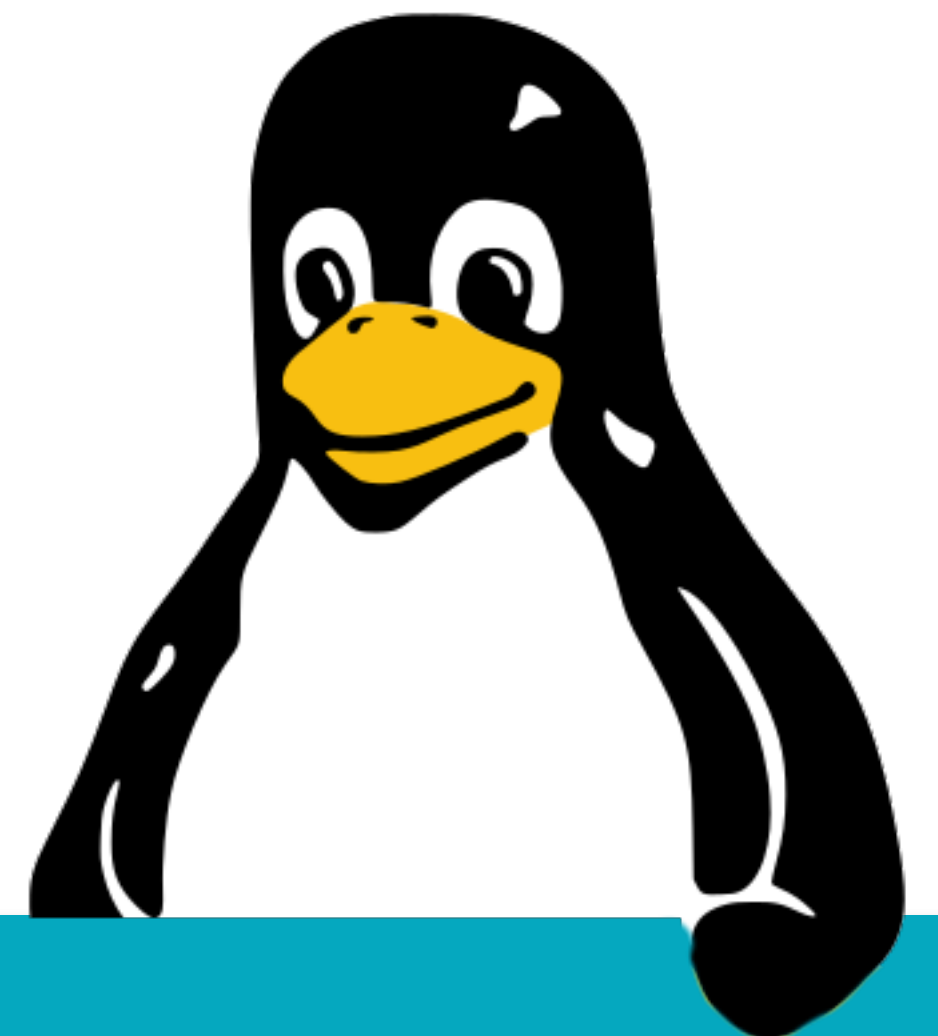


Linux, day 10

This lab is licensed under Creative Commons BY-NC-SA 4.0.
<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

You are free to share and adapt, but NOT for commercial purposes and you must attribute the source and share your own adaptations under the same license.



