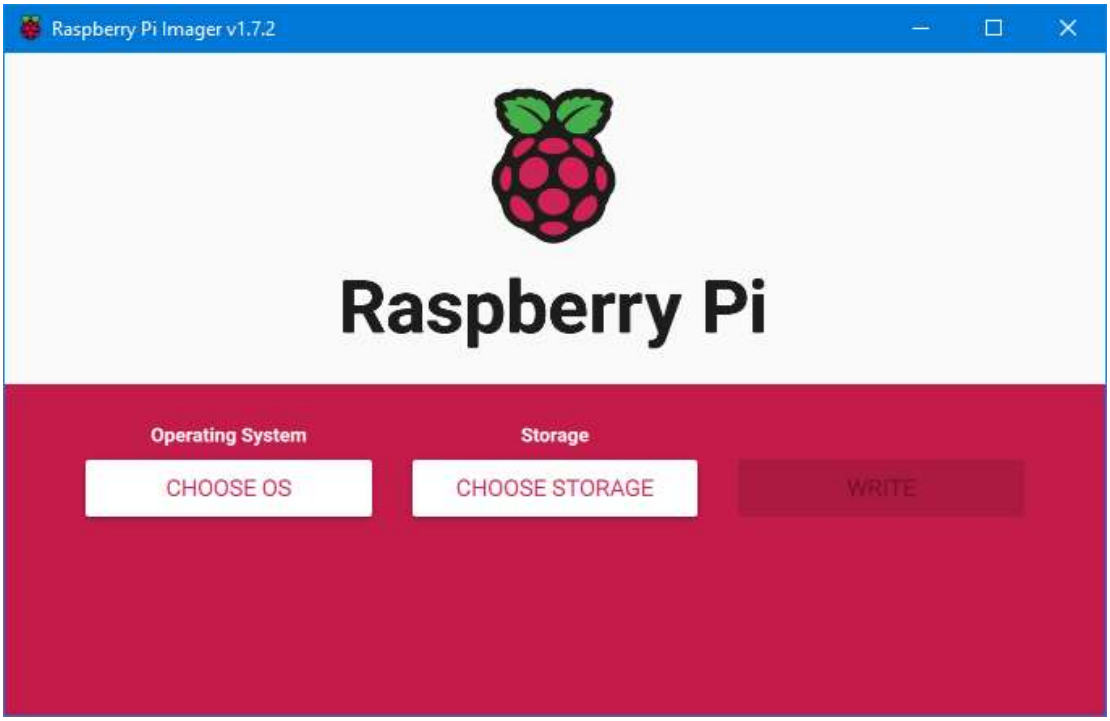
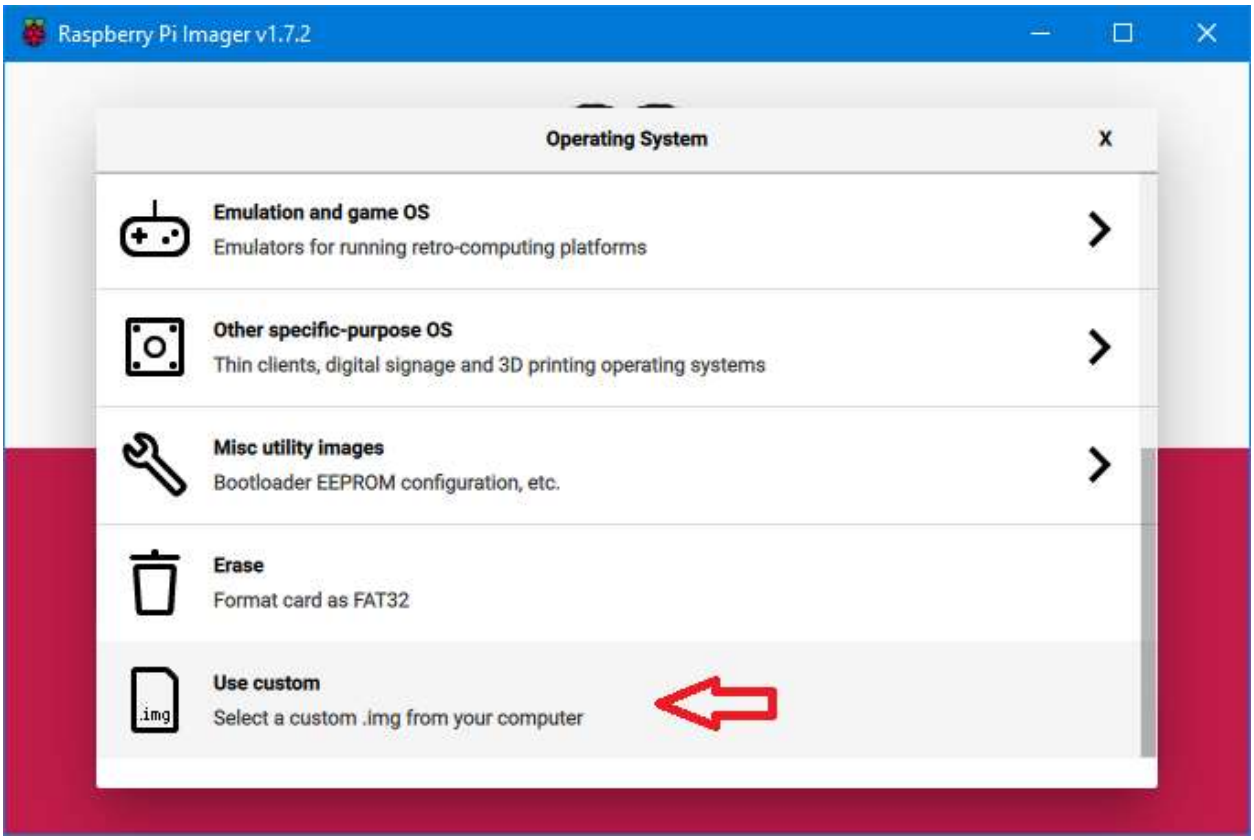
INSTALL PIKVM ON RASPBIAN BULLSEYE by @srepac

