Linux Academy

Hands-on Lab

Using Partition Tools

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- The IP address of your server is located on the Live! Lab page
- Username: linuxacademy
- Password: 123456
- Root Password: 123456

Related Courses

LPIC-1 Systems Administrator - Exam 1

Related Videos

Using Partitioning Tools

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

In this lab, we use a CentOS 6 server to create three partition tables using `fdisk`, `gdisk`, and `parted`. Three unformatted disks are provided with the Hands-on Lab for this.

Log in to the server using the credentials provided on the Hands-on Lab page. This lab guide is written for the *root* user (`sudo su -`) but can also be done as the *linuxacademy* user with the `sudo` prefix.

Available Disks

Prior to this, the disks are prepared by being physically added or allocated to the server (if working with virtual machines). The lab picks up after this process, leaving us to partition the disks.

As noted in earlier lessons, our disks are treated as files, so we can change into the `/dev` directory to see our available devices: `/dev/xvdf`, `/dev/xvdg`, `/dev/xvdj`:

```
[root@LinuxAcademy] cd /dev
[root@LinuxAcademy] ls -l xvd*
brw-rw---- 1 root disk 202,  64 May  9 12:39 xvde
brw-rw---- 1 root disk 202,  65 May  9 12:39 xvde1
brw-rw---- 1 root disk 202,  66 May  9 12:39 xvde2
brw-rw---- 1 root disk 202,  80 May  9 12:42 xvdf
brw-rw---- 1 root disk 202,  96 May  9 12:42 xvdg
brw-rw---- 1 root disk 202, 144 May  9 12:42 xvdj
```

Traditionally, SATA and SCSI drives have the prefix `sd`, while IDE hard drives used `hd`. Master boot record (MBR) systems and those under 2 terabytes can have up to four primary partitions, but no more. As such, one of these partitions can be used as an extended partition that allows numerous logical partitions contained within it.

However, larger drives and distributions that can support them are now commonplace. The GUID Partition Table (GPT) allows for more than four partitions -- up to 128 -- and requires a different partition tool. This eliminates the need for logical partitions, although it can still be used in GPT.

Our disks -- `/dev/xvdf`, `/dev/xvdg`, and `/dev/xvdj` -- make use of `xvd` prefix, which is for Xen virtual disks. We will partition each of these using a different partition tool, `fdisk`, `gdisk`, or `parted`.

Partitioning Disks

fdisk

`fdisk` is the oldest of the partitioning tools, limited to only four primary partitions with drives up to 2TB in size.

We start with our `/dev/xvdj` disk. This is a completely blank disk, with no partitions.

```
[root@LinuxAcademy] fdisk /dev/xvdj
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0x0a228e12.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by
w(rite)
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').
Command (m for help):
```

Input p to see information related to the disk:

```
Command (m for help): p
Disk /dev/xvdj: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0a228e12
    Device Boot      Start         End      Blocks   Id  System
```

Use m for a list of commands:

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

We now want to begin partitioning the disk. To speed up the formatting process, we are going to keep our partition sizes small, but you are welcome to use different sizes and even experiment with sizes to see the difference in formatting time. Input n to create a new partition:

```
Command (m for help): n
```

We are asked if we want an extended or primary partition. Choose `p` for primary, and set the partition number to `1`:

```
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
```

Set the first cylinder to the default value, `1`:

```
First cylinder (1-2610, default 1): 1
```

We can set the last cylinder by indicating the cylinder number, or using `+[cylinder]` or `+[size in K, M, or G]`. We want a 2GB parition:

```
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610): +2G
```

Print to see the new partition:

```
Command (m for help): p

Disk /dev/xvdj: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0a228e12

    Device Boot      Start         End      Blocks   Id  System
/dev/xvdj1               1         262     2104483+  83  Linux
```

Create a second partition:

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (263-2610, default 263):
Using default value 263
Last cylinder, +cylinders or +size{K,M,G} (263-2610, default 2610): +2G
```

Notice that the default first cylinder is now `263`, knowing that the first 262 cylinders are being used by the first partition.

Print the partition table:

```
Command (m for help): p

Disk /dev/xvdj: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0a228e12

   Device Boot      Start         End      Blocks   Id  System
/dev/xvdj1               1         262     2104483+  83  Linux
/dev/xvdj2             263         524     2104515   83  Linux
```

Notice that our partitions have an `Id` of `83`. This defines the file system type, with `83` being `Linux`. Use `l` to see a list of file system types:

```
Command (m for help): l
```

To change this, use the `t` option:

```
Command (m for help): t
```

We want to change our second partition:

```
Partition number (1-4): 2
```

Set it to be a Linux swap partition, which has an ID of `82`:

```
Hex code (type L to list codes): 82
```

View the differences:

```
Command (m for help): p

Disk /dev/xvdj: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0a228e12

   Device Boot      Start         End      Blocks   Id  System
/dev/xvdj1               1         262     2104483+  83  Linux
/dev/xvdj2             263         524     2104515   82  Linux swap /
Solaris
```

However, despite doing all this, it is not yet written to the partition table. The table is staged at startup but only writes when told to do so. To write the changes to the disk, use `w`:

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Confirm the changes took place:

```
[root@LinuxAcademy] fdisk -l /dev/xvdj
```

gdisk

Install `gdisk`:

```
[root@LinuxAcademy] yum install gdisk
```

`gdisk` works the same as `fdisk` -- they share the same command flags and prompt options. This means, to partition `/dev/xvdg`, use the same process as above:

```
[root@LinuxAcademy] gdisk /dev/xvdg
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-41943006, default = 2048) or {+-}size{KMGTP}: +2GB
Last sector (4196352-41943006, default = 41943006) or {+-}size{KMGTP}:
+2GB
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8300
Changed type of partition to 'Linux filesystem'
```

This creates a GPT Partition Table. Note that it allows up to 128 partitions, terabytes and petabytes can be used when defining sectors (or "cylinders"), and we set the file system type at creation. We started with our first partition about 2GB in and set to be 2GB in size. The file system number, `8300`, is expanded because there is a great selection of file system types for use with `gdisk`.

View the partition:

```
Command (? for help): p

Disk /dev/xvdg: 41943040 sectors, 20.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 592B5CFA-D470-4B9C-920B-C2B691AD7184
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 41943006
Partitions will be aligned on 2048-sector boundaries
Total free space is 37748669 sectors (18.0 GiB)
```

```
Number  Start (sector)    End (sector)  Size       Code  Name
   1         4196352         8390655   2.0 GiB     8300  Linux
filesystem
```

Create a second partition:

```
Command (? for help): n
Partition number (2-128, default 2): 2
First sector (34-41943006, default = 8390656) or {+-}size{KMGTP}:
Last sector (8390656-41943006, default = 41943006) or {+-}size{KMGTP}:
+2G
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8200
Changed type of partition to 'Linux swap'
```

This partition is made at the default first section using a Linux swap file system type.

Write to disk:

```
Command (? for help): w
Final checks complete. About to write GPT data. THIS WILL OVERWRITE
EXISTING
PARTITIONS!!
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdg.
The operation has completed successfully.
```

Confirm the changes:

```
[root@LinuxAcademy] gdisk -l /dev/xvdg
```

parted

`parted` is able to create both MBR and GPT partition tables, and allows for the resizing for partitions. Unlike `fdisk` and `gdisk`, its command prompt takes full-word commands, not single letters.

Use `parted` on `/dev/xvdf`:

```
[root@LinuxAcademy] parted /dev/xvdf
```

We now want to create an MBR device, or "label":

```
(parted) mklabel msdos
```

Create a partition:

```
(parted) mkpart primary 0 2GB
Warning: The resulting partition is not properly aligned for best
performance.
Ignore/Cancel? Ignore
```

This command includes the same information has the prompted in `fdisk` and `gdisk` -- we are making a `primary` partition, starting at the `0` sector and continuing for 2GB. The output errors references that we are using 2Gb instead of the preferred 2024M or an ending sector. We do not need to define a file system type.

Press `q` to quit; `parted` writes the partition table when quitting.

Confirm the changes:

```
[root@LinuxAcademy] fdisk -l /dev/xvdf

Disk /dev/xvdf: 21.5 GB, 21474836480 bytes
39 heads, 62 sectors/track, 17346 cylinders
Units = cylinders of 2418 * 512 = 1238016 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000967cc

    Device Boot      Start         End      Blocks   Id  System
/dev/xvdf1               1        1616     1953125   83  Linux
```

Review

We have now successfully learned to use `fdisk`, `gdisk`, and `parted` to create partitions. After this, we may want to mount the partitions to the `/etc/fstab` file and set permissions. This lab is now complete.