BLOG (HTTPS://PACKETPUSHERS.NET/BLOG/) > HOW-TO (HTTPS://PACKETPUSHERS.NET/CATEGORY/HOW-TO/) | NOVEMBER 12, 2021

TAGS: Linux (https://packetpushers.net/tag/linux/)

How to Extend the Default Ubuntu LVM Partition

JOHN W KERNS (HTTPS://PACKETPUSHERS.NET/AUTHOR/JOHN-W-KERNS/)

POV: You're a sysadmin who set up a one-off Linux machine for an app you needed, and now it's out of disk space.

You originally spun up a VM, installed a recent Ubuntu OS, and just hit *Next, Next, Finish* through the guided install. Linux is not your bread and butter, you usually deal in Windows, and you just need to get this done.

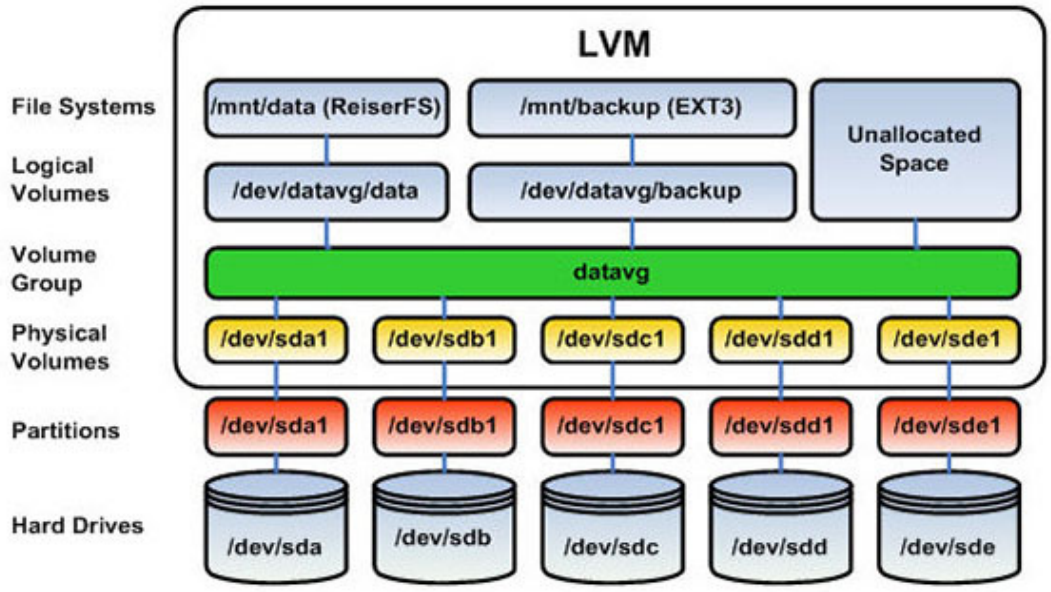
Approx Reading Time: 10 minutes

> Note: In my examples, I'm using an "all-defaults" setup of **Ubuntu 20.04 Server** with a single **100GB disk**.

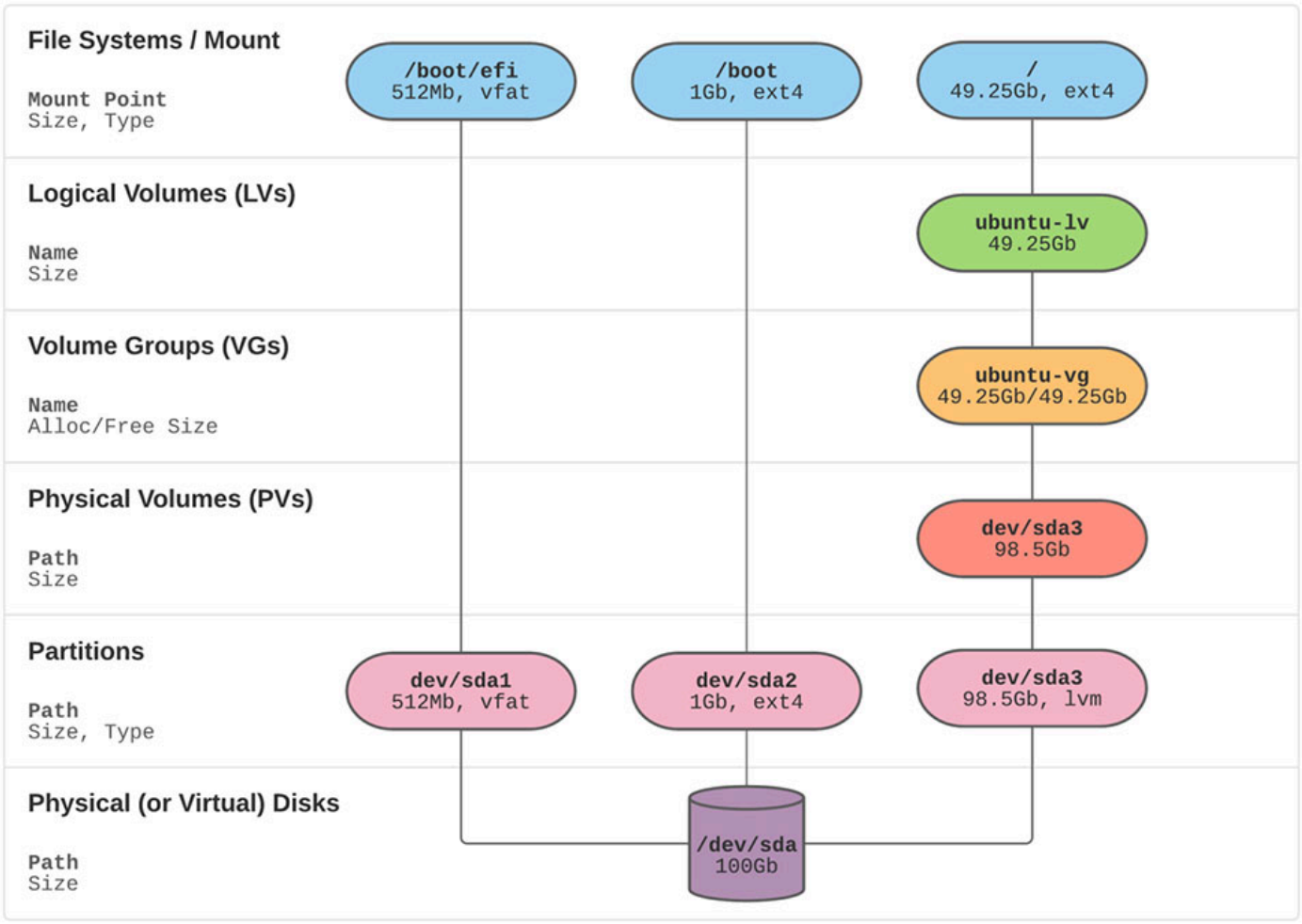
Linux LVM Briefly Explained

If you followed the default settings in the Ubuntu installation, then the storage for your Linux OS is probably using the Logical Volume Manager (LVM). LVM is an abstraction framework which exists between your physical (or virtual) disks and your Linux file system (which is likely ext4 (https://en.wikipedia.org/wiki/Ext4)). It is used to group separate block devices (partitions) together into Volume Groups (VGs), and then chop those VGs up into logical block devices, or Logical Volumes (LVs). LVs are the abstracted block devices upon which your usable file system resides.

