Pro Ubuntu Server Administration

Copyright © 2009 by Sander van Vugt

ISBN-13 (pbk): 978-1-4302-1622-3

ISBN-13 (electronic): 978-1-4302-1623-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

CHAPTER 1

Performing an Advanced Ubuntu Server Installation

Installing Ubuntu Server with RAID

You know how to install Ubuntu Server. There are, however, some additional challenges that you may face when installing Ubuntu Server in a network. Most important of those challenges is that your server may need a software-based RAID solution. If you want to configure your server with software RAID, and especially if you want to use LVM volumes on top of that, installing Ubuntu Server can be quite hard. In this chapter you'll learn all you need to know about such an installation.

What's So Special About an Enterprise Installation?

You may ask: what's the big deal about an enterprise network installation of Ubuntu Server versus a "normal" Ubuntu Server installation? There are some important differences when installing Ubuntu Server in an enterprise environment in which other servers are used as well, as this section explains. First, take a look at the recommended minimal installation requirements for a normal server installation:

- 256 MB of RAM
- 500 MHz CPU
- 4 GB hard drive
- Optical drive

The next few sections discuss some of the most significant differences between a network installation and a simple stand-alone installation.

Server Hardware

The first major difference between a demo installation in your test network and an enterprise network installation is in the server hardware itself. When setting up a server in an enterprise environment, you probably want some redundancy. You can implement that redundancy by making sure that some devices have a backup available. For example, most data-center-grade servers have a dual power supply, two network cards, and at least two hard disks. The advantage? If one breaks, the server can start using the other. And the big deal is that all of this happens automatically.

Some of the setup of this redundant hardware is done in the hardware itself. I don't cover that in this book. Some setup can be software based as well. For example, the use of software RAID, or NIC teaming (also known as NIC bonding), makes sure that two network boards are presented as one single network interface. The purpose of that ? It can add redundancy to your network card, or if you prefer, it can increase performance because two network cards bundled together can handle twice the workload of a single network card working alone.

Connection to a SAN

Next, your server may be connected to a storage area network (SAN). If you've never worked with a SAN before, no worries—just consider it a bunch of external disks for the moment. Chapter 7 covers in depth setting up Ubuntu Server as a SAN. Typically, a specialized network card called a host bus adapter (HBA) takes care of the connection to a SAN. Such a host adapter may use iSCSI, which sends SCSI packets encapsulated in IP over a copper-based network, or it may be a Fibre Channel card, using an expensive Fibre Channel infrastructure.

If your server is connected to a SAN, you normally would want to have some redundancy in the SAN as well. This redundancy is implemented by using multiple HBAs that connect to the SAN using different network connections. Now, there is something unique about this scenario. Normally, when the HBA in your server connects to the SAN, it gets an additional storage device. For instance, if you have a local hard disk in your server, you would normally see it as the device `/dev/sda`. If just one HBA connects to the shared storage on the SAN, the HBA would offer your server access to an external hard drive, which would be seen by your server as a new storage device, typically `/dev/sdb`.

Now imagine the situation in which two different HBAs use separate network connections to connect to the same shared storage area on the SAN. Each of the two HBAs would give you an additional external device, so you would see an additional `/dev/sdb` and `/dev/sdc`. There is one problem with that, though: both `/dev/sdb` and `/dev/sdc` would refer to the same storage device! That normally is not a good idea, and that is where multipath comes in. When using multipath, an additional kernel module is loaded. The purpose of this module is to tell the operating system that the devices `/dev/sdb` and `/dev/sdc` (in this example) are just the same device. As you can understand, when connecting your server to a redundant SAN, the configuration of multipath is an absolute requirement.

Authentication Handling

One last difference when installing your server in a network environment is that typically you would implement an external authentication mechanism. If you have only one server, it makes perfect sense to handle user authentication on that server itself. However, if you have more than one server, it makes sense to use a service that takes care of authentication for you at a centralized location in the network. This refers to a server that has already been set up in the network for this purpose. Such a service might be your LDAP server or a Microsoft Active Directory environment. The Ubuntu Server installation process helps you to set that up as well. In the next section you'll read all about it.

Preparing for the Installation in a Network

You now know what to take care of when installing Ubuntu Server in a network environment. So let's talk about the installation itself. In this section you'll read how a typical server installation in a network environment takes place. I'll assume that you have installed Ubuntu Server before, so I'll be rather brief on the obvious parts, and more in depth with regard to the advanced parts of the installation. Before you start the actual installation, you should understand what I'm going to install here for purposes of demonstration.

The server that you are going to read about in this section has the following properties:

- Two quad-core processors
- 8 GB of RAM
- Five disks
- Two Gigabit Ethernet network boards

Note You may not have the hardware described here available. That's no problem, because you can create a configuration like this rather easily using virtualization software like VMware. Okay, it's a problem to create two virtual CPUs with quad core each, and it will be a problem allocating 8 GB of RAM in most situations as well, but processors and RAM don't make that big of a difference when performing the installation anyway. The focus here is on disk and network setup. And using a free virtualization solution like VMware Server, you can just create as many disks and as many Ethernet network boards as you like.

It's fine if your server has additional properties, but from the installation perspective, having the preceding list of properties really is all that matters. Before you insert the installation CD and start the installation, it helps to make a plan. Most important is the planning of your disk setup. In a typical server installation, what you want above all is redundancy and performance at the same time. This means that if a disk breaks, the other disks should take over immediately. To reach this goal, you would probably want to work with some kind of RAID setup.

Which RAID?

There are two ways to set up RAID on your server: hardware based and software based. If your server has a hardware RAID controller, you should consult the documentation for that controller. Every RAID controller is different, and there is no generic way in which I can describe how to set that up. If your server does not have hardware RAID, you can use a software-based RAID solution. Software RAID often does not offer the same level of performance as hardware RAID, but the advantage is that you don't have to pay anything extra to use it. When implementing software RAID, the following four methods are of interest:

- *RAID 0*: This RAID method is referred to as disk striping. Actually, RAID 0 just bundles two disks together. This is excellent for performance, because you have two controllers that can handle the data flow simultaneously, but RAID 0 is not built for redundancy and fault tolerance. If one disk in a RAID 0 array breaks, you can't access any data on the array anymore.
- *RAID 1*: RAID 1 is all about disk mirroring. One disk is used as the active disk and handles all I/O; the other disk is used only as a hot backup disk. Everything that happens on the active disk happens on the backup disk as well, so at all times, the backup disk will be the same. Therefore, if the active disk fails, the backup disk can take over easily. This is a very safe method of working, but it doesn't offer the best performance. Therefore, especially if you are in an environment in which lots of files are written to the storage devices, you either should not use RAID 1 or should create a RAID 1 array that uses two controllers to increase write speed on the RAID. For rather static volumes, however, RAID 1 is an excellent solution.

- *RAID 10*: RAID 10 offers you the best of both worlds: it's RAID 0 with RAID 1 behind it. So, you have excellent performance and excellent fault tolerance at the same time. There is one disadvantage, though: you need a minimum of four disks to set it up.
- *RAID 5*: If you need to write huge amounts of data, RAID 5 is what you need. To set up RAID 5, you need a minimum of three disks. When a file is written, it is spread over two of the three disks, and the third disk is used to write parity information for this file. Based on this parity information, if something goes wrong with one of the disks in the RAID 5 array, the RAID software is always able to reconstruct the data in a very fast way. To promote optimal performance, in RAID 5 the parity information is spread over all the disks in the array. So there is no dedicated disk that stores this information, and that promotes very good performance as well.

Note The parity information that is used in a RAID setup creates some kind of a checksum for all files on the RAID. If a disk in the RAID gets lost, the original file can be reconstructed based on the parity information.

Apart from the RAID technologies mentioned here, there are other RAID solutions as well. However, everything else relates in some way to the techniques mentioned here. In the example that I'll show in this chapter, you will install a server that has a RAID 1 array for the system files and a RAID 5 array to store data files.

On top of the RAID arrays, you need some disk storage mechanism as well. Basically, there are two options: use logical volumes or use traditional partitions. Especially for data volumes, it is a very clever idea to use logical partitions. Not only are these easily resizable, but they also offer the snapshot feature. Using snapshot technology makes it a lot easier to make a backup of open files. Most backup programs have a problem backing up files that are in use, whereas if you use snapshot technology, you can freeze the status of your volumes, which allows you to back up anything on the snapshot. Also, when using logical volumes, you can go far beyond the maximum of 16 partitions that you can create when using traditional partitions. The one disadvantage is that you can't boot from a logical volume.

