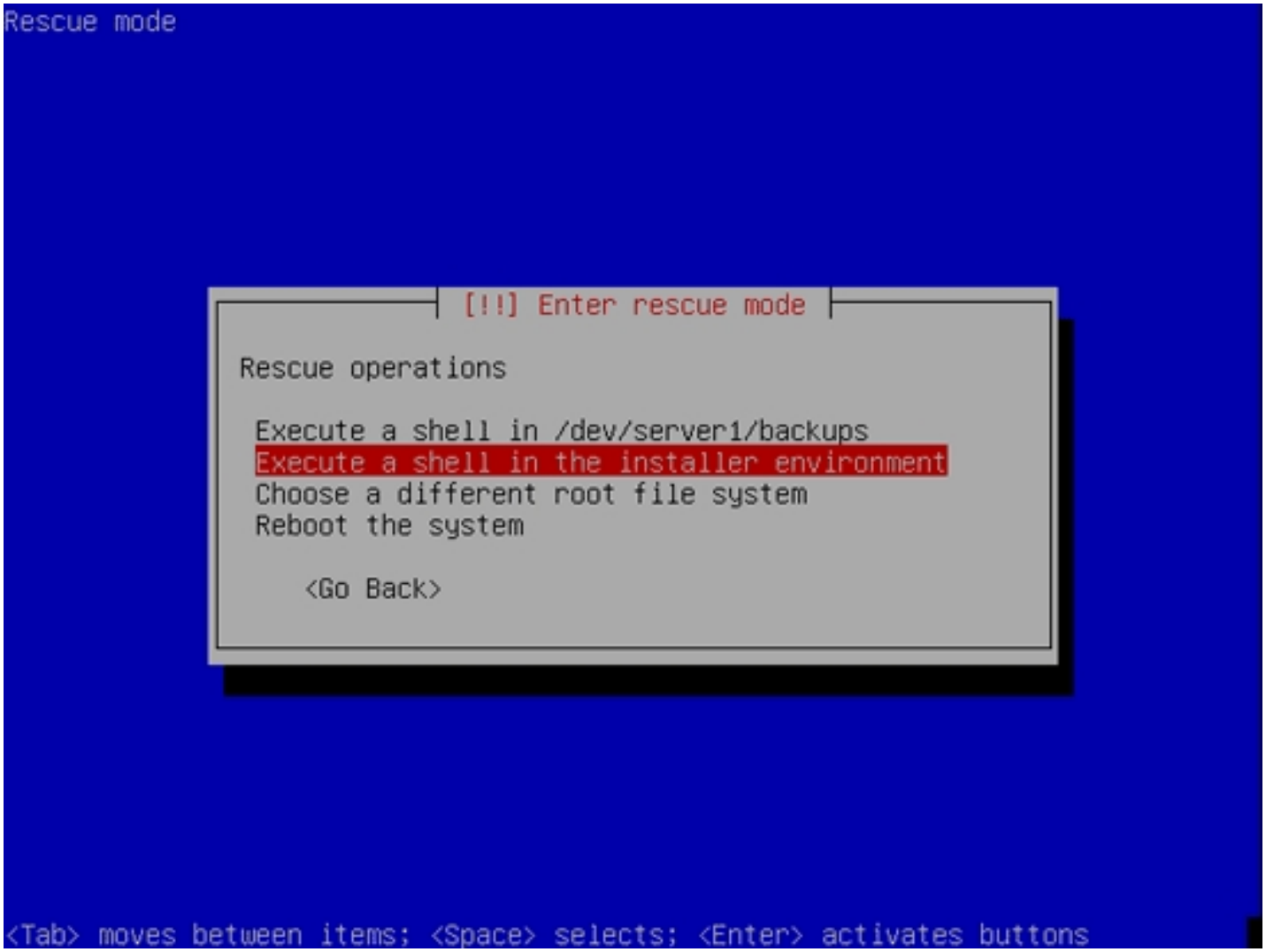
You can also accept the default domain name (which is empty):



