3.Configure local storage

**List, create, delete partitions on MBR and GPT disks**

A **disk** can be used as a simple entity or broken up into one or more **partitions**.

**Disks** are generally called **/dev/sda**, **/dev/sdb**, etc, in **physical** servers (**s** for **scsi**even though they’ve got **IDE**, **SATA** or **SAS** interfaces) and **/dev/vda**, **/dev/vdb**, etc, in **virtual** machines.

**Partitions** get their names from the **disk** name itself and add a number starting at **1** (**/dev/sda1**, **/dev/sda2**, etc or **/dev/vda1**, **/dev/vda2**, etc).

A **partition table** is a special structure containing partitions organization.

Not recent disks use **512-byte sectors** and the **MBR** partition table (**MBR** stands for Master Boot Record). This organization allows for **4** **primary** partitions only. If you want more than that, you need to create an **extended** partition (using one of the 4 primary slots), and then create **logical**partitions inside. More annoying, on disks with capacity greater than **2TB**, space above this limit is not available.

To work around all these limitations, recent disks use **4096-byte sectors** and the **GPT** partition table (**GPT** stands for GUID – Globally Unique IDentifier – Partition Table). More details are available on the [GPT Wikipedia page](http://en.wikipedia.org/wiki/GUID_Partition_Table).

Historically, two commands exist to manipulate disks and partitions: **fdisk** and **parted**.  
As the **fdisk** command doesn’t handle **GPT**partition tables, it is not advisable to use it any more (for your information, some details are given at the end of this page about the **fdisk** command).  
Recently, a new tool called **gdisk** has been created to deal with **GPT** partition tables, offering an alternative to the **parted**command.

**Caution:** In this tutorial, we are dealing with real disks. Any mistake could **entirely destroy** your system.

## **The parted Command**

To start the **parted** command, type:

# **parted**

GNU Parted 2.1

Using /dev/sda

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted)

To list all the disks and partitions, type:

(parted) **print all**

Model: ATA Hitachi HDP72505 (scsi)

Disk /dev/sda: 500GB

Sector size (logical/physical): 512B/512B

Partition Table: msdos

Number Start End Size Type File system Flags

1 1049kB 525MB 524MB primary ext4 boot

2 525MB 500GB 500GB primary lvm

**Model: ATA Hitachi HDP72505 (scsi)**

**Disk /dev/sdb: 500GB**

**Sector size (logical/physical): 512B/512B**

**Partition Table: msdos**

**Number Start End Size Type File system Flags**

Model: Linux device-mapper (linear) (dm)

Disk /dev/mapper/vg\_root-lv\_root: 497GB

Sector size (logical/physical): 512B/512B

Partition Table: loop

Number Start End Size File system Flags

1 0.00B 497GB 497GB ext4

Model: Linux device-mapper (linear) (dm)

Disk /dev/mapper/vg\_root-lv\_swap: 2147MB

Sector size (logical/physical): 512B/512B

Partition Table: loop

Number Start End Size File system Flags

1 0.00B 2147MB 2147MB linux-swap(v1)

Here, we’ve got a disk called **/dev/sdb** without partition but with a **MBR** partition table (Partition Table: msdos).  
To select the **/dev/sdb** disk, type:

(parted) **select /dev/sdb**

Using /dev/sdb

To create a **GPT** partition table on the **/dev/sdb** disk, type:

(parted) **mktable gpt**

Warning: The existing disk label on /dev/sdb will be destroyed and all data on

this disk will be lost. Do you want to continue?

Yes/No? **Y**

(parted) **print**

Model: ATA Hitachi HDP72505 (scsi)

Disk /dev/sdb: 500GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Number Start End Size File system Name Flags

Note: Type **mktable msdos** to create a **MBR** partition table.

To create a primary partition with the **ext4** type (here starting at **1MB** and finishing at **400GB**), type:

(parted) **mkpart primary ext4 1MB 400GB**

Note1: Specifying **ext4** doesn’t format the partition in **ext4**, it only tags it as **ext4** partition.  
Note2: The partition doesn’t start at **0** but **1MB** to avoid disk alignment problems.  
Note3: To specify all the remaining space, use **-1** as end position.  
Note4: With **parted**, **1GB**=**1000MB**.