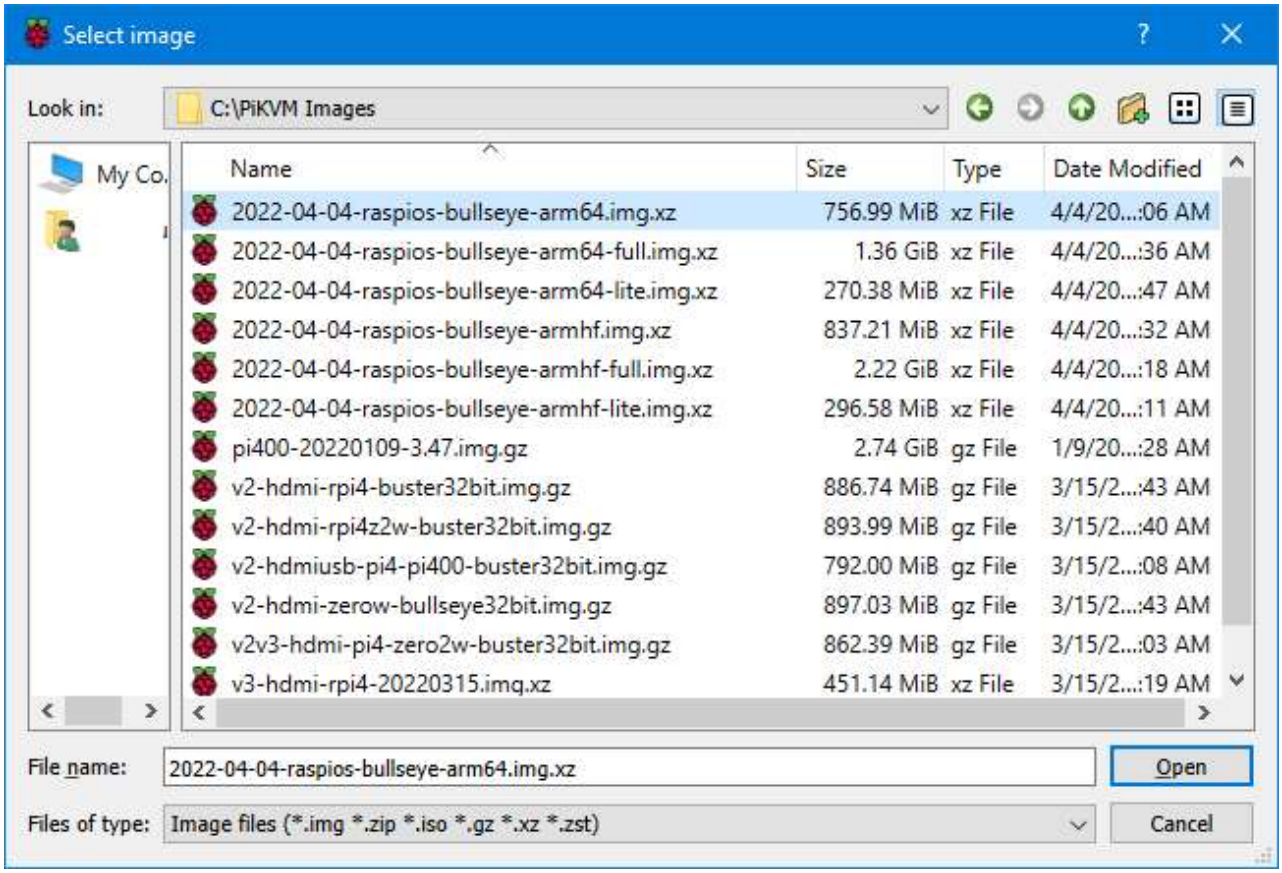
1. Download the 64-bit desktop bullseye image and run Raspberry Pi Imager. Click **Choose OS**.