Below is a good visualization of how LVM works. In this example, we have five different disks, each with a single partition mapped to Physical Volumes (PVs), all being grouped into a single Volume Group (VG). The Volume Group is chopped up into two different Logical Volumes (LVs), and each LV is being used for a filesystem.

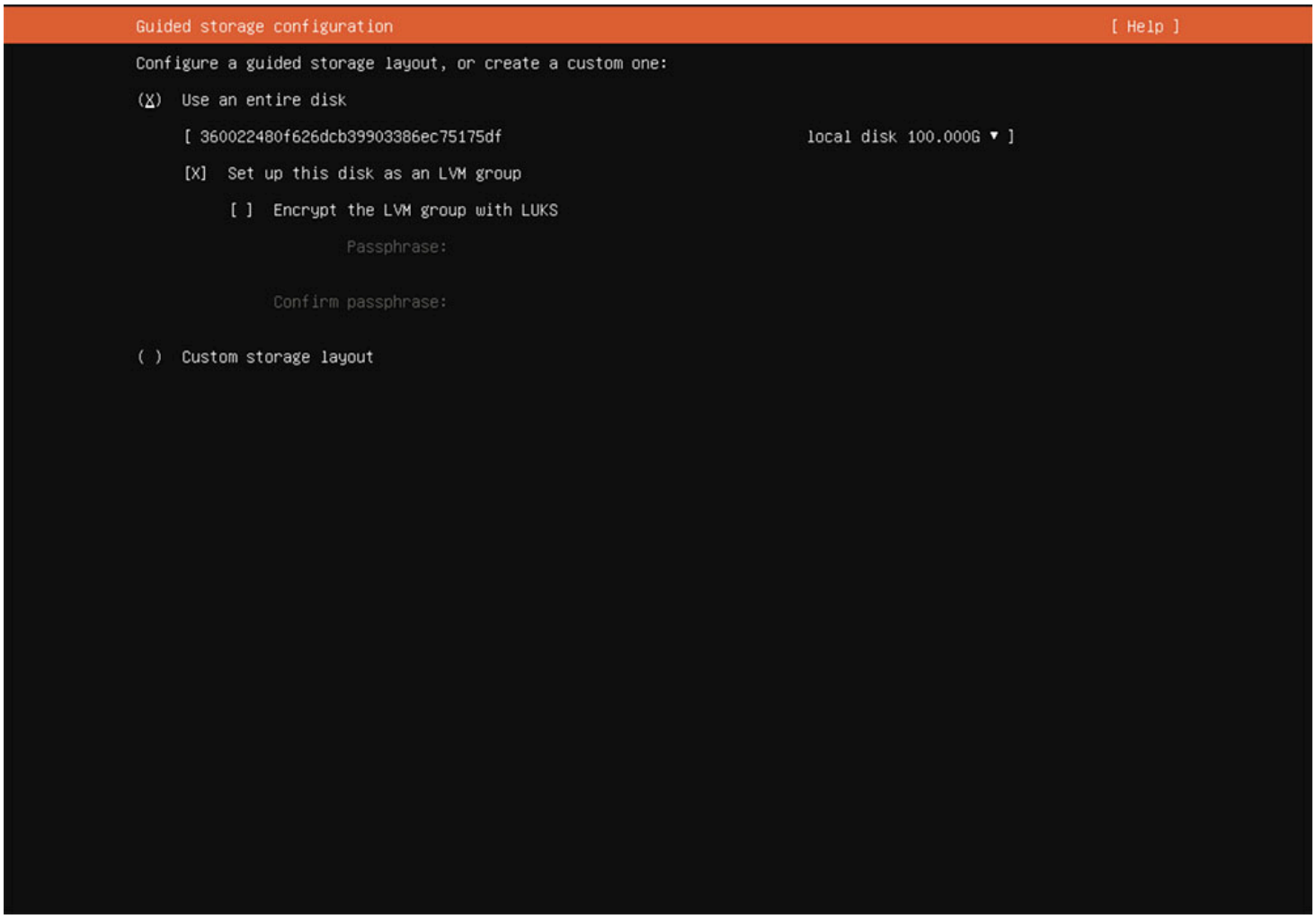


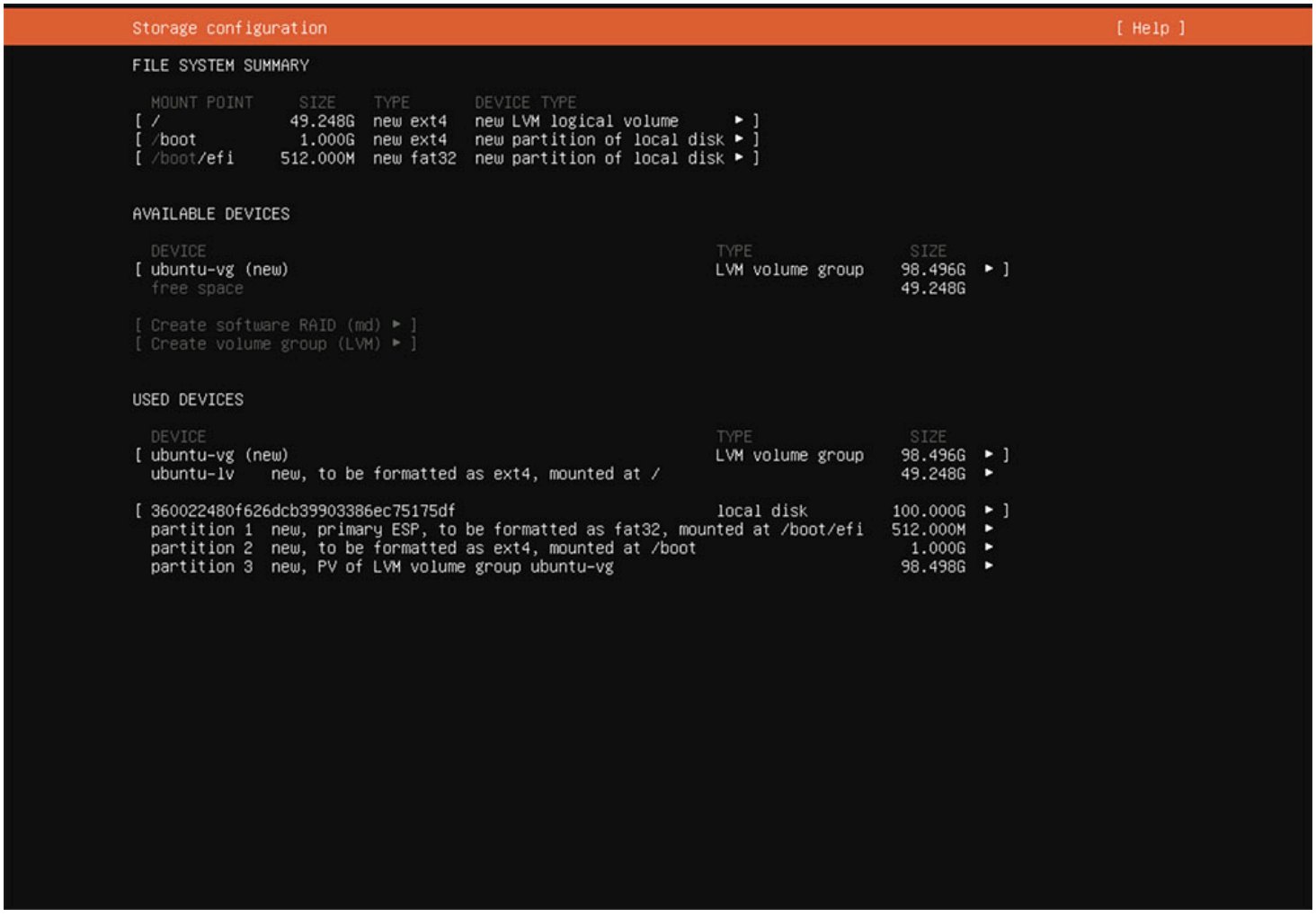
Using a similar visualization, the below diagram shows how the Ubuntu installer (using all default options) divided up my 100GB disk.



Ubuntu Installer Default Settings

When installing Ubuntu, it has you approve a storage layout in a couple different screens (shown below). By default this storage layout will have a couple small boot partitions, and a third partition, which will be used by your LVM to create your root filesystem. You should be able to see the consistency between the screens below and the diagram above.

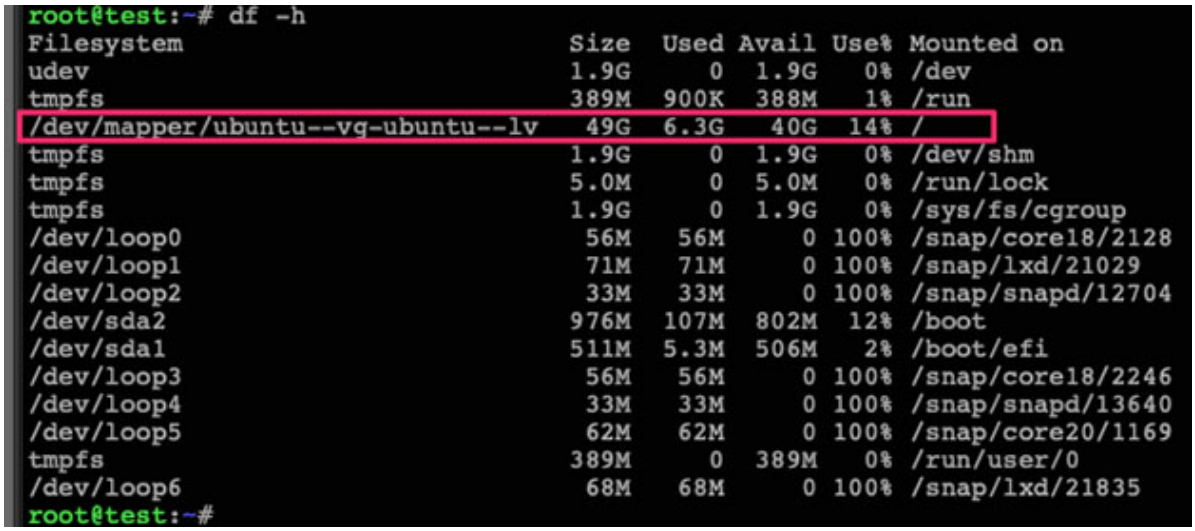

Guided storage configuration
[Help]
Configure a guided storage layout, or create a custom one:
(X) Use an entire disk
[360022480f626dcb39903386ec75175df local disk 100.000G ▾]
[X] Set up this disk as an LVM group
[ ] Encrypt the LVM group with LUKS
Passphrase:
Confirm passphrase:
( ) Custom storage layout


Storage configuration
[Help]
FILE SYSTEM SUMMARY
MOUNT POINT SIZE TYPE DEVICE TYPE
[/ 49.248G new ext4 new LVM logical volume ▸]
[/boot 1.000G new ext4 new partition of local disk ▸]
[/boot/efi 512.000M new fat32 new partition of local disk ▸]
AVAILABLE DEVICES
DEVICE TYPE SIZE
[ubuntu-vg (new) LVM volume group 98.496G ▸]
free space 49.248G
[Create software RAID (md) ▸]
[Create volume group (LVM) ▸]
USED DEVICES
DEVICE TYPE SIZE
[ubuntu-vg (new) LVM volume group 98.496G ▸]
ubuntu-lv new, to be formatted as ext4, mounted at / 49.248G ▸
[360022480f626dcb39903386ec75175df local disk 100.000G ▸]
partition 1 new, primary ESP, to be formatted as fat32, mounted at /boot/efi 512.000M ▸
partition 2 new, to be formatted as ext4, mounted at /boot 1.000G ▸
partition 3 new, PV of LVM volume group ubuntu-vg 98.498G ▸