Sometimes when setting up the first partition, you will get a warning: “**Warning: The resulting partition is not properly aligned for best performance.**”  
To solve this problem, type the following command where **400GB** is the size of the partition (you can use **100%** instead of **400GB** if you want to allocate all the disk):

(parted) **mkpart primary ext4 0% 400GB**

To check that the first partition is correctly aligned, type:

(parted) **align-check optimal 1**

1 aligned

To create a **swap** partition with a size of **2GB** (here starting at **400GB** and finishing at **402GB**), type:

(parted) **mkpart primary linux-swap 400GB 402GB**

Note1: **parted** checks that both partitions don’t overlap.  
Note2: If, at a later stage, you want to change the type of partition, don’t drop and recreate the partition: format the partition as you want and **parted** will normally detect the new type.

To print the result, type:

(parted) **print**

Model: ATA Hitachi HDP72505 (scsi)

Disk /dev/sdb: 500GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Number Start End Size File system Name Flags

1 1049kB 400GB 400GB ext4 primary

2 400GB 402GB 2000MB primary

To set the **first** partition as **bootable**, type:

(parted) **set 1 boot on**

(parted) **print**

Model: ATA Hitachi HDP72505 (scsi)

Disk /dev/sdb: 500GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Number Start End Size File system Name Flags

1 1049kB 400GB 400GB ext4 primary **boot**

2 400GB 402GB 2000MB primary

Note: Type **set 1 boot off** to remove the **bootable** flag.

To remove the **swap** partition (here partition number **2**), type:

(parted) **rm 2**

To exit the **parted** prompt, type:

(parted) **quit**

To update the disk configuration seen by the kernel, type:

# **partprobe /dev/sdb**

The **parted** command can also be used for **file system** management. However, this usage is deprecated.

## **The gdisk Command**

Install the **gdisk** package:

# **yum install -y gdisk**

Execute the **gdisk** command (here with the **/dev/vda** disk as parameter):

# **gdisk /dev/vda**

GPT fdisk (gdisk) version 0.8.6

Partition table scan:

MBR: MBR only

BSD: not present

APM: not present

GPT: not present

Type **?** to display all the options:

Command (? for help): **?**

b back up GPT data to a file

c change a partition's name

d delete a partition

i show detailed information on a partition

l list known partition types

n add a new partition

o create a new empty GUID partition table (GPT)

p print the partition table

q quit without saving changes

r recovery and transformation options (experts only)

s sort partitions

t change a partition's type code

v verify disk

w write table to disk and exit

x extra functionality (experts only)

? print this menu

Type **p** to print the partition table:

Command (? for help): **p**

Disk /dev/vda: 12582912 sectors, 6.0 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): C6F7C323-530D-40B5-A985-241A1B181354

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 12582878

Partitions will be aligned on 2048-sector boundaries

Total free space is 1318845 sectors (644.0 MiB)

Number Start (sector) End (sector) Size Code Name

1 2048 1026047 500.0 MiB 8300 Linux filesystem

2 1026048 11266047 4.9 GiB 8E00 Linux LVM

Type **n** to create a new partition:

Command (? for help): **n**

Partition number (3-128, default 3): **3**

First sector (34-12582878, default = 11266048) or {+-}size{KMGTP}: **34**

Last sector (34-2047, default = 2047) or {+-}size{KMGTP}: **2047**

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, Enter = 8300): **8300**

Changed type of partition to 'Linux filesystem'

Type **p** to display the partition table:

Command (? for help): **p**

Disk /dev/vda: 12582912 sectors, 6.0 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): C6F7C323-530D-40B5-A985-241A1B181354

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 12582878

Partitions will be aligned on 2048-sector boundaries

Total free space is 1316831 sectors (643.0 MiB)

Number Start (sector) End (sector) Size Code Name

1 2048 1026047 500.0 MiB 8300 Linux filesystem

2 1026048 11266047 4.9 GiB 8E00 Linux LVM

3 34 2047 1007.0 KiB 8300 Linux filesystem

Type **w** to write the partition table 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/vda.

The operation has completed successfully.

To force the kernel to read the updated **partition table**, type:

# **partprobe**

Source: [Sander van Vugt’s video about gdisk (5min/2014)](https://www.youtube.com/watch?v=V5_d_3oZCLs).

## The fdisk Command

To list all the partitions, type:

# **fdisk -l**

To create a **primary** partition on a disk (here **/dev/vda**), type:

# **fdisk /dev/vda**

Press ‘**c**‘, ‘**u**‘, then ‘**p**‘ to print the partition table.  
Then press ‘**n**‘ (for new), type the partition number (between **1** and **4**), the first sector and the size.  
Finally, press ‘**w**‘ to save the partition table.