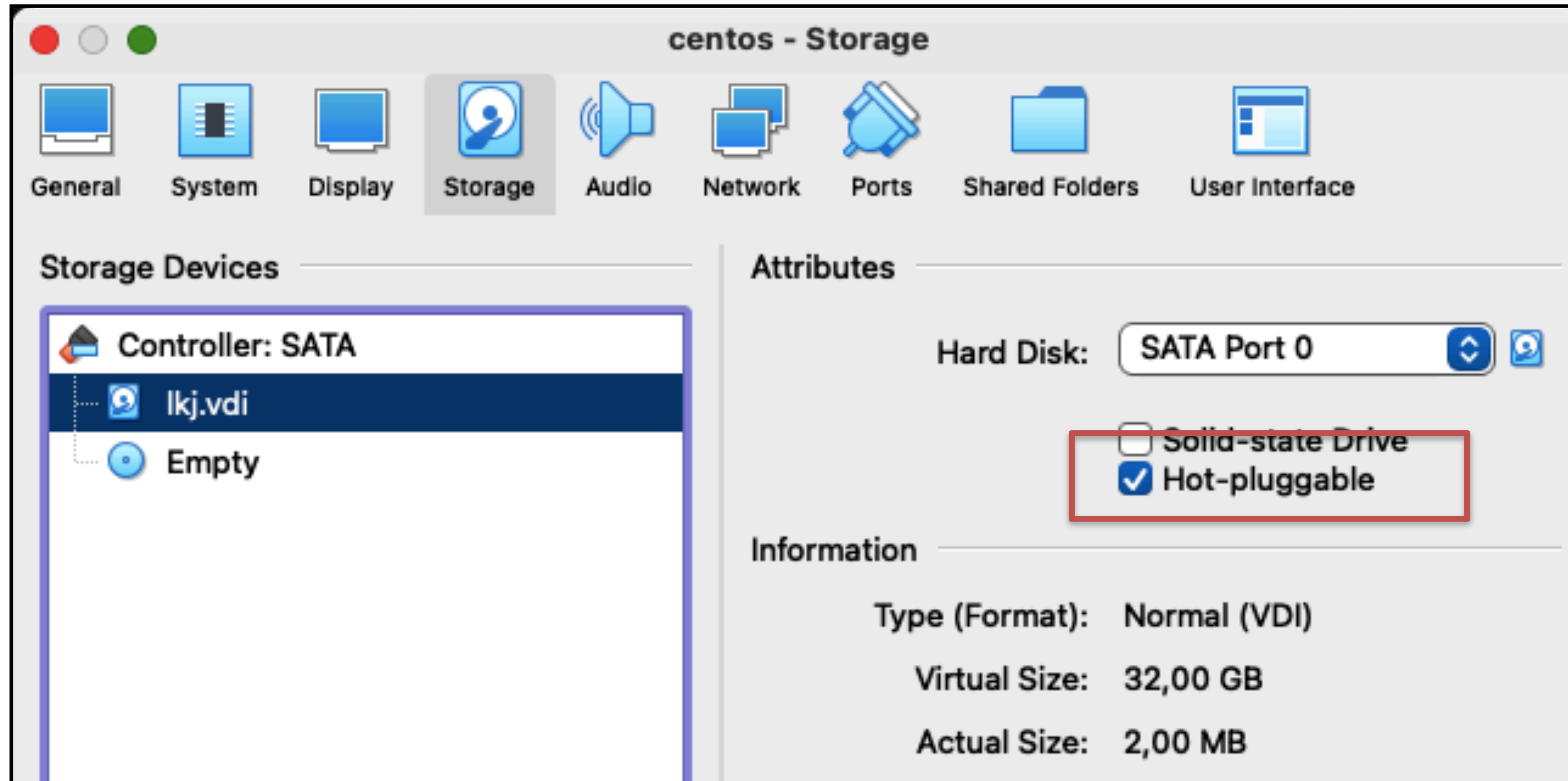
MDadm



Prepping your lab: VBox

- In VirtualBox, change the three dummy disks.
 - Enable hot-plugging.

Prepping your lab: VBox



Prepping your lab: mounts

- In the Linux VM verify the disks are unused.
 - *mount | grep "sd?"*
 - Unmount your mounts of *sdb*, *sdc* and *sdd*.
 - *grep -i "sd?" /etc/fstab*
 - Remove any lines for *sdb*, *sdc* and *sdd*.
 - *sudo mkdir /mnt/data*

Prepping disks for RAID

- Some people suggest you partition them,
 - Others say to just use the whole device.
 - *mdadm* says "partition table will be meaningless"
- Apply GPT partitioning.
 - Use partition type "*Linux RAID*" (29) in *fdisk*.
 - Or "*FD00*" in *gdisk*.