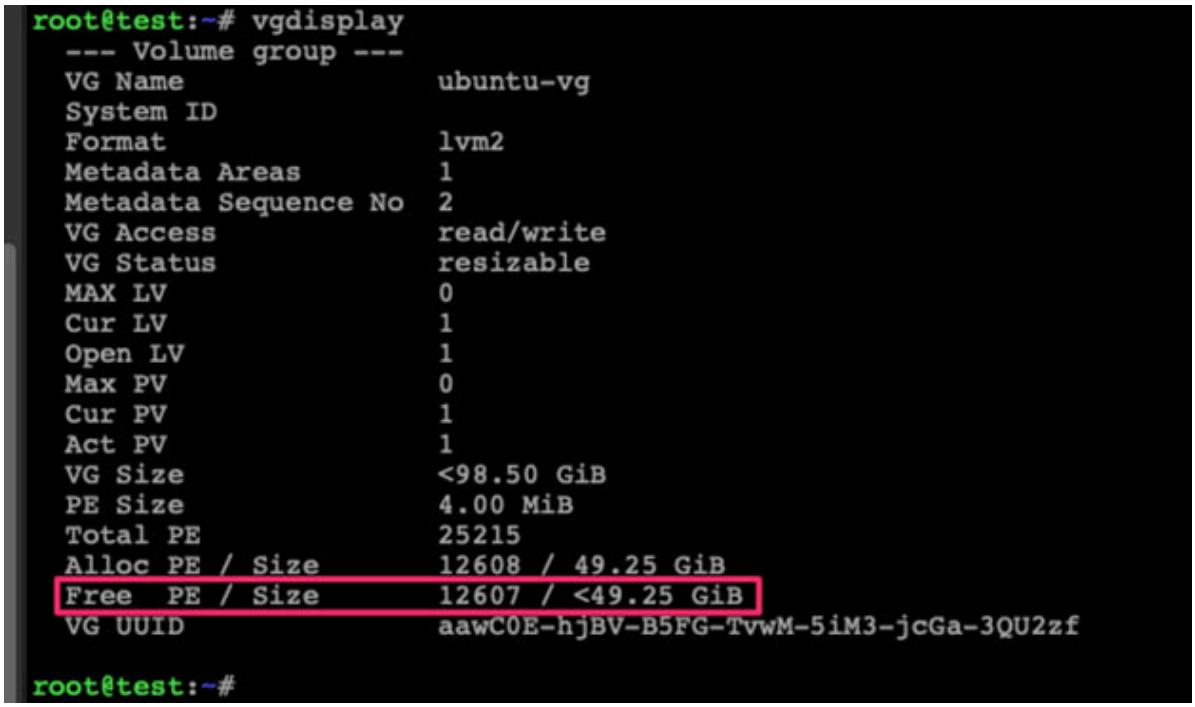
Use Your Default Free Space

As you can see above: the Ubuntu installer (by default) **left almost half of my disk space unusable by the root file system!** I've looked around to find an explanation on why these are the default settings, but can't find anything. Before extending your underlying hypervisor disk or storage volume, you may want to see if you have free space available and ready to be used to extend your existing file system. If you used the Ubuntu defaults during installation, then there is a good chance you have this free space.

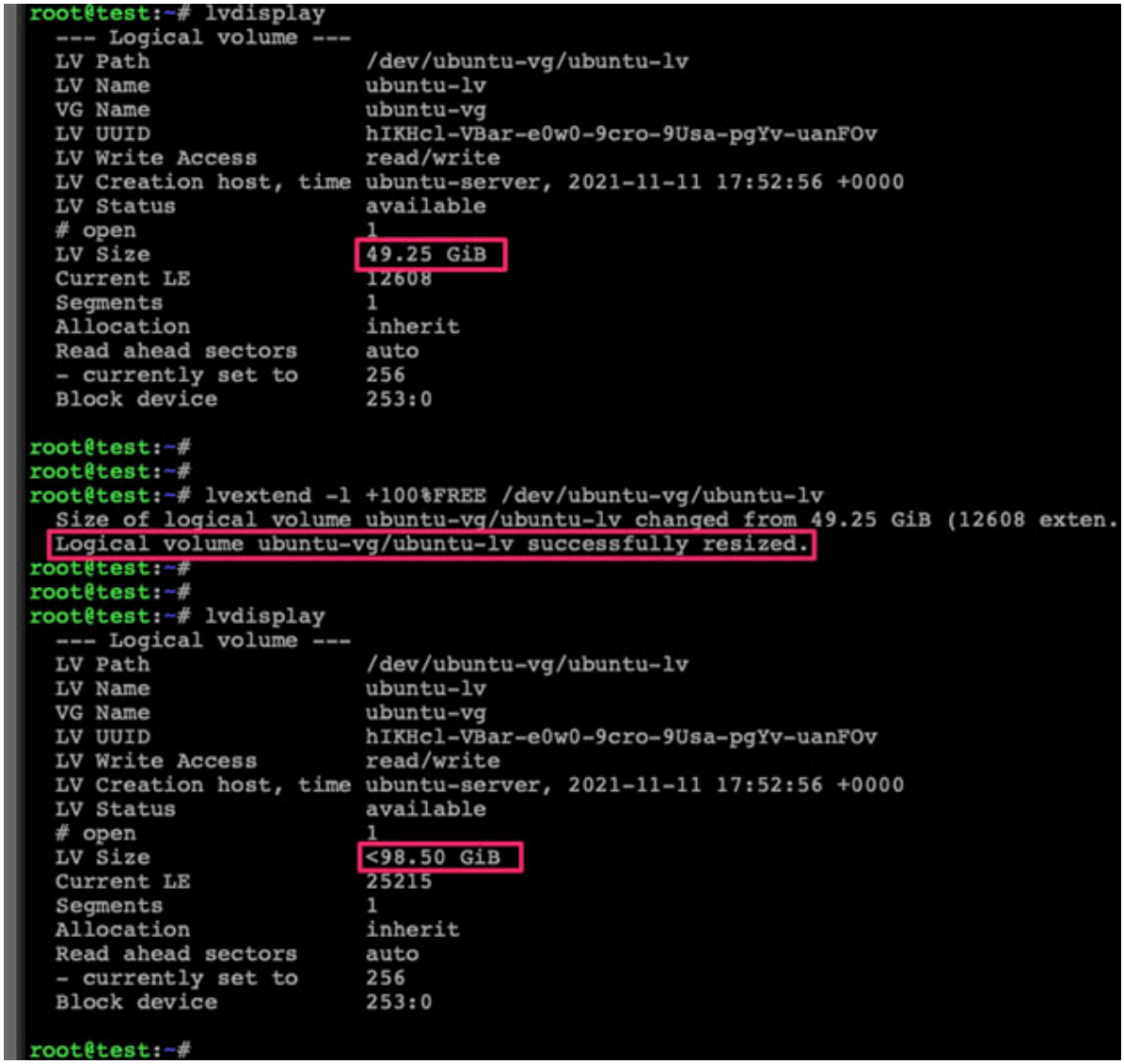
Start by checking your root filesystem free space with df -h. As you can see I am only using **14%** of my **~49GB** volume, but we'll pretend I'm close to 100% and need to make that 49GB volume larger.