To delete a **primary** partition on a disk (here **/dev/vda**), type:

# **fdisk /dev/vda**

Press ‘**c**‘, ‘**u**‘, then ‘**p**‘ to print the partition table.  
Then press ‘**d**‘ (for delete) and type the partition number (between **1** and **4**).  
Finally, press ‘**w**‘ to save the partition table.

To set the type of a **primary** partition (here **/dev/vda3**), type:

# **fdisk /dev/vda**

Press ‘**c**‘, ‘**u**‘, then ‘**p**‘ to print the partition table.  
Then press ‘**t**‘ (for tag), type the partition number (here ‘**3**‘) and the partition type (**83** for linux, **8e**for Linux LVM, **82** for swap).  
Finally, press ‘**w**‘ to save the partition table.

To force the kernel to read the updated **partition table**, type:

# **partprobe**

**Create and remove physical volumes, assign physical volumes to volume groups, and create and delete logical volumes**

Before dealing with physical volumes, volume groups and logical volumes, you need to know how to create partitions on disk.

## **LVM Presentation**

**LVM** stands for **L**ogical **V**olume **M**anager. It’s the system component that is able to deal with partitions in a modern way.

Traditionally, Unix disks were divided into slices called partitions. It was possible to decide the size of each partition but, once done, you couln’t change it without destroying all data on it.

With **LVM**, you can allocate disk space in a flexible manner.

**LVM** relies on three main concepts:

* **physical volume** or **pv**: a pv is a whole disk or a partition of a disk,
* **volume group** or **vg**: a vg contains one or more pv; a pv can only belong to one vg,
* **logical volume** or **lv**: a lv represents a portion of a vg; a lv can only belong to one vg; it’s on a lv that you can create a file system.

**LVM** brings flexibility compared to the previous system only based on partitions:

* you can easily increase the size of a logical volume if you’ve got some space available in the associated volume group,
* if you don’t have any available space anymore, you can add a new disk as a physical volume, add it to the volume group and increase the size of your logical volume: the logical volume will span two or more disks without any additional operations,
* creating or extending logical volumes don’t require any downtime.

## **Configuration Report**

Before starting any operation, you need to know the current configuration. Type:

# lsblk -a

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 6G 0 disk

├─vda1 252:1 0 390M 0 part /boot

└─vda2 252:2 0 5.5G 0 part

├─rhel-swap 253:0 0 552M 0 lvm [SWAP]

└─rhel-root 253:1 0 3G 0 lvm /

With the **lsblk** command, you get quickly the following information:

* it is a virtual machine (**vda**, **sda** would indicate a physical server),
* there is only one disk **/dev/vda** of **6GB**,
* the disk is divided into two partitions (**vda1** and **vda2**) respectively with a size of **390MB**and **5.5GB**,
* the **vda1** partition is mounted under **/boot**,
* the **vda2** partition consists in two logical volumes (**lvm**) **swap** and **root** in a volume group called **rhel**,
* the **swap** logical volume is used by the system as a swapping area (**[SWAP]**) of **552MB**,
* the **root** logical volume is mounted under **/** with a size of**3GB**,
* there is around **2GB** of free available space (**5.5GB**–**552MB**–**3GB**=**2GB**) in the **vda2**partition,
* none of the partitions are in **R**ead-**O**nly mode (**RO**=**0**) or **R**e**M**ovable (**RM**=**0**).

Note1: **lsblk** should be the first command to type when dealing with partitions.  
Note2: By default, the **lsblk** command doesn’t display empty devices. The **-a** option corrects this problem.

## **Physical Volume Management**

To create a physical volume (here **/dev/vda**), type:

# pvcreate /dev/vda

To remove a physical volume not belonging to any volume group (here **/dev/vda**), type:

# pvremove /dev/vda

To get the list of the physical volumes, type:

# pvs

## **Volume Group Management**

To create a volume group (here called **vg** using a physical volume **/dev/vda**) with a physical extent size of **8MB**, type:

# vgcreate -s 8m vg /dev/vda

To add a physical volume (here **/dev/vdb**) to an existing volume group (here **vg**), type:

# vgextend vg /dev/vdb

To remove a physical volume (here **/dev/vdb**) from an existing volume group (here **vg**), type:

# vgreduce vg /dev/vdb

To remove an existing volume group (here called **vg**), type:

# vgremove vg

To get the list of the volume groups, type:

# vgs

## **Logical Volume Management**

To create a logical volume (here called **lv\_vol** with a size of **1GB** in the **vg** volume group), type:

# lvcreate --size 1G --name lv\_vol vg

To remove a logical volume (here called **lv\_vol** in the **vg** volume group), type:

# lvremove /dev/vg/lv\_vol

To get the list of the logical volumes, type:

# lvs

## **Additional Resources**

You can also watch **Ralph Nyberg**‘s video about [LVM (18min/2015)](https://www.youtube.com/watch?v=iLQJIDC6MAM) or **Red Hat Training**‘s [Resizing a volume group (5min/2015)](https://www.youtube.com/watch?v=KmmpvmzwE7Y).  
Beyond the **RHCSA** objectives, you can read this tutorial about [Creating a Thinly Provisioned Logical Volume](http://thegeekdiary.com/centos-rhel-how-to-create-a-thinly-provisioned-logical-volume/).