Prepping our disks (three times)

```
$ sudo fdisk /dev/sdb  
g  
n    # Three times enter, y to overwrite  
t  
29   # Type RAID  
w
```

Making a RAID1, with spare

- Remember how to use those backslashes?

```
$ sudo mdadm --create --verbose \  
  --level=1 --metadata=1.2 \  
  --raid-devices=2 /dev/md/MyRAID1 \  
  /dev/sdb /dev/sdc \  
  --spare-devices=1 /dev/sdd
```


Then use it

- Format, mount and use.

```
$ sudo mkfs.ext4 /dev/md/MyRAID1
```

```
$ sudo mount /dev/md/MyRAID1 /mnt/data
```

```
$ sudo touch /mnt/data/testfile
```

Saving your array config

- The location of the config file differs per Linux!
 - *sudo find /etc -name "mdadm.conf"*

```
$ sudo mdadm --detail --scan | \
sudo tee -a /etc/mdadm.conf
# Change that path!
```

Checking RAID health

- *mdadm* software RAID is pretty basic!
 - *cat /proc/mdstat*
 - *sudo mdadm --detail /dev/md/MyRAID1*
 - *journalctl | grep -e "kernel: md" -e "mdadm"*

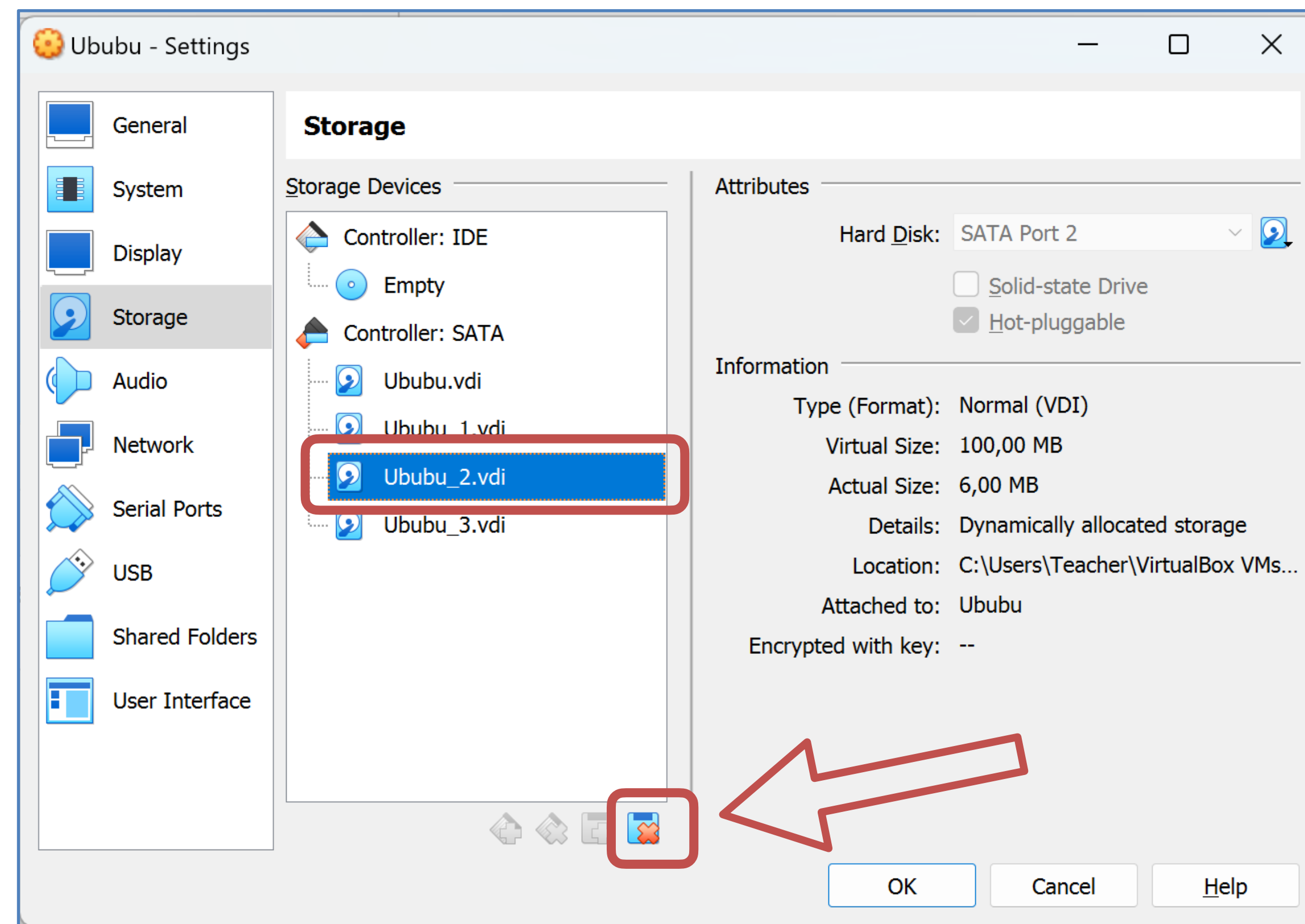
Fun time

- On the guest OS, keep an eye on RAID.

```
$ sudo mdadm --monitor /dev/md/MyRAID1
```

Let's break it!