Choosing a File System

Next, you need to consider what you want to do on top of these logical volumes. In all cases, you need to format the logical volume so that a file system is created that allows you to store files on your server. Typically, the following file systems are available:

- *Ext2*: The traditional Linux file system since shortly after Linux was created. It is very stable, but does not offer journaling functionality, which means that it can take a (very) long time to repair the file system in case of problems. Use Ext2 on small volumes that are mainly read-only.
- *Ext3*: Basically, Ext3 is just Ext2 with journaling added to it. Journaling allows you to recover very fast in case problems do occur. Ext3 is a good solution to store data, but it is not the best file system when you are using many (read: more than about 5,000) files in one directory. The indexing methodology used in Ext3 is rather limited.
- *XFS*: XFS was created by supercomputer manufacturer SGI as an open source file system. The most important property of XFS is that it is meant for "large." That means large files, large amounts of data, and large file systems. XFS also is a complete 64-bit file system. It has some excellent tuning options as well, a journal, and a very well-tuned index. All that makes XFS currently the best solution to store data files.
- *ReiserFS*: In the late 1990s Hans Reiser created ReiserFS, a revolutionary file system that was organized in a totally different way compared to the early file systems that were available at that time. Because of this completely new approach, ReiserFS offered supreme performance, especially in environments in which many small files had to be handled. Some other (minor) issues were addressed as well and that made it a very nice file system for data volumes. However, kernel support for ReiserFS has never been great and that has lead to stability issues. In specific environments in which many large files need to be handled, ReiserFS may still be a good choice, but be aware that ReiserFS is not very stable and you will have problems with it sooner or later.
- *JFS*: Journaled File System was developed by IBM as one of the first file systems that offered journaling. The development of this file system has stopped, however, and therefore I don't recommend its use on new servers.

Based on the preceding information, you should now be capable of creating a blueprint for the disk layout that your server is going to use. Table 1-1 provides an overview of what I'm going to install on my server in this chapter. The items in parentheses are recommended sizes when working from a VMware test environment or any other test environment in which available storage is limited.

Note Chapter 4 covers advanced file system management tasks. ReiserFS management is included as well. Normally I wouldn't recommend using ReiserFS anymore, but to make it easier for you to apply the contents of Chapter 4, in the example setup, I'm setting up a ReiserFS file system as well.

Table 1-1. *Blueprint of Server Disk Layout*

Directory	Size	File System	Storage Back End	Storage Device
`/boot`	100 MB (50 MB)	Ext2	Primary partition	RAID 1
`/`	20 GB (5 GB)	Ext3	LVM volume	RAID 1
`/var`	5 GB (1 GB)	XFS	LVM volume	RAID 1
`/tmp`	1 GB (1 GB)	ReiserFS	LVM volume	RAID 1
`/srv`	50 GB (1 GB)	XFS	LVM volume	RAID 5
`/home`	50 GB (1 GB)	XFS	LVM volume	RAID 5

Now that we've done our homework, it's time to start. In the next section you'll read how to actually install this configuration.

Installing Ubuntu Server

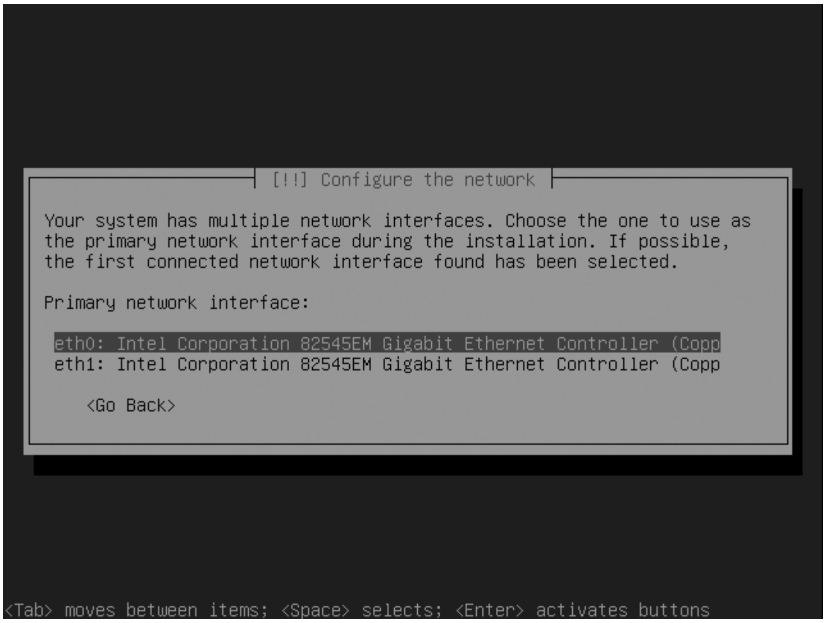
One more check to do before you start the actual installation: Ubuntu Server is available in 64- and 32-bit versions. To get the most out of your server hardware, make sure to use a 64-bit version of Ubuntu Server. For instance, 32-bit Ubuntu can't address more than 4 GB of RAM, and even if you use the special PAE (Physical Address Extension) version of the kernel that uses 36 bits instead of the default 32 bits to address memory, you can't go beyond 32 GB of RAM. When using 64 bits, however, you can address exabytes of memory, which is probably enough for the next couple of years. To use a 64-bit version of the operating system, your drivers must be available in 64-bit versions as well. In some cases, that may be a problem. So it's best to do some research and check if drivers for your hardware devices are available in 64-bit versions. If they are, use 64 bits; otherwise use 32 bits.

Note To understand what I'm covering in this book, it doesn't really matter whether you're using the 32-bit or 64-bit Ubuntu server version. Using a different version of the operating system doesn't change much the way in which you will work with Ubuntu Server.

Starting the Installation

This section describes how to perform the first steps of the installation:

1. Boot your server from the installation media and select the installation language that you want to use. For non-US-based installations, don't forget to use the F3 button from the initial installation window to set your local keyboard layout.
2. After you specify your local settings, the installation program shows the available network interface cards that it has found and asks you to select a primary network board, as shown in Figure 1-1. Because NIC teaming, which is needed to add both network cards to one interface, is not supported by the installation program, just select your first network board and press Enter.



```
[!!] Configure the network

Your system has multiple network interfaces. Choose the one to use as
the primary network interface during the installation. If possible,
the first connected network interface found has been selected.

Primary network interface:

 eth0: Intel Corporation 82545EM Gigabit Ethernet Controller (Copp
 eth1: Intel Corporation 82545EM Gigabit Ethernet Controller (Copp

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

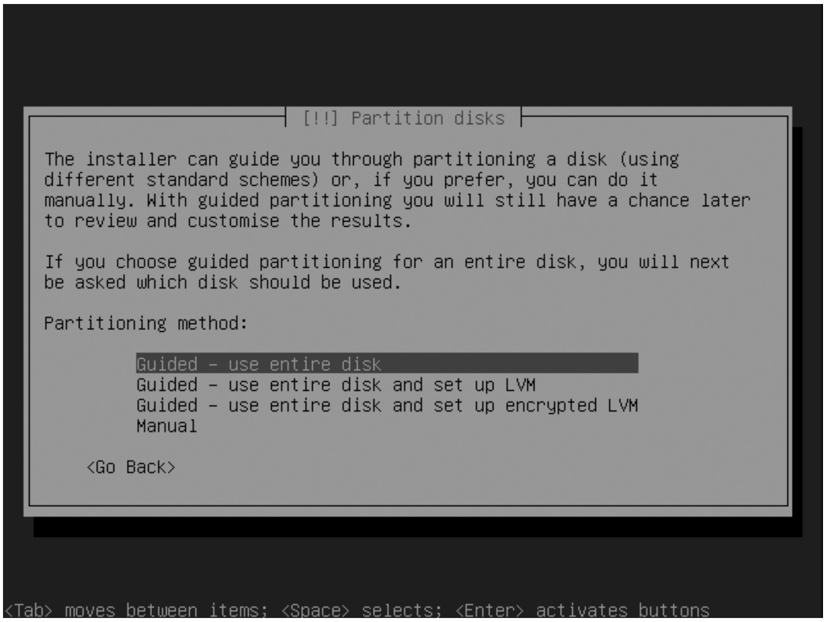
Figure 1-1. *NIC teaming is not supported by the installation program, so just press Enter to configure the first Ethernet interface using DHCP.*

3. Enter a name for your server. The default name Ubuntu is probably not sufficient, so make sure to specify something unique.
4. Select your time zone and wait for the Partition Disks interface to appear.

Creating a Software-Based RAID Solution

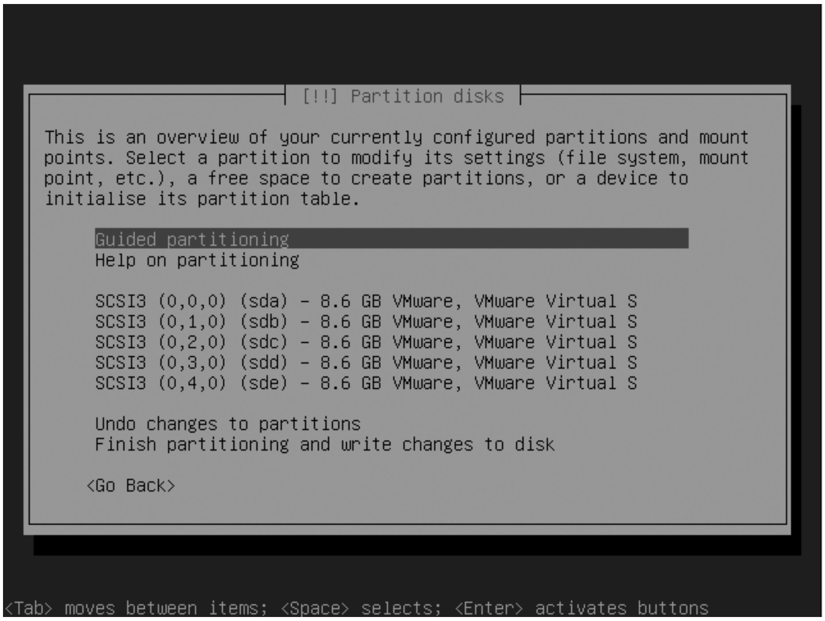
This section describes how to set up Ubuntu Server using software RAID to provide for maximal redundancy. Using software RAID helps you to get the best performance and redundancy if no hardware-based RAID solution is available.

1. From the Partition Disks screen, select Manual and press Enter. Make sure not to select the Guided option, which is the default, as you can see in Figure 1-2. You next see an overview of all disk devices that the installer has found, as shown in Figure 1-3.