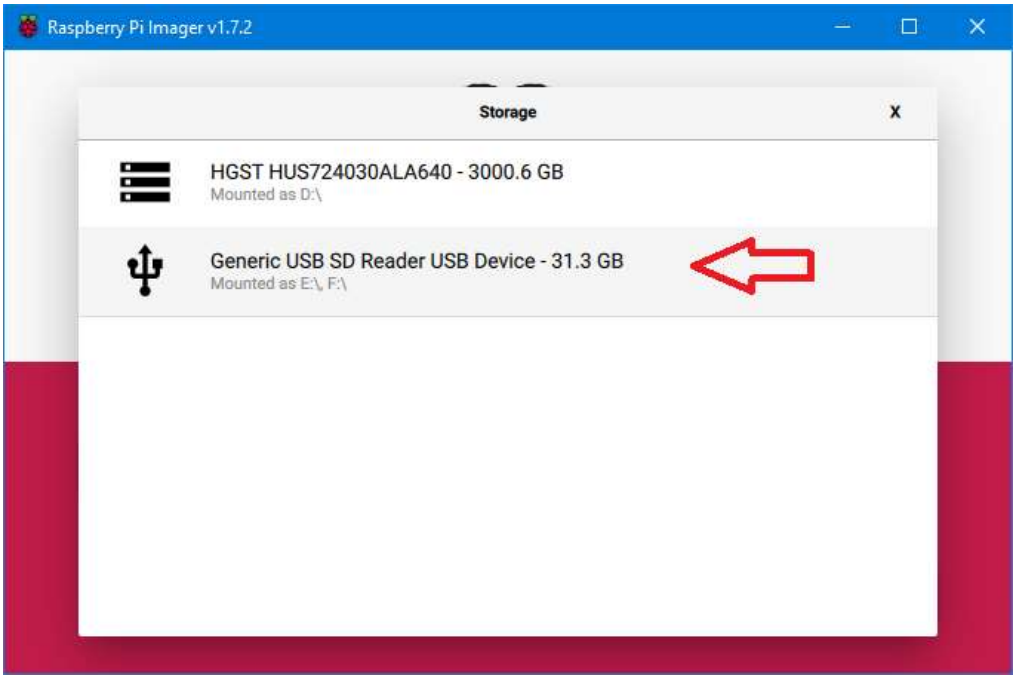
2. Click **Use Custom** to use the downloaded 64-bit raspbian bullseye desktop image.