- Remove the second 100MB virtual disk.



What happened?

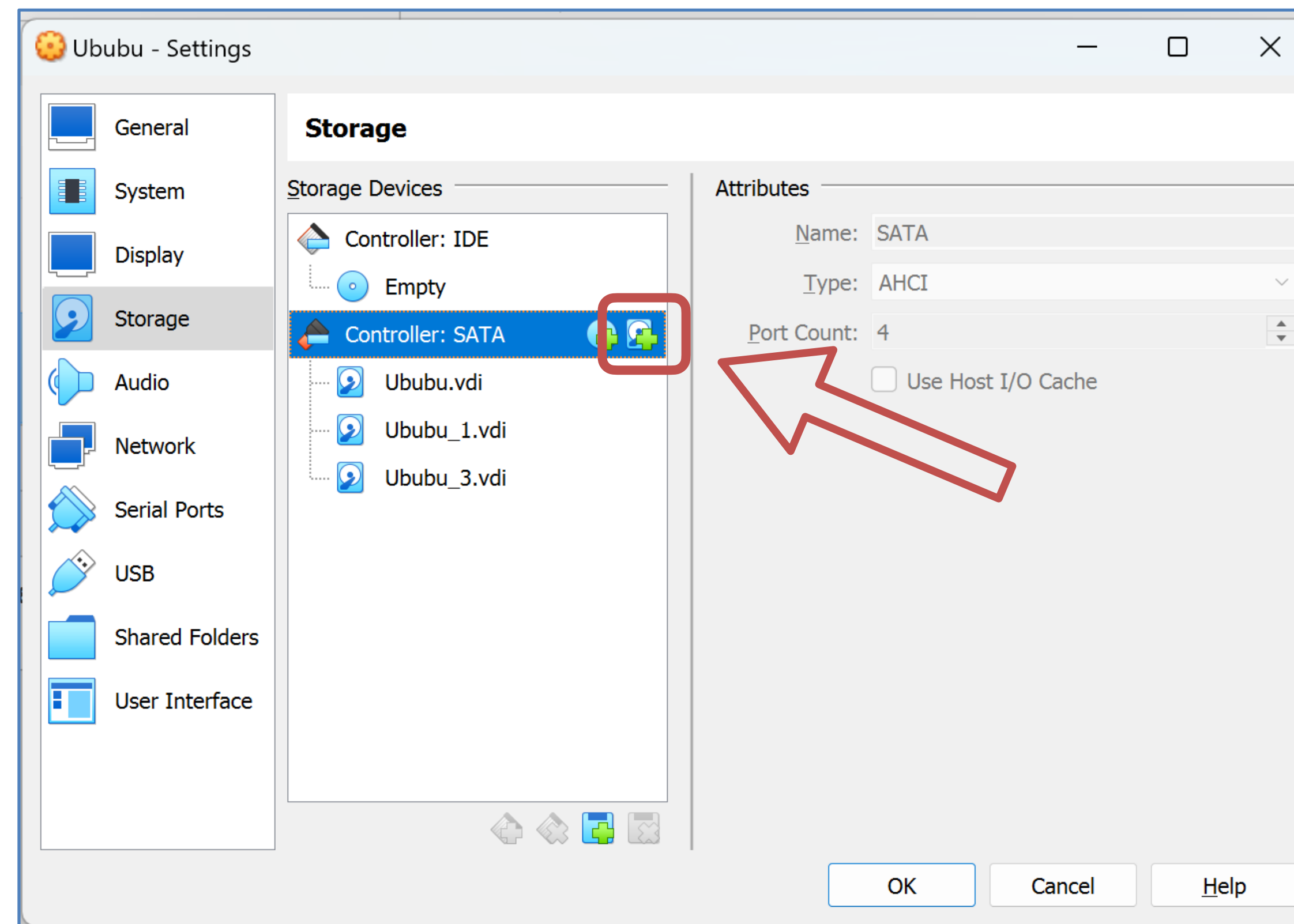
- *mdadm* noticed the failure,
 - Grabbed the hot spare and started rebuilding.
 - The failed disk disappeared.
- Check *"/proc/mdstat"*, *"mdadm"* and *"journalctl"*.