```
[!!] Partition disks

The installer can guide you through partitioning a disk (using
different standard schemes) or, if you prefer, you can do it
manually. With guided partitioning you will still have a chance later
to review and customise the results.

If you choose guided partitioning for an entire disk, you will next
be asked which disk should be used.

Partitioning method:

         Guided - use entire disk
         Guided - use entire disk and set up LVM
         Guided - use entire disk and set up encrypted LVM
         Manual

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-2. *Don't press Enter to accept default values in this screen!*

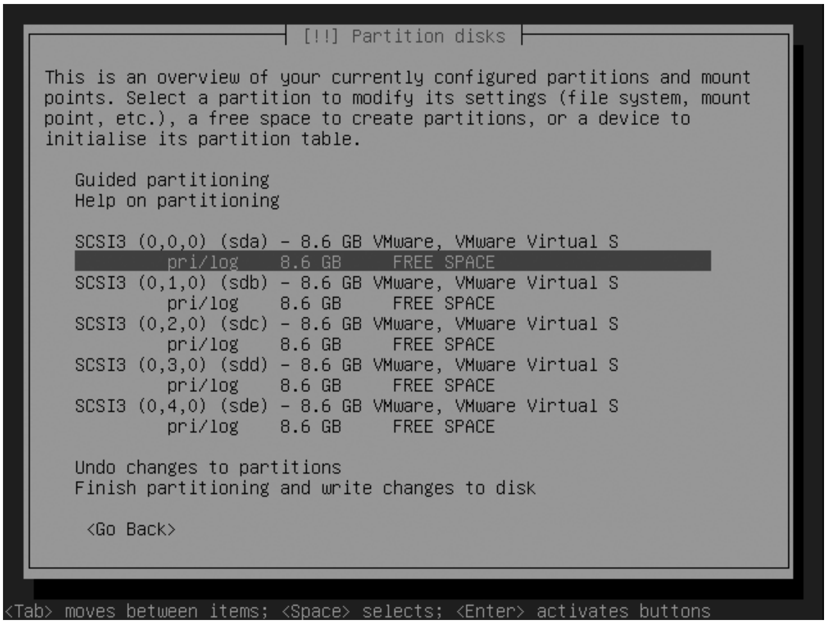


```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

     Guided partitioning
     Help on partitioning

     SCSI3 (0,0,0) (sda) - 8.6 GB VMware, VMware Virtual S
     SCSI3 (0,1,0) (sdb) - 8.6 GB VMware, VMware Virtual S
     SCSI3 (0,2,0) (sdc) - 8.6 GB VMware, VMware Virtual S
     SCSI3 (0,3,0) (sdd) - 8.6 GB VMware, VMware Virtual S
     SCSI3 (0,4,0) (sde) - 8.6 GB VMware, VMware Virtual S

     Undo changes to partitions
     Finish partitioning and write changes to disk

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-3. *The installer gives an overview of all available disk devices.*

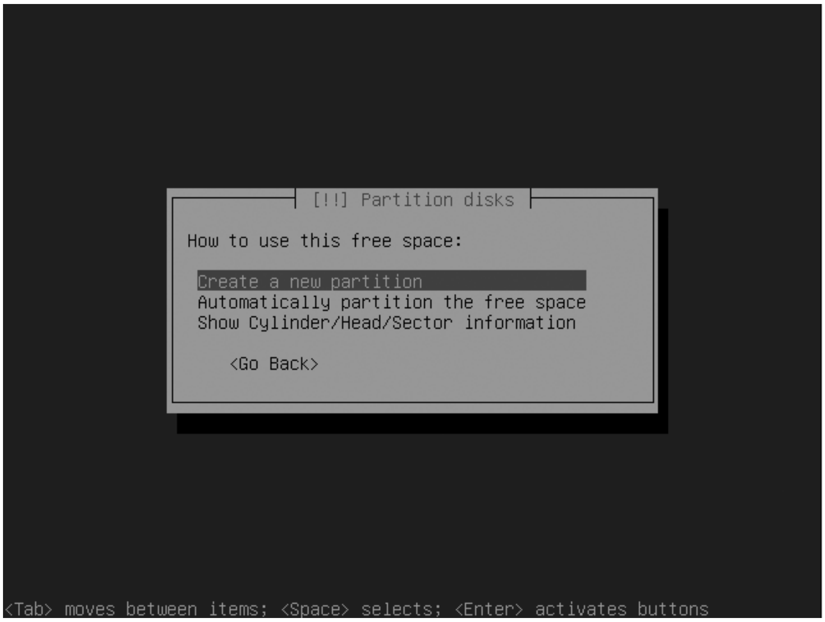
2. From the overview in Figure 1-3, you have to select all disk devices one by one (this means that you have to perform this step five times). If there is nothing on the disks yet, the installer will ask you if it needs to set up a partition table. Select Yes and then press Enter. After you have done this, the result will look similar to the screen shown in Figure 1-4.

3. Select the first FREE SPACE indicator that you see (as shown in Figure 1-4) and press Enter. You'll see the Partition Disks interface shown in Figure 1-5. From this interface, make sure to select Create a New Partition and then press Enter.



```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

   Guided partitioning
   Help on partitioning

   SCSI3 (0,0,0) (sda) - 8.6 GB VMware, VMware Virtual S
            pri/log    8.6 GB     FREE SPACE
   SCSI3 (0,1,0) (sdb) - 8.6 GB VMware, VMware Virtual S
            pri/log    8.6 GB     FREE SPACE
   SCSI3 (0,2,0) (sdc) - 8.6 GB VMware, VMware Virtual S
            pri/log    8.6 GB     FREE SPACE
   SCSI3 (0,3,0) (sdd) - 8.6 GB VMware, VMware Virtual S
            pri/log    8.6 GB     FREE SPACE
   SCSI3 (0,4,0) (sde) - 8.6 GB VMware, VMware Virtual S
            pri/log    8.6 GB     FREE SPACE