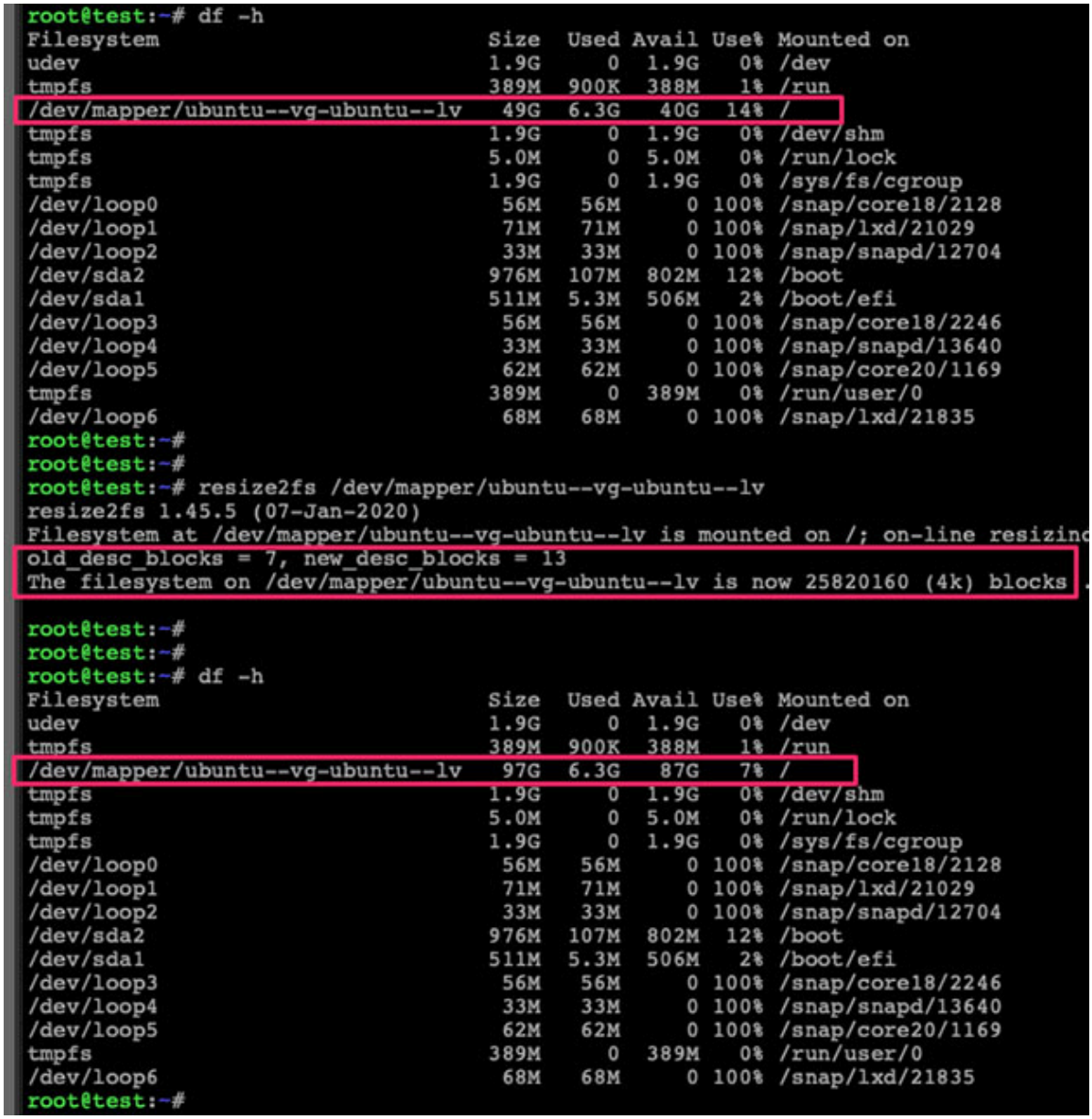
To check for existing free space on your Volume Group (where it is left by the installer default settings), run the command vgdisplay and check for free space. Here you can see I have **49.25GB** of free space ready to be used. If you don't have any free space, move on to the next section to use some free space from an extended physical (or virtual) disk.



To use up that free space on your Volume Group (VG) for your root Logical Volume (LV), first run the lvdisplay command and check the Logical Volume size, then run lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv to extend the LV to the maximum size usable, then run lvdisplay one more time to make sure it changed.



At this point you have increased the size of the block volume where your root filesystem resides, but you still need to extend the filesystem on top of it. First, run df -h to verify your (almost full) root file system, then run resize2fs /dev/mapper/ubuntu–vg-ubuntu–lv to extend your filesystem, and run df -h one more time to make sure you're successful.