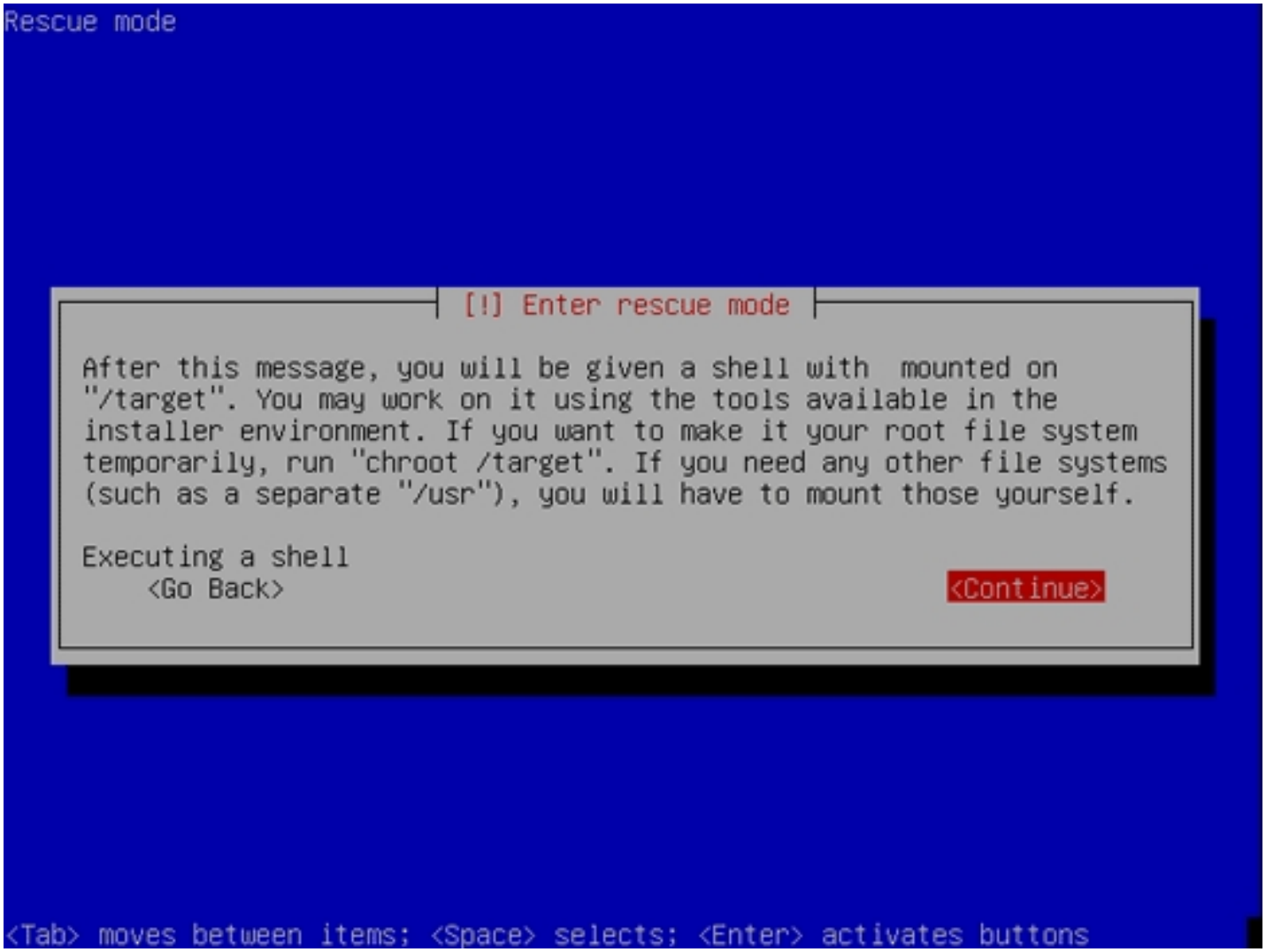
Select the backup volume (*/dev/server1/backups*) as the root file system:



Then select *Execute a shell in the installer environment*:



Hit *Continue*:



Now we have a shell:

Run

```
mount
```

and you should see that */dev/server1/backups* is mounted on */target*. So the *dd* image of the */dev/server1/root* volume should be */target/root.dd*. To restore it, we simply run

```
dd if=/target/root.dd of=/dev/server1/root
```

That's it. It can take a few minutes until the task is finished. Afterwards you can remove the Live-CD and boot into the normal system again.

5 Links

- Debian: **http://www.debian.org**
- Debian Etch VMware Image for this Tutorial: **http://downloads.howtoforge.com/files/Debian_Etch_LVM_Snapshots.zip**
- HowtoForge LVM Tutorial: **http://www.howtoforge.com/linux_lvm**