   Undo changes to partitions
   Finish partitioning and write changes to disk

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

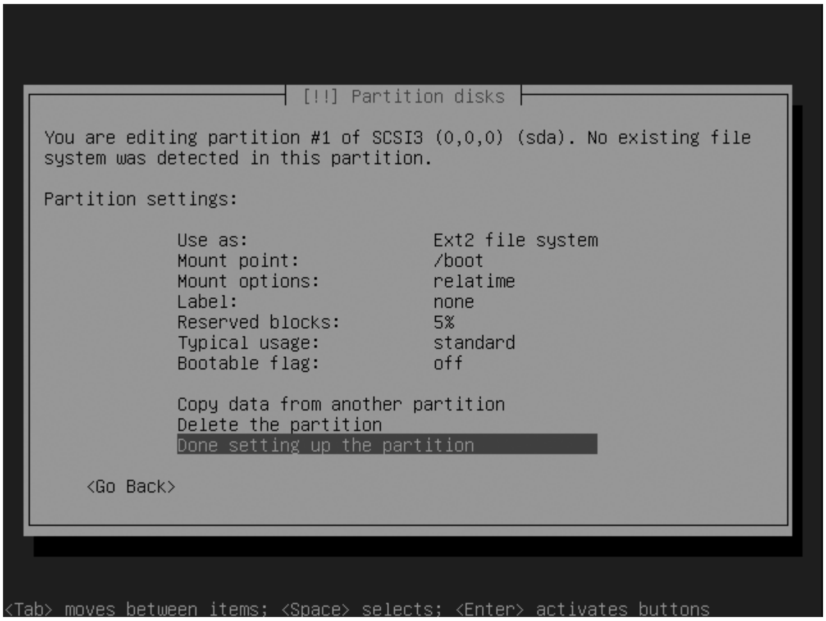
Figure 1-4. *Before proceeding, make sure that you see something similar to this.*



```
[!!] Partition disks

How to use this free space:

 Create a new partition
 Automatically partition the free space
 Show Cylinder/Head/Sector information

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-5. *Select Create a New Partition and press Enter to continue.*

4. The installer asks what partition size you want to use. On the first device only, create a 100 MB partition, which you will use later as the boot partition, and make it a primary partition that is at the beginning of your hard disks. Press Enter after specifying the partition size, which takes you to the screen shown in Figure 1-6. Make sure that you specify to use it as an Ext2 file system and use `/boot` as the mount point, as shown in Figure 1-6. For all other parameters, you can accept the default values on this partition. Next, select Done Setting Up the Partition and press Enter to proceed.



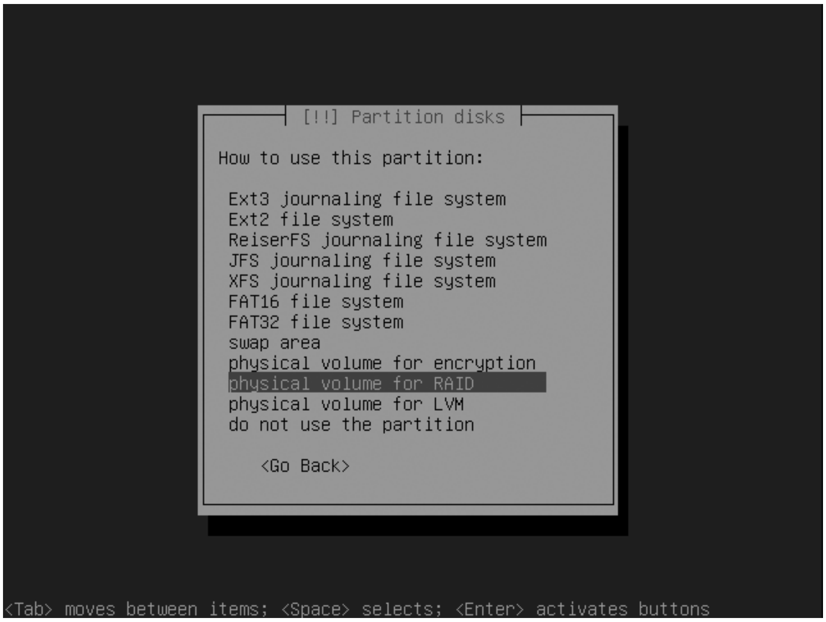
```
[!!] Partition disks

You are editing partition #1 of SCSI3 (0,0,0) (sda). No existing file
system was detected in this partition.

Partition settings:

            Use as:                  Ext2 file system
            Mount point:             /boot
            Mount options:           relatime
            Label:                   none
            Reserved blocks:         5%
            Typical usage:           standard
            Bootable flag:           off

            Copy data from another partition
            Delete the partition
            Done setting up the partition

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-6. *Make a boot partition first.*

5. Next, you must set up the free space that remains on the first hard disk as space to be used by the software RAID. To set up the partition, follow these steps:

   - Select the next FREE SPACE indicator and press Enter to proceed.
   - Select Create a New Partition and accept the partition size that the installation program suggests. Write this size down, because you'll need to use exactly the same size on the second hard disk.
   - Select Continue and press Enter to use this new partition. In the following screen, select it to be a primary partition.

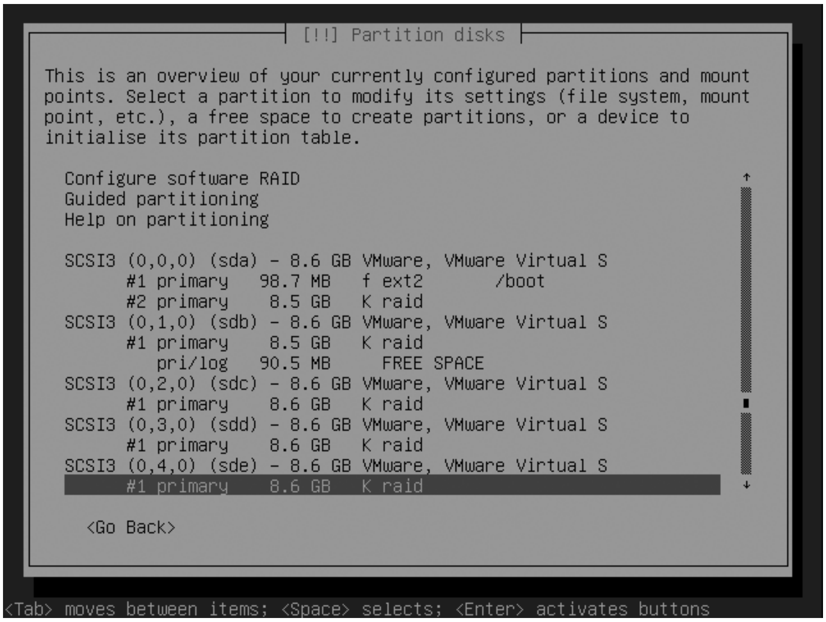
- In the partition settings window (the screen shown in Figure 1-6), specify that the partition should be used as a physical volume for RAID, as shown in Figure 1-7.
- Select Done Setting Up the Partition.



```
[!!] Partition disks

How to use this partition:

 Ext3 journaling file system
 Ext2 file system
 ReiserFS journaling file system
 JFS journaling file system
 XFS journaling file system
 FAT16 file system
 FAT32 file system
 swap area
 physical volume for encryption
 physical volume for RAID
 physical volume for LVM
 do not use the partition

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

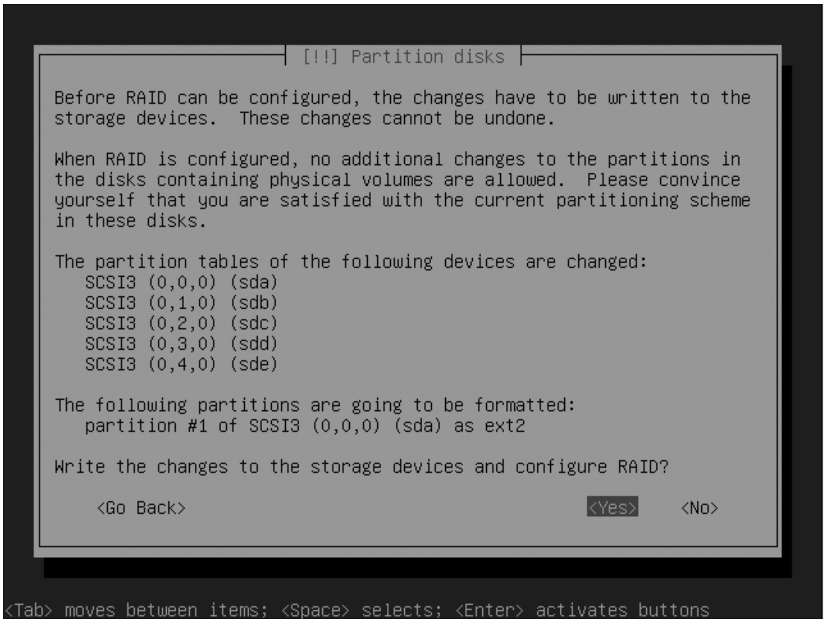
Figure 1-7. *Select Physical Volume for RAID to mark the partition as usable for software RAID.*

6. Repeat the procedure from Step 5 for the free space on all other disk devices. On the second disk, the partition should be the exact same size as the RAID partition on disk 1, and on all other disks, you can use all available disk space. When finished, the Partition Disks screen should look similar to Figure 1-8.
7. Still from within the partition overview, select the option Configure Software RAID (at the top of the screen) and press Enter. This takes you to the screen shown in Figure 1-9. Select Yes and press Enter to write the current partitioning to your hard disks.



```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

  Configure software RAID
  Guided partitioning
  Help on partitioning

  SCSI3 (0,0,0) (sda) - 8.6 GB VMware, VMware Virtual S
        #1 primary   98.7 MB   f ext2       /boot
        #2 primary    8.5 GB   K raid
  SCSI3 (0,1,0) (sdb) - 8.6 GB VMware, VMware Virtual S
        #1 primary    8.5 GB   K raid
           pri/log   90.5 MB     FREE SPACE
  SCSI3 (0,2,0) (sdc) - 8.6 GB VMware, VMware Virtual S
        #1 primary    8.6 GB   K raid
  SCSI3 (0,3,0) (sdd) - 8.6 GB VMware, VMware Virtual S
        #1 primary    8.6 GB   K raid
  SCSI3 (0,4,0) (sde) - 8.6 GB VMware, VMware Virtual S
        #1 primary    8.6 GB   K raid