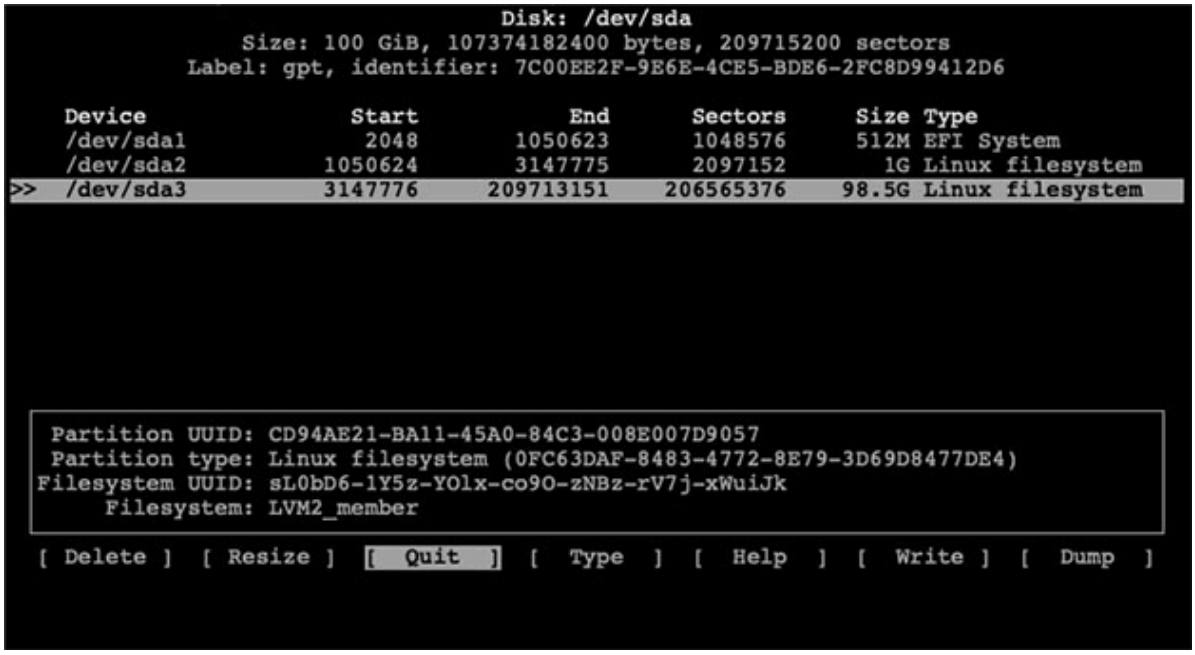


And that's it. You just allocated the free space left behind by the Ubuntu installer to your root filesystem. If this is still not enough space, continue on to the next section to allocate more space by extending an underlying disk.

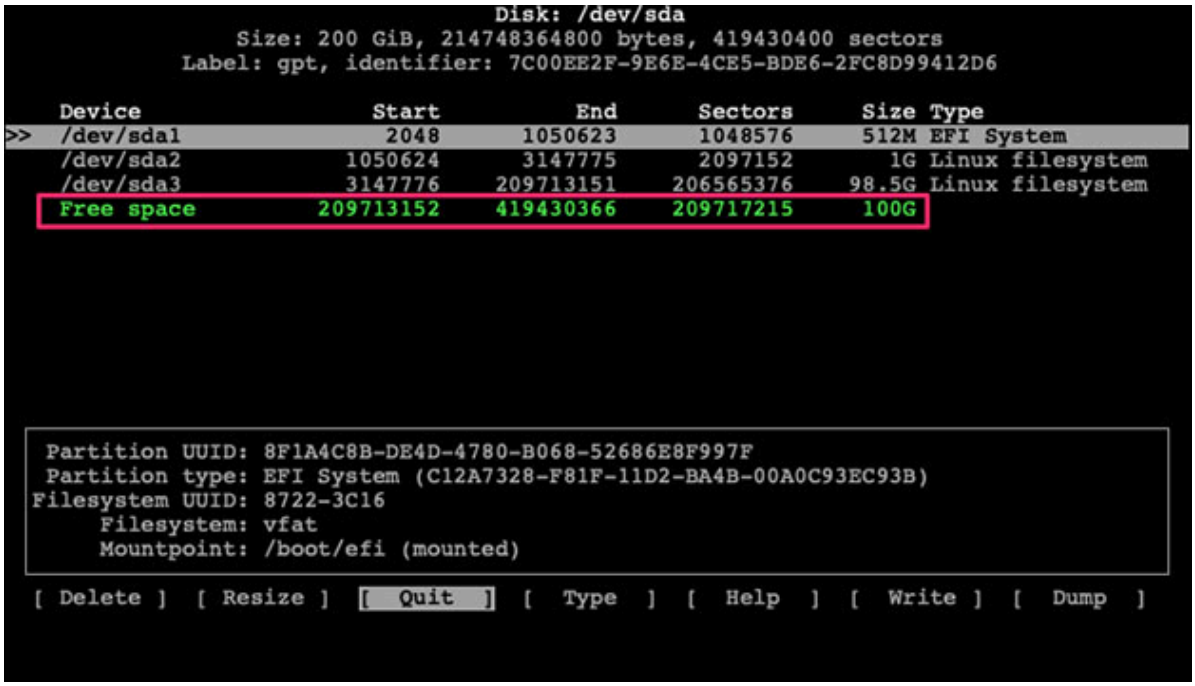
Use Space from Extended Physical (or Virtual) Disk

First you need to increase the size of the disk being presented to the Linux OS. This is most likely done by expanding the virtual disk in KVM/VMWare/Hyper-V or by adjusting your RAID controller / storage system to increase the volume size. You can often do this while Linux is running; without shutting down or restarting. I've extended my **100GB** disk to **200GB** for my example machine.

Once that is done, you may need to get Linux to rescan the disk for the new free space. Check for free space by running cfdisk and see if there is free space listed, use "q" to exit once you're done.



If you don't see free space listed, then initiate a rescan of /dev/sda with echo 1>/sys/class/block/sda/device/rescan. Once done, rerun cfdisk and you should see the free space listed.



Select your /dev/sda3 partition from the list and then select "**Resize**" from the bottom menu. Hit **ENTER** and it will prompt you to confirm the new size. Hit **ENTER** again and you will now see the /dev/sda3 partition with a new larger size.

Select "**Write**" from the bottom menu, type **yes** to confirm, and hit **ENTER**. Then use "**q**" to exit the program.