Question!?

- What would be a nice way to test availability?
- If your boss wants you to prove "*mdadm*" is good,
 - How would you run the tests?
 - Can Python or shell scripts help?

Let's rebuild

- Re-attach the virtual disk you removed.



Let's rebuild

- And we will re-add the disk to the RAID set.

```
$ sudo mdadm --manage /dev/md/MyRAID1 \  
--add /dev/sdc
```

```
$ sudo cat /proc/mdstat
```

Done? Break it all.

```
$ sudo umount /mnt/data  
$ sudo mdadm --stop /dev/md/MyRAID1  
$ sudo rm /etc/mdadm.conf # adjust path!  
  
$ sudo mdadm --zero-superblock /dev/sdb  
$ sudo mdadm --zero-superblock /dev/sdc  
$ sudo mdadm --zero-superblock /dev/sdd
```

For your homework: RAID5

- Needs three virtual disks!

```
$ sudo mdadm --create --verbose \  
--level=5 --metadata=1.2 \  
--raid-devices=3 /dev/md/MyRAID5 \  
/dev/sdb /dev/sdc /dev/sdd
```

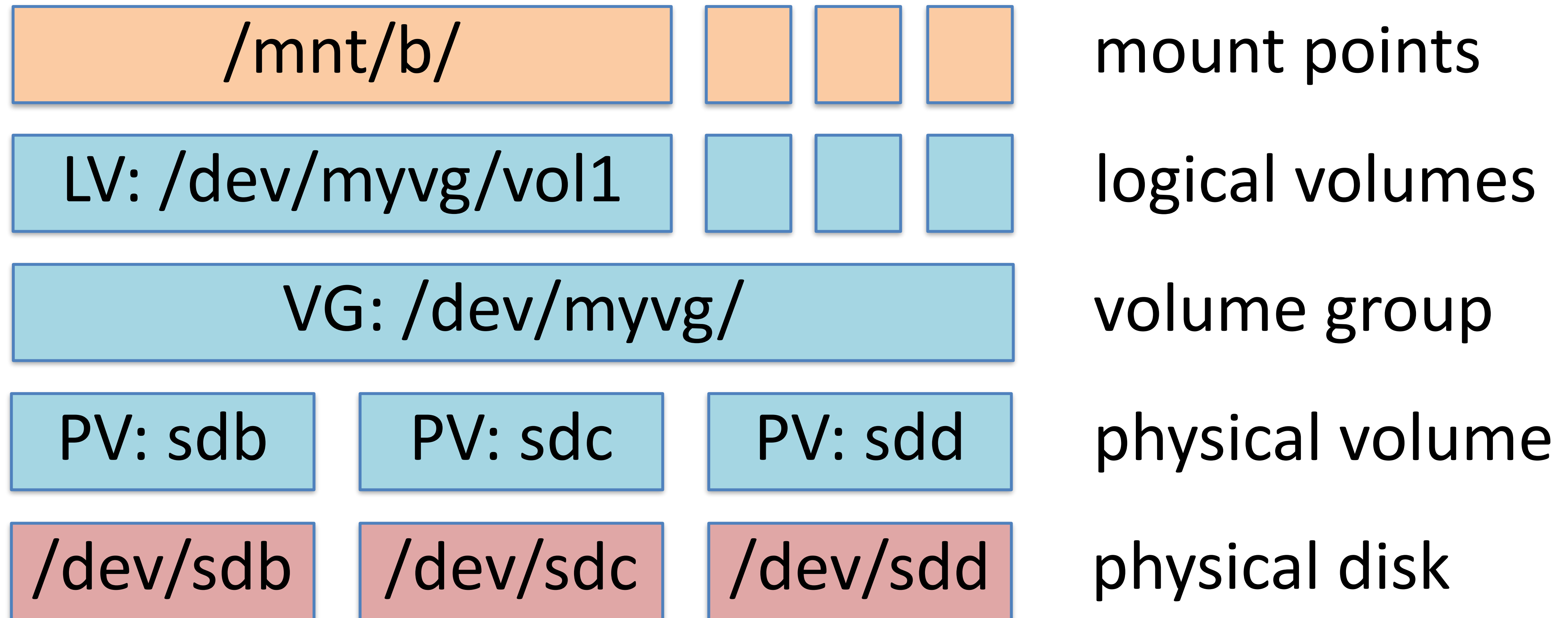
LVM



The problem we need to solve

- Partitioning disks allows some flexibility.
- But growing, shrinking and moving?
 - That's a huge hassle!
- LVM allows much greater flexibility,
 - Through another layer of virtualization.

LVM concepts



Prepping our disks (three times)

```
$ sudo fdisk /dev/sdb  
g  
n    # Three times enter, y to overwrite  
t  
30   # Type LVM
```

Creating phys. volumes

- This adds the device to LVM control

```
$ sudo pvcreate /dev/sdb1  
$ sudo pvcreate /dev/sdc1  
$ sudo pvcreate /dev/sdd1
```


That may fail...

- "*pvcreate*" will notice if disks are re-used.
 - Wipe and try again!

```
$ sudo wipefs -a /dev/sdb  
$ sudo wipefs -a /dev/sdc  
$ sudo wipefs -a /dev/sdd
```

Making the volume group

- This collects the disks for usage.

```
$ sudo vgcreate myvg \  
/dev/sdb1 /dev/sdc1 /dev/sdd1
```

```
$ sudo vgdisplay myvg
```

Our first volume!

- This collects the disks for usage.

```
$ sudo lvcreate -L 50M -n vol1 myvg
```

```
$ sudo vgdisplay
```

```
$ sudo lvdisplay