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

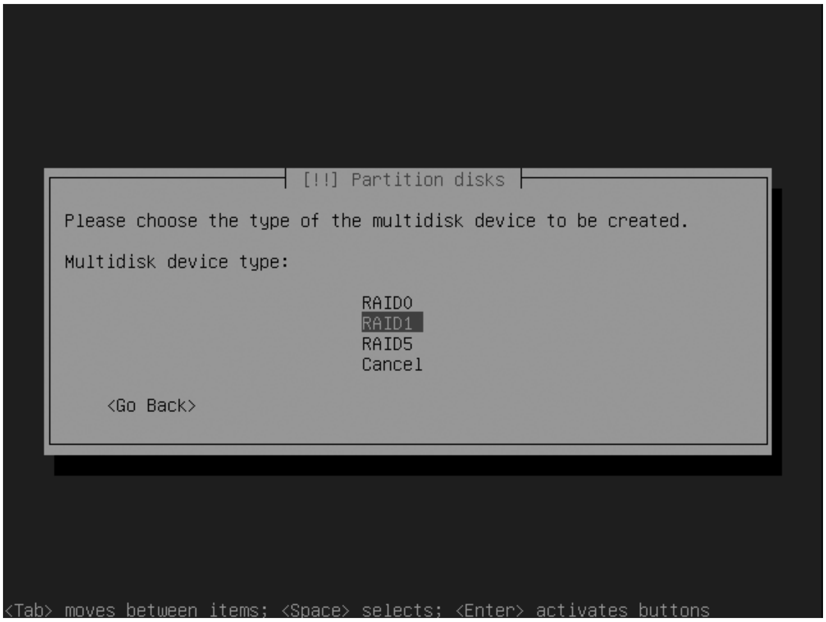
Figure 1-8. *After setting up all partitions that you want to use in your RAID setup, the Partition Disks screen looks like this.*



```
[!!] Partition disks

Before RAID can be configured, the changes have to be written to the
storage devices.  These changes cannot be undone.

When RAID is configured, no additional changes to the partitions in
the disks containing physical volumes are allowed.  Please convince
yourself that you are satisfied with the current partitioning scheme
in these disks.

The partition tables of the following devices are changed:
   SCSI3 (0,0,0) (sda)
   SCSI3 (0,1,0) (sdb)
   SCSI3 (0,2,0) (sdc)
   SCSI3 (0,3,0) (sdd)
   SCSI3 (0,4,0) (sde)

The following partitions are going to be formatted:
   partition #1 of SCSI3 (0,0,0) (sda) as ext2

Write the changes to the storage devices and configure RAID?

    <Go Back>                                       <Yes>    <No>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-9. *Before creating the RAID sets, you need to write the partitions to disk.*

8. The installer takes you automatically to the Multidisk utility, which helps you to create the RAID device. From there, select Create MD Device and press Enter. Next, select from the three different RAID types that are supported by this utility (see Figure 1-10): RAID 0, RAID 1, and RAID 5. In this procedure, I show you how to create a RAID 1 set and a RAID 5 set, so first select RAID1 and press Enter.



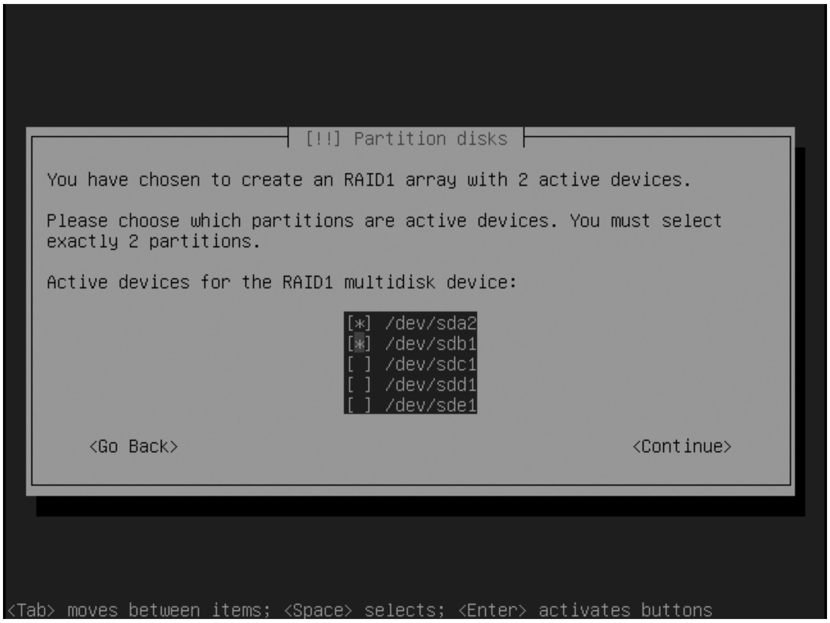
```
[!!] Partition disks

Please choose the type of the multidisk device to be created.

Multidisk device type:

                    RAID0
                    RAID1
                    RAID5
                    Cancel

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-10. *First make the RAID 1 array for your operating system files.*

9. Specify the number of active devices that you want to use. For a RAID 1 array, this would typically be two devices. This value is added by default, so select Continue and press Enter to proceed.

10. Indicate how many spare devices you want to use. In RAID 1, there are no spare devices, so accept the value of 0, which is entered by default, select Continue, and press Enter.

11. To complete the RAID setup, you need to add partitions that you've marked as RAID partitions to the RAID set. Make sure to select `/dev/sda2` and `/dev/sdb1` (see Figure 1-11) and then select Continue and press Enter.



```
[!!] Partition disks

You have chosen to create an RAID1 array with 2 active devices.

Please choose which partitions are active devices. You must select
exactly 2 partitions.

Active devices for the RAID1 multidisk device:

[*] /dev/sda2
[*] /dev/sdb1
[ ] /dev/sdc1
[ ] /dev/sdd1
[ ] /dev/sde1

<Go Back>                                          <Continue>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

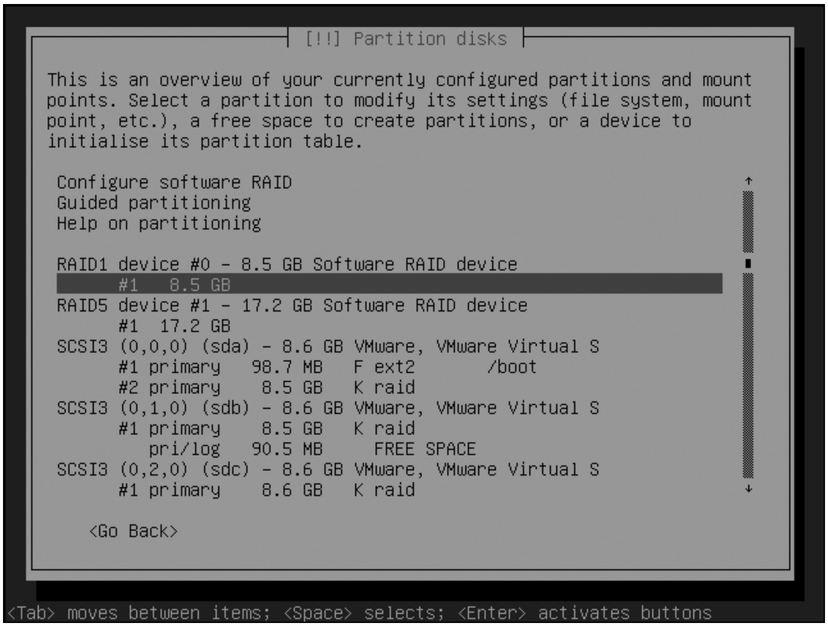
Figure 1-11. *After selecting the devices you want to add to your RAID set, select Continue and press Enter.*

12. Back in the main RAID configuration menu, select Create MD Device once more to create the RAID 5 array. Now add all remaining devices to this array. Make sure to use at least three devices as active devices. You don't need any spare devices here. When you're back in the main RAID menu, select Finish and press Enter to write the RAID configuration to your server's hard drive.

Creating LVM Logical Volumes on Top of a Software RAID Device

You now have set up two software RAID devices. These devices can be used just as you would use a normal hard disk as a storage device. On a hard disk, you can set up traditional partitions, as well as LVM logical volumes. For maximal flexibility, it is a good idea to use LVM logical volumes. In this section you'll learn how to set them up on top of the software RAID devices you've just created. This procedure also works if you are using hardware RAID; just use the hardware RAID device names instead of the `md` device names used in this procedure.

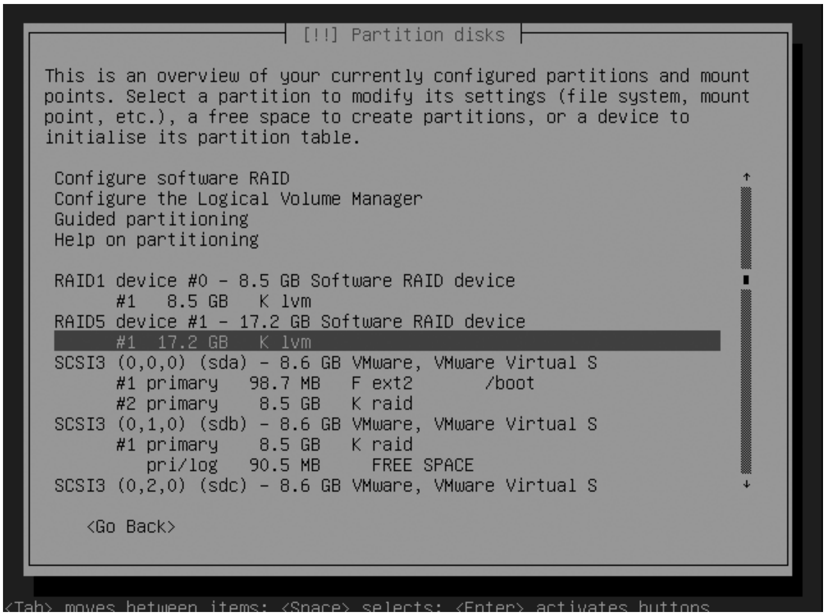
1. You now automatically return to the main menu, in which you'll see the two RAID devices that you've just created. Select the size specification that you see directly under the RAID 0 device (see Figure 1-12) and press Enter to start putting some logical volumes on the device. This brings you to the Partition Disks interface that you've seen a couple of times before.



