

Creating Partitions and Mounting Drives in a Linux Server

This article will walk you through partitioning, formatting, and mounting a hard drive in a Linux server. This article only applies to hard drives which contain partitions under 2TB.

This article also assumes that you are accessing the server as the root user, or can become the root user via `su` or by using `sudo`

Once a new hard drive has been installed to your server, the Linux kernel assigns a device file to the drive. This is typically `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and so on. As a preliminary step, check that the disk is visible in your system (the following is output from a system with only one physical disk drive):

```
ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2
```

`/dev/sda1` and `/dev/sda2` represent partitions on the `/dev/sda` drive

Here is an example showing what the output may look like after a new disk is installed (`/dev/sdb` is the new disk):

```
ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb
```

Using `fdisk` to Create Partitions

If you are not logged in as root:

```
su -
```

```
fdisk /dev/sdb
```

(be sure to replace this with `/dev/sdX`, where `X` is the new drive!)

As instructed by the utility, switch off DOS compatible mode and change the units to sectors by entering the `c` and `u` options within the utility.

Use the `n` option for new partition and `p` for primary partition

Specify the partition number and where the partition will begin and end. The disk's first partition needs to begin at the first available sector. In this example, we will span the partition across the whole disk. If you want multiple partitions, you can specify the size and the sector for the partition to begin and end on.

```
Partition number (1-4): 1  
First sector (2048-67108863, default 2048):  
Using default value 2048  
Last sector, +sectors or +size{K,M,G} (2048-67108863, default 67108863):  
Using default value 67108863
```

Next, write changes to the disk using the `w` command in the `fdisk` utility.

If we now look at the devices again we will see that the new partition is visible as `/dev/sdb1`:

```
ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb /dev/sdb1
```

Now that the disk is partitioned, a filesystem has to be created in order for the kernel to properly read/write data from the drive. We will use the `mkfs.ext4` utility which takes as arguments the label and the partition.

In our example, we will create an ext4 filesystem on `/dev/sdb1`, and give it a label of 'newdir'

```
/sbin/mkfs.ext4 -L newdir /dev/sdb1
```

Finally, now that our partition has a filesystem, the filesystem needs to be mounted to a directory so it can be accessed. To do this, a mount point needs to be created.

For example:

```
mkdir /newdir
```

```
mount /dev/sdb1 /newdir
```

Now the filesystem on the first partition on the new disk is ready to be accessed by your OS!

Our awesome support team is always standing by to offer assistance with creating partitions, formatting them, and mounting them. Don't hesitate to reach out to us via chat, email, or phone if you need help!