**Configure systems to mount file systems at boot by Universally Unique ID (UUID) or label**

## **Mount with UUID**

**UUID** stands for **U**niversal **U**nique **ID**. It is 128-bit value that uniquely identifies a disk or partition of a disk.

To get the list of disk **UUID**, type:

# blkid

To define the disk or partition that you want to mount (here **lv\_vol**) and append it to the **/etc/fstab**file, type:

# blkid | grep lv\_vol >> /etc/fstab

Then, edit the **/etc/fstab** file and change the new added line with the correct parameters (here **/mnt** as mount point, **ext4** as file system type, etc).

UUID=".." /mnt ext4 defaults 1 2

To mount the disk or partition, type:

# mount /mnt

## **Mount with label**

In addition to **UUID**, a disk or a partition of a disk can receive a label. It is a string of at most **16**characters long in the case of ext2/ext3/ext4.

To assign a label to a disk or partition (here **/dev/vdb1**), type:

# e2label /dev/vdb1 TEST

or

# tune2fs -L TEST /dev/vdb1

To get the label associated with a disk or a partition (here **/dev/vdb1**), type:

# e2label /dev/vdb1

TEST

To mount an **ext4** formatted partition (here **/dev/vdb1**), edit the **/etc/fstab** file and paste the following line:

LABEL=TEST /mnt ext4 defaults 1 2

To mount the disk or partition, type:

# mount /mnt

**Add new partitions and logical volumes, and swap to a system non-destructively**

You’ve got three options when adding some swap space to a system:

* use a logical volume inside a volume group,
* use a new partition,
* use a file system (this is not a good solution performance-wise but could be required if all partitions are already used).

## **Logical Volume Configuration**

If you decide to create a logical volume, follow these steps:

Create a logical volume (here called **lv\_swap** with a size of **1G** in the **vg** volume group):

# lvcreate --size 1G --name lv\_swap vg

Prepare the swap logical volume:

# mkswap /dev/vg/lv\_swap

Add the swap logical volume to the system:

# swapon /dev/vg/lv\_swap

Choose one of these commands to check the result:

# swapon -s

# cat /proc/swaps

Edit the **/etc/fstab** file and add the following line (you can replace the beginning of the line with the **UUID** of the swap logical volume):

/dev/mapper/vg-lv\_swap swap swap defaults 0 0

Note: to remove the swap logical volume, remove the line previously created in the **/etc/fstab** file and type:

# swapoff /dev/vg/lv\_swap

# lvremove /dev/vg/lv\_swap

## **Partition Configuration**

If you decide to create a new partition, follow these steps:

Create a new partition with fdisk (here on the **/dev/vda** disk):

# fdisk /dev/vda

At the fdisk prompt, type ‘**c**‘, then ‘**u**‘ and finally ‘**p**‘ to print the partition table.  
Still at the fdisk prompt, type ‘**n**‘ to create a new partition, type the partition number, the first sector and the partition size.  
Don’t forget to give the swap type to this partition: press ‘**t**‘, then the partition number, then ‘**82**‘.  
Exit the fdisk prompt with the ‘**w**‘ to write the partition table on disk.

Ask the kernel to read again the partition table (where X is the number of the swap partition):

# partprobe /dev/vdaX

Prepare the swap partition:

# mkswap /dev/vdaX

Add the swap partition to the system:

# swapon /dev/vdaX

Choose one of these commands to check the result:

# swapon -s

# cat /proc/swaps

Edit the **/etc/fstab** file and add the following line (you can replace the beginning of the line with the **UUID** of the swap partition):

/dev/vdaX swap swap defaults 0 0

Note: to remove the swap partition, remove the line previously created in the **/etc/fstab** file and type:

# swapoff /dev/vdaX

## Additional Resources

Beyond the exam objectives, you can learn [how to prioritize the devices used for swap partition](http://thegeekdiary.com/centos-rhel-how-to-prioritize-the-devices-used-for-swap-partition/).