```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

 Configure software RAID
 Guided partitioning
 Help on partitioning

 RAID1 device #0 - 8.5 GB Software RAID device
       #1   8.5 GB
 RAID5 device #1 - 17.2 GB Software RAID device
       #1  17.2 GB
 SCSI3 (0,0,0) (sda) - 8.6 GB VMware, VMware Virtual S
       #1 primary   98.7 MB   F ext2       /boot
       #2 primary    8.5 GB   K raid
 SCSI3 (0,1,0) (sdb) - 8.6 GB VMware, VMware Virtual S
       #1 primary    8.5 GB   K raid
          pri/log   90.5 MB     FREE SPACE
 SCSI3 (0,2,0) (sdc) - 8.6 GB VMware, VMware Virtual S
       #1 primary    8.6 GB   K raid

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-12. *Select the size specification under the name of the RAID device and press Enter.*

2. From the Partition Disks interface, change the status of the partition from Do Not Use to Use As. Press Enter, select Physical Volume for LVM, and press Enter again. Next, back in the Partition Disks interface, select Done Setting Up the Partition and press Enter. In the Partition Disks interface, the first RAID device now is marked as an LVM volume. Repeat Steps 1 and 2 from this procedure to make the other RAID device usable as an LVM device as well. After doing this successfully, the Partition Disks interface should look similar to Figure 1-13.

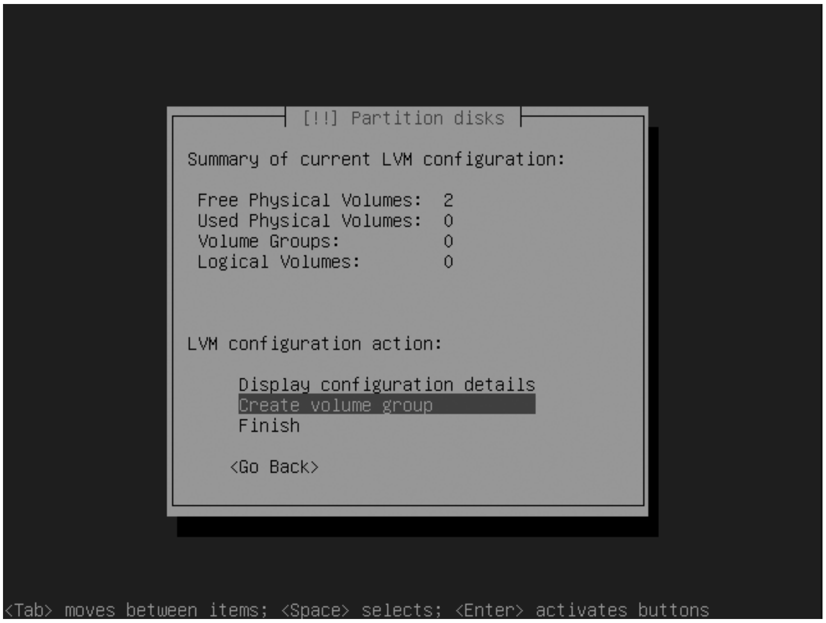


```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

 Configure software RAID
 Configure the Logical Volume Manager
 Guided partitioning
 Help on partitioning

 RAID1 device #0 - 8.5 GB Software RAID device
       #1   8.5 GB   K lvm
 RAID5 device #1 - 17.2 GB Software RAID device
       #1  17.2 GB   K lvm
 SCSI3 (0,0,0) (sda) - 8.6 GB VMware, VMware Virtual S
       #1 primary   98.7 MB   F ext2       /boot
       #2 primary    8.5 GB   K raid
 SCSI3 (0,1,0) (sdb) - 8.6 GB VMware, VMware Virtual S
       #1 primary    8.5 GB   K raid
          pri/log   90.5 MB     FREE SPACE
 SCSI3 (0,2,0) (sdc) - 8.6 GB VMware, VMware Virtual S

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-13. *Before you start to create logical volumes, your Partition Disks interface should look similar to this.*

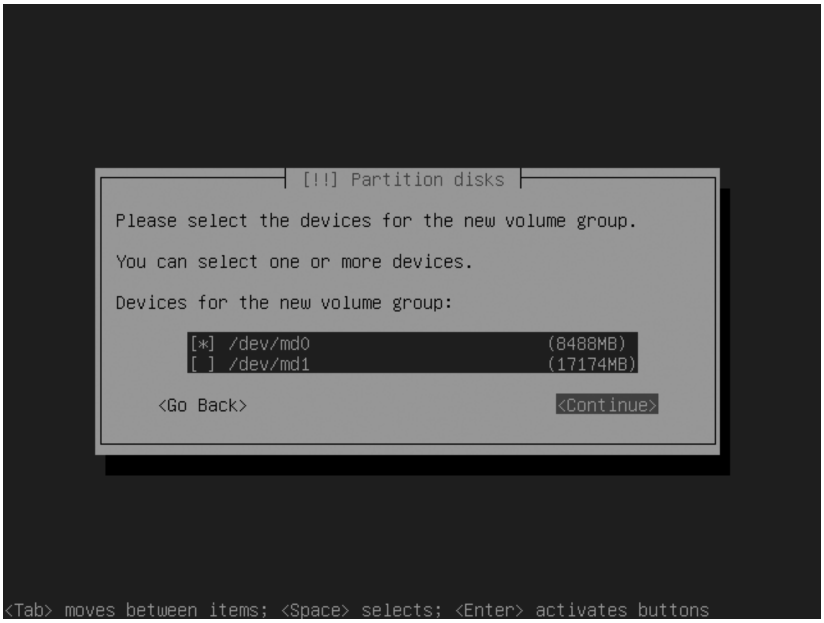
3. Now that you've marked the RAID devices as physical volumes in the LVM setup, which makes them usable for LVM, it's time to put some LVM logical volumes on them. First, you are going to create a volume for the root directory and swap space and put that on the RAID 1 array. Next, you'll create the `/var`, `/home`, and `/tmp` volumes on the RAID 5 array. Start by selecting the option Configure the Logical Volume Manager (press Enter). In the overview, you should see that there currently are two physical volumes available. To create the highest possible redundancy, you'll create a volume group on each of them. To do this, from the interface that you see in Figure 1-14, select Create Volume Group and press Enter.

4. Next, you need to provide a name for the volume group you want to create. In this scenario, because you want a volume group that is clearly divided between two physical storage devices, I suggest using the names of these devices for the volume group name. So make a volume group with the name RAID1 on the RAID 1 array, and next create a volume group with the name RAID5 on the RAID 5 array. After creating the volume group, select the storage device on which you want to create it. As you can see in Figure 1-15, the names of these storage devices will be `/dev/md0` for the first RAID device and `/dev/md1` for the second RAID device. Make sure that you create both volume groups now. After selecting the storage device, select Continue and press Enter.



```
[!!] Partition disks

Summary of current LVM configuration:

 Free Physical Volumes:  2
 Used Physical Volumes:  0
 Volume Groups:          0
 Logical Volumes:        0


LVM configuration action:

     Display configuration details
     Create volume group
     Finish

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

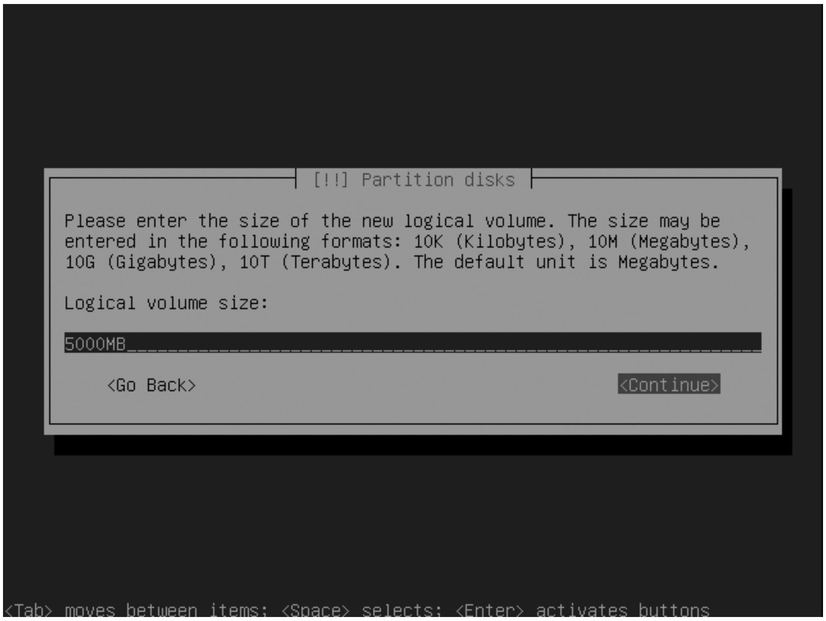
Figure 1-14. *Select Create Volume Group and press Enter to start defining a volume group.*



```
[!!] Partition disks

Please select the devices for the new volume group.

You can select one or more devices.

Devices for the new volume group:

       [*] /dev/md0                        (8488MB)
       [ ] /dev/md1                        (17174MB)

    <Go Back>                                      <Continue>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-15. *Select the storage device that you want to use, select Continue, and press Enter.*

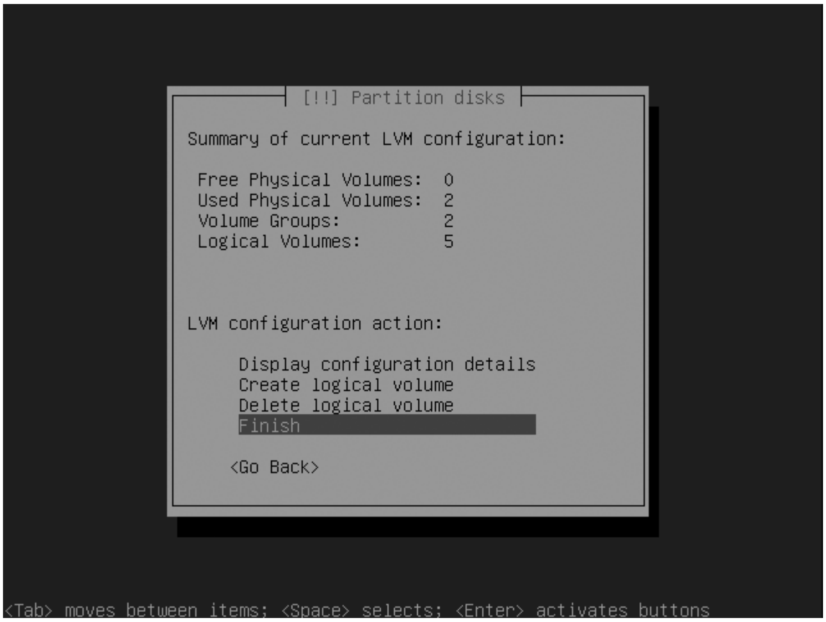
5. You next see a summary of the current LVM configuration that shows that you now have two volume groups. From here, select Create Logical Volume and press Enter. This gives you a list of all existing volume groups. Select the volume group in which you want to create the logical volume and press Enter. Next, enter the name of the logical volume you want to create and proceed to the next step. In the following screen (see Figure 1-16), enter the size of the logical volume that you want to create. It's a good idea never to use all available disk space; this gives you the flexibility to resize easily at a later stage. Select Continue and press Enter.



```
[!!] Partition disks

Please enter the size of the new logical volume. The size may be
entered in the following formats: 10K (Kilobytes), 10M (Megabytes),
10G (Gigabytes), 10T (Terabytes). The default unit is Megabytes.

Logical volume size:

5000MB

<Go Back>                                        <Continue>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-16. *It's a good idea never to use all available disk space when creating the logical volumes.*

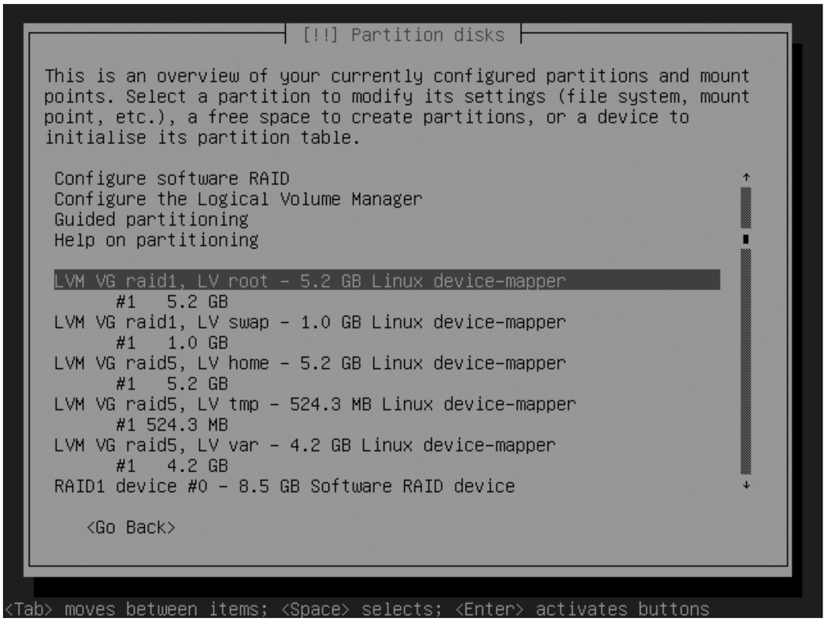
6. Create all the logical volumes that you want to use, and then, from the LVM summary screen, shown in Figure 1-17, select Finish and press Enter to complete this procedure.



```
[!!] Partition disks

Summary of current LVM configuration:

 Free Physical Volumes:  0
 Used Physical Volumes:  2
 Volume Groups:          2
 Logical Volumes:        5


LVM configuration action:

     Display configuration details
     Create logical volume
     Delete logical volume
     Finish

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-17. *After creating all logical volumes, select Finish and press Enter to complete this part of the procedure.*

7. At this stage you have created the RAID devices, put some logical volume groups on top of them, and created the logical volumes that you want to use in the volume groups. Now it's time for the final step: you need to put some file systems in the volume groups. To do this, select the logical volumes one by one. From the Partition Disks interface, which should now look similar to that shown in Figure 1-18, you can easily recognize by their names the logical volumes that you've created. Select the line that marks the free space that is available on the logical volumes and press Enter.



```
[!!] Partition disks

This is an overview of your currently configured partitions and mount
points. Select a partition to modify its settings (file system, mount
point, etc.), a free space to create partitions, or a device to
initialise its partition table.

 Configure software RAID
 Configure the Logical Volume Manager
 Guided partitioning
 Help on partitioning

 LVM VG raid1, LV root - 5.2 GB Linux device-mapper
       #1   5.2 GB
 LVM VG raid1, LV swap - 1.0 GB Linux device-mapper
       #1   1.0 GB
 LVM VG raid5, LV home - 5.2 GB Linux device-mapper
       #1   5.2 GB
 LVM VG raid5, LV tmp - 524.3 MB Linux device-mapper
       #1 524.3 MB
 LVM VG raid5, LV var - 4.2 GB Linux device-mapper
       #1   4.2 GB
 RAID1 device #0 - 8.5 GB Software RAID device

    <Go Back>

<Tab> moves between items; <Space> selects; <Enter> activates buttons
```

Figure 1-18. *Select the logical volumes one by one to put a file system on them.*

8. You are now back in the overview screen of the Partition Disks interface. Select the line labeled Use As and select the file system that you want to use on this logical volume. After selecting the file system, select the directory on which you want to mount this volume. Next, select Done Setting Up the Partition and press Enter. Repeat this procedure for all the logical volumes that you have created. After doing this for all your volumes, select Finish Partitioning and Write Changes to Disk and press Enter. This writes the disk layout that you have created and brings you to the next step of the procedure.

Completing the Installation

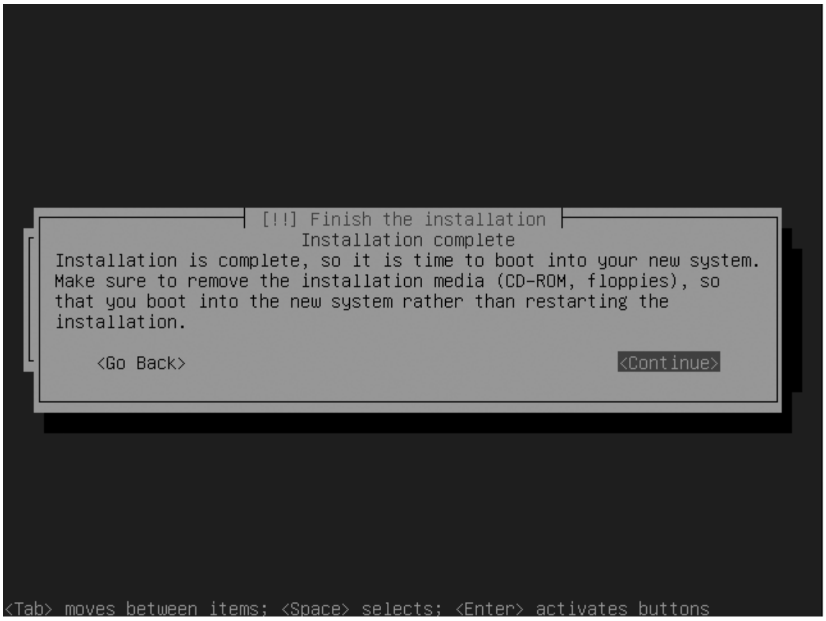
Now that you've set up the disk layout for your server, it's time to complete the installation. This includes creating a user account, specifying where to authenticate, and selecting one or more of the predefined installation patterns for your server. In the following procedure, you'll learn what's involved.

1. As you probably already know, on Ubuntu server you won't work as root by default. Therefore, you are asked to enter the name of a new user that will be created now. Your first name might not be a bad idea. Make sure this user has a password as well.

2. If you need a proxy server to access the outside world, you can now enter the data that you need to access this proxy server. You'll do this as a URL. This can be a basic URL, but you can also use a URL that includes a username and password. For instance, if a special user account `server1` with the password `p@ssw0rd` needs to authenticate on the proxy with the name `squid` that is reachable on port 3120, the line you enter looks as follows: `http://server1:p@ssw0rd@squid:3120`. If no proxy information is needed, you can just skip this field and leave it empty.

3. Select one or more of the installation patterns available for your server. This is not an important option, because later you can install all software packages that are offered here. Now the software is copied to your server. Wait until this is completed.

4. You'll now see a message that installation is complete (see Figure 1-19). Press Enter now to boot into your new system.