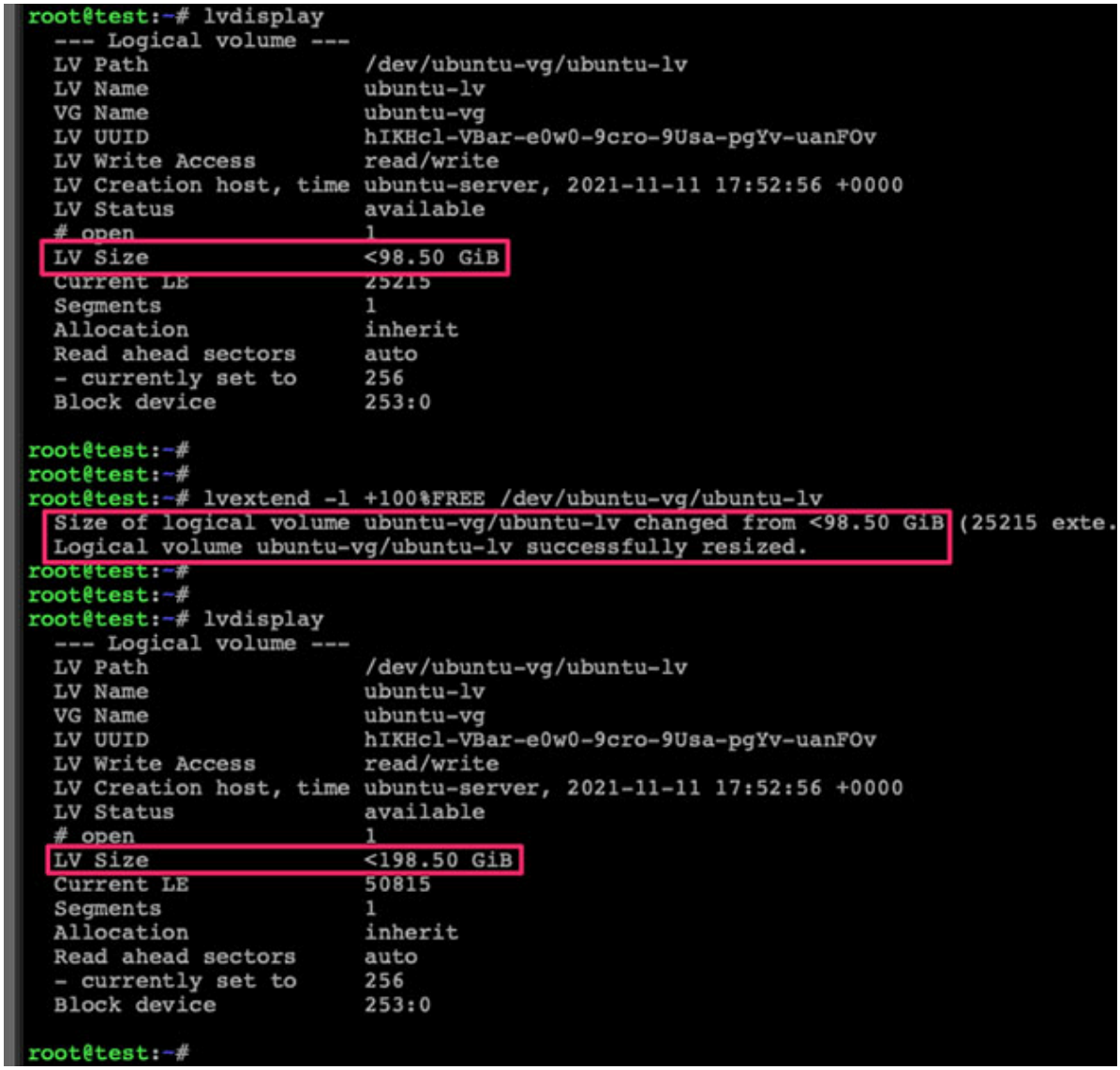
Now that the LVM partition backing the /dev/sda3 Physical Volume (PV) has been extended, we need to extend the PV itself. Run pvresize /dev/sda3 to do this and then use pvdisplay to check the new size.

```
root@test:~# pvresize /dev/sda3
  Physical volume "/dev/sda3" changed
  1 physical volume(s) resized or updated / 0 physical volume(s) not resized
root@test:~#
root@test:~#
root@test:~#
root@test:~#
root@test:~# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda3
  VG Name               ubuntu-vg
  PV Size               <198.50 GiB / not usable 1.98 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              50815
  Free PE               25600
  Allocated PE          25215
  PV UUID               sL0bD6-1Y5z-YOlx-co9O-zNBz-rV7j-xWuiJk

root@test:~#
```

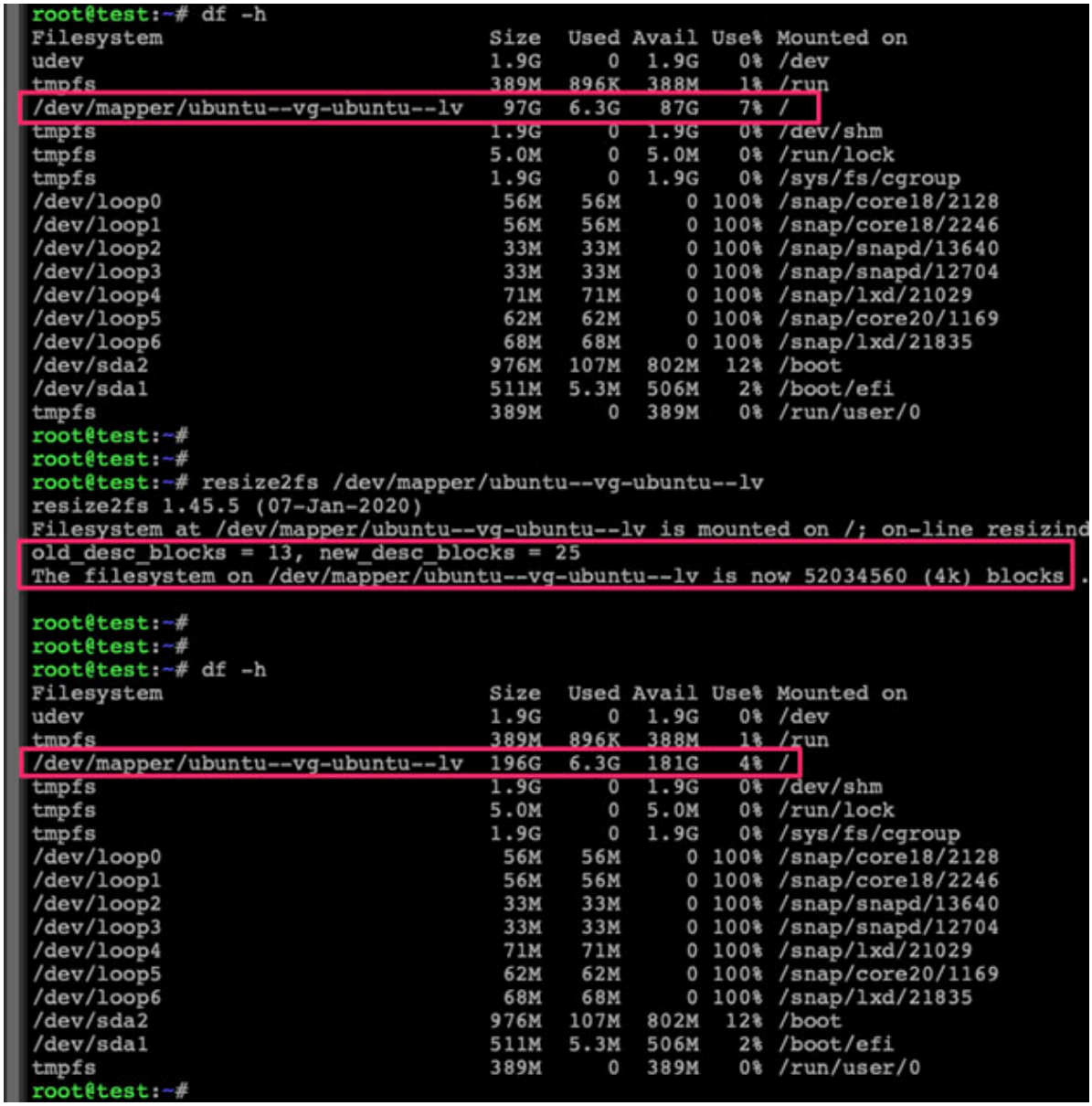
As you can see above, my PV has been increased from 98.5GB to 198.5GB. Now let's check the Volume Group (VG) free space with vgdisplay.

```
root@test:~# vgdisplay
  --- Volume group ---
  VG Name               ubuntu-vg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <198.50 GiB
  PE Size               4.00 MiB
  Total PE              50815
  Alloc PE / Size       25215 / <98.50 GiB
  Free  PE / Size       25600 / 100.00 GiB
  VG UUID               aawC0E-hjBV-B5FG-TvwM-5iM3-jcGa-3QU2zf

root@test:~#
```