```

Then use it

- Format, mount and use.

```
$ sudo mkfs.ext4 /dev/myvg/vol1
```

```
$ sudo mount /dev/myvg/vol1 /mnt/data
```

```
$ sudo touch /mnt/data/testfile
```

Check things out

- The commands start with pv*, vg*, lv*.

pvdisplay	Show all physical volumes
vgdisplay	Show all volume groups
lvdisplay	Show all logical volumes

Resizing file systems

- For example, on a volume that was 50 MB.

```
$ sudo lvresize -L 70M /dev/myvg/vol1
```

```
$ sudo resize2fs /dev/myvg/vol1
```

Tear it all down!

- Breaking what we made is easy.

```
$ sudo umount /mnt/data  
$ sudo lvremove /dev/myvg/vol1  
$ sudo vgremove /dev/myvg  
$ sudo pvremove /dev/sdb1 /dev/sdc1 \  
/dev/sdd1
```

Want more?

- You can have multiple volume groups.
 - Each with multiple volumes.
- You can add more disks.
- You can grow existing groups and volumes.

What will we do today?

- ~~Recap~~
- ~~(More) storage management~~
 - ~~Devices, MDadm, LVM~~
- TLS, cryptography and certificates
- Extra lab: NFS server
- Closing: homework and Q&A

LAB: NFS server



Time for research

- Can you make the following?
 - Setup your Ubuntu VM as NFS server.
 - Make /var/userdata on Ubuntu.
 - Make that directory an NFS file share.
 - Setup Fedora as NFS client, and mount the share.

Closing

Homework

- Reading:
 - Chapter 12
 - Chapter 21
 - Chapter 23
 - Chapter 26

Homework

- Go do:
 - Retry your RAID1 setup, incl. the "failed" device.
 - Use your three disks to make one *mdadm* RAID5.
 - On the RAID5 set, make an LVM volume group
 - And a 100MB logical volume, on */mnt/data*.

Homework

/mnt/b/

mount

LV: /dev/myvg/vol1

log. vol

VG: /dev/myvg/

vol. grp.

PV: /dev/md/MyRAID5

phys. vol.

/dev/md/MyRAID5

MDadm

/dev/sdb

/dev/sdc

/dev/sdd

phys. disk

Advanced homework

- Go do:
 - Take the 100MB logical volume you made.
 - And expand it to 150MB.
 - You need to grow both the logical volume,
 - as well as the file system.