```
[!!] Finish the installation
Installation complete
Installation is complete, so it is time to boot into your new system.
Make sure to remove the installation media (CD-ROM, floppies), so
that you boot into the new system rather than restarting the
installation.
```

<Go Back>
<Continue>
<Tab> moves between items; <Space> selects; <Enter> activates buttons

Figure 1-19. *When the installation is completed, press Enter to start the installed system.*

Post-Installation Tasks

Now you have a functional instance of Ubuntu Server, but you are not done yet. In a typical data-center installation, a server has several network cards, and to get the best out of these network cards, you probably want to set up NIC bonding. You can configure NIC bonding in two ways: for maximal redundancy or for maximal performance. Your enterprise server might connect to a SAN as well. This adds another requirement: setting up multipathing. The next two subsections describe how to accomplish these tasks.

Setting Up NIC Bonding

Even the most basic network servers have at least two network cards. Some models even have four or more network cards installed. This not only is useful if you want to connect your server to several networks at the same time, but also enables you to use these multiple network cards to increase performance or redundancy. The technique used for this is referred to as NIC bonding. Using NIC bonding, you'll get a new network device that typically has the name `bond0`. Two of the physical network cards are assigned to this device. When setting up NIC bonding, you don't work with `eth0` and `eth1` anymore; you'll work with `bond0` instead. That means that the IP address is assigned to `bond0` as well.

Speaking in a general way, there are two directions you can go with NIC bonding. When setting it up for redundancy, one of the network cards will be active, whereas the other one is on standby and takes over only if the active network card fails. This enables fault tolerance on the network connection for your server. In the high-performance setup, the two network cards in the bonding configuration work together to handle the work load. That means that you could create a 2 gigabit connection by bundling two 1-GB network cards. In the following procedure you'll learn how to set up NIC bonding on Ubuntu Server. But before starting, take a look at the list in Table 1-2 of the different options that you can use when setting up NIC bonding.

Table 1-2. *NIC Bonding Options*

Option	Use
mode=0	This mode enables round robin. That means that packets are transmitted in a sequential way between the network cards assigned to the bonding interface. Using this option, you'll get load balancing and redundancy at the same time.
mode=1	Using this mode, one of the network cards is active and the other one is passive. This enables fault tolerance, but doesn't do anything for performance.
mode=2	This mode divides the work load between associated devices, but tries to keep a network connection on the same device as long as possible. This offers redundancy and load balancing at the same time, but with better performance than mode=0. Because this mode is also more complicated, it is more prone to errors.

Option	Use
mode=3	Use this mode to transmit everything on all slave devices. This mode provides fault tolerance with the shortest recovery times; a network card may fail without service interruption.
mode=4	This mode uses IEEE 802.3ad Dynamic Link aggregation. This bundles the network cards in the bond device as one, but does require additional configuration on the network switch.
mode=5	This mode distributes connections to the network cards in the bond device, thus enabling load balancing. One of the major advantages to using this mode is that no additional configuration is needed on the switch.

The following procedure explains how to set up NIC bonding:

1. Create a module alias in `/etc/modprobe.d/aliases`, as shown in Listing 1-1. This makes sure that the right kernel module is loaded when setting up NIC bonding. Make sure that the names of the kernel devices for `eth0` and `eth1` in Listing 1-1 are replaced by the real module names needed for your network cards. Because the bonding is between `eth0` and `eth1` only, you don't need to do anything on `eth2` here.

   Listing 1-1. *Use /etc/modprobe.d/aliases to Load the Correct Kernel Modules*

   ```
   # Append to the bottom of this file:
   alias bond0 bonding
   alias eth0 e100
   alias eth1 e100
   alias eth2 e100
   options bonding mode=0 miimon=100
   ```

2. In the `/etc/modprobe/arch` directory, you'll find a file that contains settings for your specific kernel architecture (see Listing 1-2). For instance, if you are using an i386 kernel, the name of this file is `i386`. Edit this file to make sure the right options are loaded for the bonding device.

   Listing 1-2. *Edit the /etc/modprobe.d/arch/i386 File to Contain the Correct Bonding Options*

   ```
   # Append to the bottom of this file:
   alias bond0 bonding
   options bonding mode=0 miimon=100 downdelay=20
   ```

3. As a final step, you need to modify the file that contains the configuration of your network card to create the `bond0` device and its properties. The example in Listing 1-3 shows what the content of this file looks like. Make sure to change server-specific settings, such as the MAC address used in this example.

Listing 1-3. *Change the /etc/network/interfaces File to Create the bond0 Device*

```
# 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 192.168.0.120
#   netmask 255.255.255.0
#   network 192.168.0.0
#   broadcast 192.168.0.255
#   gateway 192.168.0.1
auto bond0
iface bond0 inet static
   address 192.168.0.120
   netmask 255.255.255.0
   network 192.168.0.0
   broadcast 192.168.0.255
   gateway 192.168.0.1
   hwaddress ether 00:03:B3:48:50:2C
   post-up ifenslave bond0 eth0 eth1
```

This completes all required steps to set up the bond device. To verify that it comes up when your server starts, reboot your server now.

Setting Up Multipathing

A server in a data center is often connected to a SAN. To avoid having a single point of failure, your server probably has two HBAs that are configured to two storage networks that connect to the same SAN filer (see Figure 1-20).

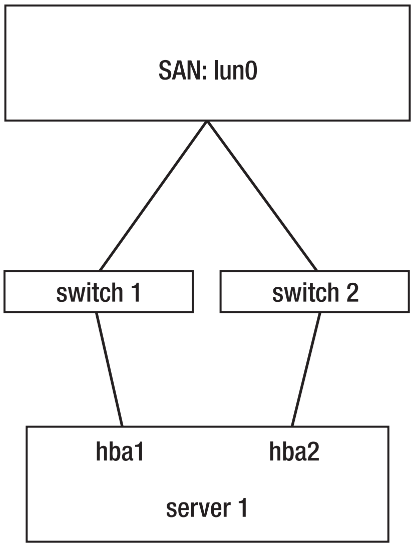


Figure 1-20. *Schematic overview of the way a server is connected to the SAN*

This redundancy in the SAN connections would normally produce an undesirable result. Let's say your server has a redundant connection to one LUN on the SAN. For the moment, consider the SAN to be a black box, and a LUN to be some storage that is allocated in a logical group on the SAN, like you would use partitions on hard drives. See Chapter 7 for information on how to set up a SAN yourself. The server would see this LUN via both the first and second HBAs. Because the server normally has no way of determining that it sees the same device on the SAN twice, it would offer you two new devices—for instance, `/dev/sdb` coming via the first HBA, and `/dev/sdc` coming via the second HBA. As you can imagine, this is not good.

To prevent your server from getting two devices, you should use multipathing. Initializing this is not too difficult; Ubuntu Server has a multipath daemon that you can load to take care of this problem. Installing it is a really simple two-step procedure:

1. Use `apt-get install multipath-tools` to install the multipath software.
2. Enter the command `/etc/init.d/multipathd` to start the multipath process, and make sure that it automatically restarts when you restart your server.

The result of these two steps is that you'll get a new storage device. Use `fdisk -l` or `lsscsi` to find out what the name of this device is. This storage device refers to your SAN. When setting up partitions on the SAN, make sure to use this storage device, and not the `/dev/sdb` and `/dev/sdc` devices that your server also still gives you.

Summary

In this chapter you have learned how to set up a server in an enterprise environment. As you have seen, this is quite a different approach from setting up a server in a small business or home environment. You have learned how to set up software RAID, and LVM logical volumes on top of that. You have also learned how to set up NIC bonding for optimized network card performance, and multipath for optimized SAN connections. In the next chapter you'll learn how to set up Ubuntu Server for workstation imaging.