We can see above that the VG has 100GB of free space. Now let's check the size of our upstream Logical Volume (LV) using lvdisplay, extend the LV to use up all the VG's free space with lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv, and then check the LV one more time with lvdisplay to make sure it has been extended.



At this point, the block volume underpinning our root filesystem has been extended, but the filesystem itself has not been resized to fit that new volume. To do this, run df -h to check the current size of the file system, then run resize2fs /dev/mapper/ubuntu–vg-ubuntu–lv to resize it, and df -h one more time to check the new file system available space.

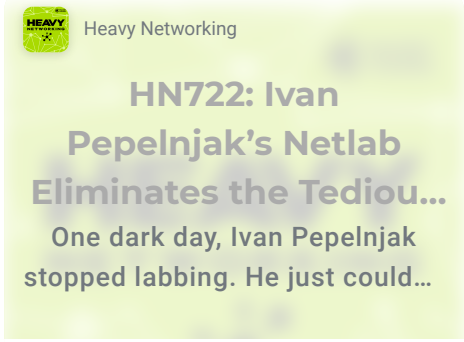


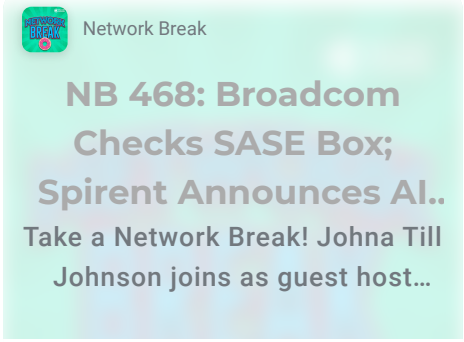
And there you go. You've now taken an expanded physical (or virtual) disk and moved that free space all the way up through the LVM abstraction layers (https://packetpushers.net/podcast/day-two-cloud-081-abstractions-should-save-typing-not-thinking/) to be used by your (critically full) root file system. Time to check it off the to-do list and move on to the next IT emergency.

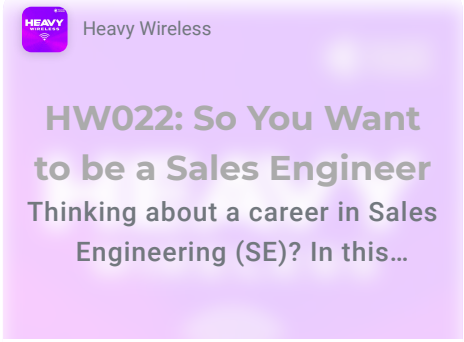
About John W Kerns: John Kerns is a network and automation engineer for a VAR based in Southern California and has been in the industry for over 12 years. He maintains a few open-source projects on GitHub and writes for Packet Pushers.

GITHUB (HTTPS://GITHUB.COM/PACKETSAR/))

Related podcasts







(https://packetpushers.net)

Packet Pushers is a highly trusted source of technical deep-dives, industry news & analysis, and community for networking professionals.

(https://www.youtube.com/channel/UC7vAUu1TQAwzuq8wajJw4kA)

(https://www.linkedin.com/company/packet-pushers-interactive-llc)

(https://www.facebook.com/PacketPushersNet/)

LISTEN

Day Two Cloud (https://packetpushers.net/podcast/day-two-cloud/)

Heavy Networking (https://packetpushers.net/podcast/heavy-networking/)

Heavy Strategy (https://packetpushers.net/podcast/heavy-

strategy/)

Heavy Wireless (https://packetpushers.net/podcast/heavy-wireless/)

IPv6 Buzz (https://packetpushers.net/podcast/ipv6-buzz/)

Kubernetes Unpacked (https://packetpushers.net/podcast/kubernetes-unpacked/)

Network Automation Nerds (https://packetpushers.net/podcast/network-automation-nerds/)

Network Break (https://packetpushers.net/podcast/network-break/)

Packet Protector (https://packetpushers.net/podcast/packet-protector/)

WATCH

Network tutorials (https://packetpushers.net/videos/#networking-tutorials)

(https://packetpushers.net/videos/#video-bytes)

READ

Blog (https://packetpushers.net/blog/)

Human Infrastructure (https://packetpushers.net/newsletters/#human-infrastructure)

Packet Capture (https://packetpushers.net/newsletters/#packet-capture)

GROW

Networking jobs (https://jobs.packetpushers.net/)

Slack community (https://packetpushers.net/community/)

CONNECT

About Packet Pushers (https://packetpushers.net/about/)

Meet the hosts (https://packetpushers.net/hosts/)

Become a sponsor (https://packetpushers.net/sponsors/)

Pitch your news
(https://packetpushers.net/pitch/)

Book us to speak
(https://packetpushers.net/speaking/)

Say hello
(https://packetpushers.net/hello/)