Increase diskstorage for Linux VMs

The overall process involves several layers: disk → partition → LVM physical volume → LVM logical volume → filesystem. Each layer has its own size and doesn't grow automatically just because the layer below it got bigger — that's why several steps are needed.

Before you start: Although these steps are generally safe and non-destructive, it's recommended to create a backup or snapshot of the VM beforehand, especially in production environments.

1. Check the new disk size

```
fdisk -l
```

Lists all block devices and their partitions. Check whether the disk (e.g. `/dev/sda`) already shows the larger total size. If not, reboot the VM once so the kernel picks up the new disk size.

2. Resize the partition

```
cfdisk
```

In the interactive menu:

1. Select the affected partition (usually the **last** partition on the disk)
2. Choose **Resize** and confirm the new size (usually defaults to all available free space)
3. Choose **Write** and confirm with `yes`
4. Exit with **Quit**

> `cfdisk` only changes the partition table, not the data itself. Growing a partition is therefore uncritical — it's just important that the partition being resized is the last one on the disk.

3. Verify the partition size

```
fdisk -l <pathtomaindevice> #example path: /dev/sda
```

Check that the target partition (e.g. `/dev/sda3`) now shows the expected new size.

4. Cross-check with `parted` (optional)

```
parted
```

In the `parted` prompt:

1. `print` — shows the current partition table
2. As an alternative to `cfdisk`, you can also resize here using `resizepart`: enter the partition number, then the new end point (or `100%` for maximum size)
3. `quit` — exit the program

5. Grow the LVM Physical Volume (PV)

```
pvresize <pathtonewdevice>
```

Example: `pvresize /dev/sda3`

Extends the physical volume to use the newly available space on the partition. Use `pvs` to verify the new PV size.

6. Grow the LVM Logical Volume (LV)

```
lvextend -l +100%FREE <pathtolocalpartition>
```

`-l +100%FREE` allocates all free space in the volume group to the logical volume. If you only want to allocate part of it, you can use e.g. `-L +20G` instead.

7. Grow the filesystem

```
resize2fs <pathtolocalpartition>
```

Grows the filesystem (ext2/ext3/ext4) to the new size of the logical volume — only this step actually makes the additional space usable by the system.

> ⚠ For **XFS**, `resize2fs` won't work — use `xfs_growfs <mountpoint>` instead (note: this takes the mount point, not the device path). You can check the filesystem type with `df -T`.

8. Verify the result

```
df -h
```

9. Reboot

```
reboot
```

A reboot isn't strictly necessary since the changes are already active, but it's good practice to confirm the VM comes back up cleanly with the new configuration.

Common pitfalls

- **Wrong partition selected** — always check with `fdisk -l` or `lsblk` before each step to confirm which device/partition is affected.
- **Not the last partition** — a partition can generally only be extended into unallocated space immediately following it.

- **Forgetting the filesystem step** — `pvresize`/`lvextend` only grow the LVM layers; the filesystem still needs to be extended separately with `resize2fs`/`xfs_growfs`.
- **XFS with `resize2fs`** — fails or does nothing useful; use `xfs_growfs` instead.
- **No backup** — always back up before modifying partition tables.

Revision #5
Created 2022-12-21 13:14:35 UTC
Updated 2026-08-21 08:02:16 UTC by Luca Noah Caprez