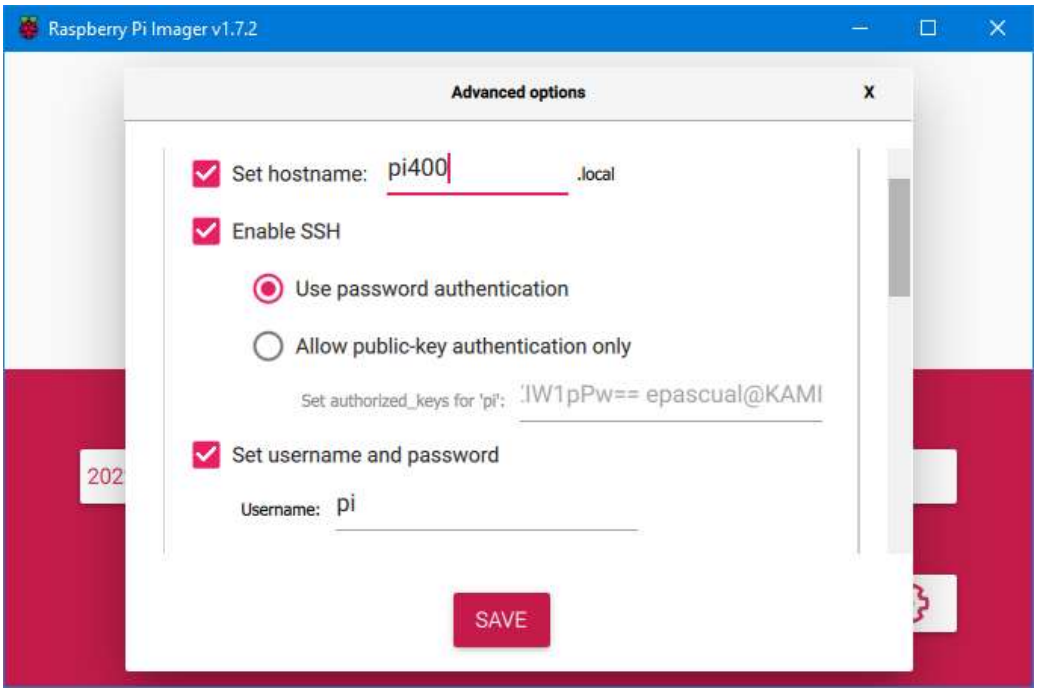
3. Browse for and select the correct image name then click **Open**.

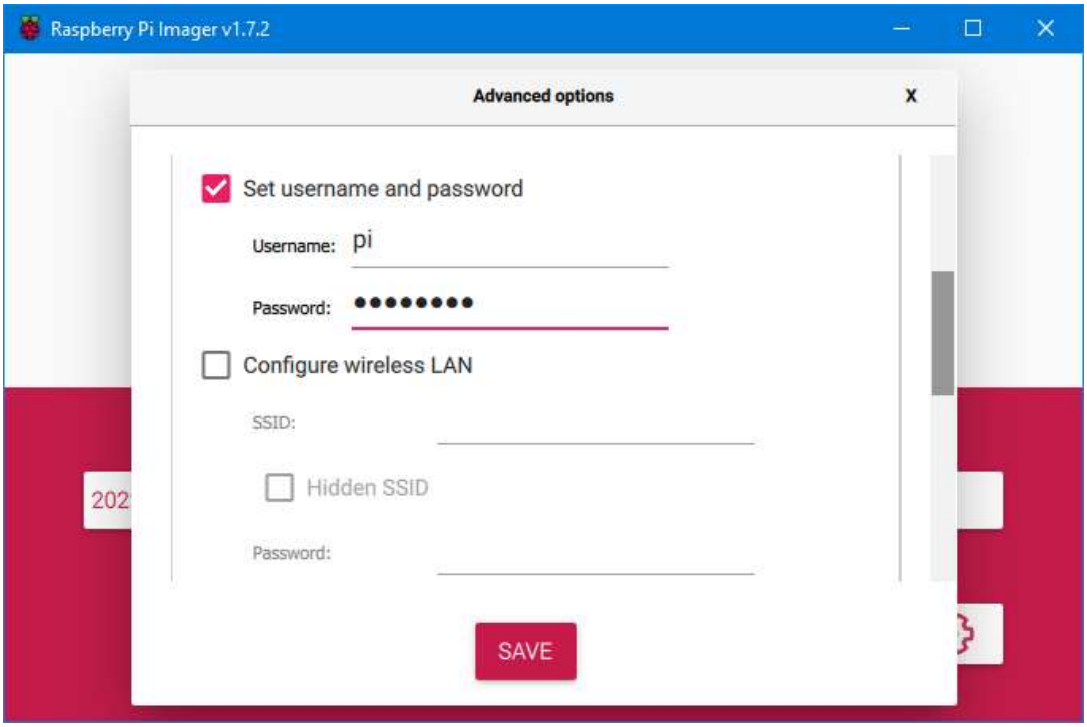


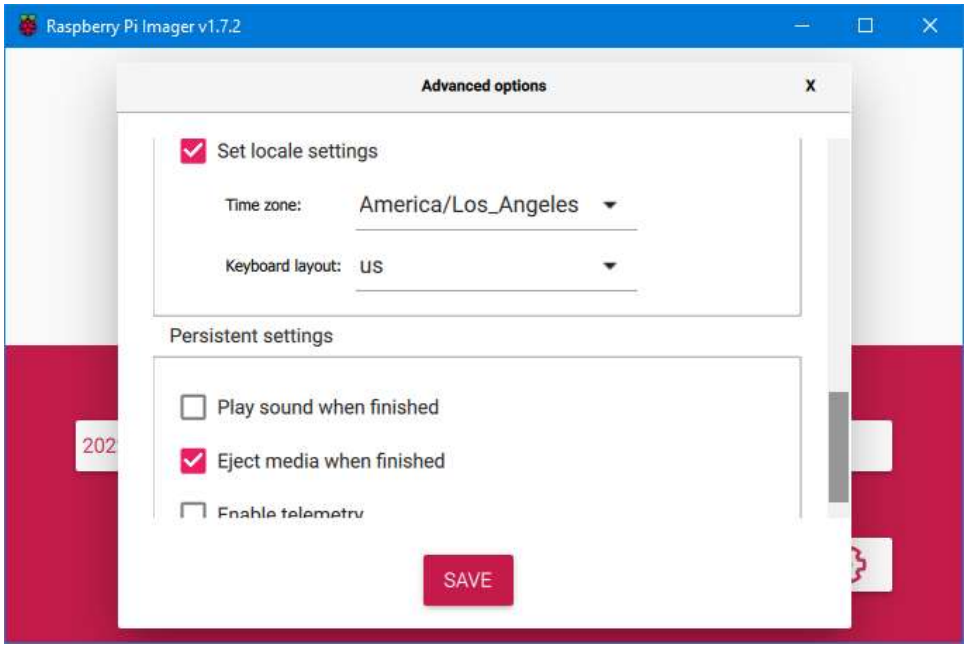
4. **Choose Storage** -> Select the correct USB SD Reader.



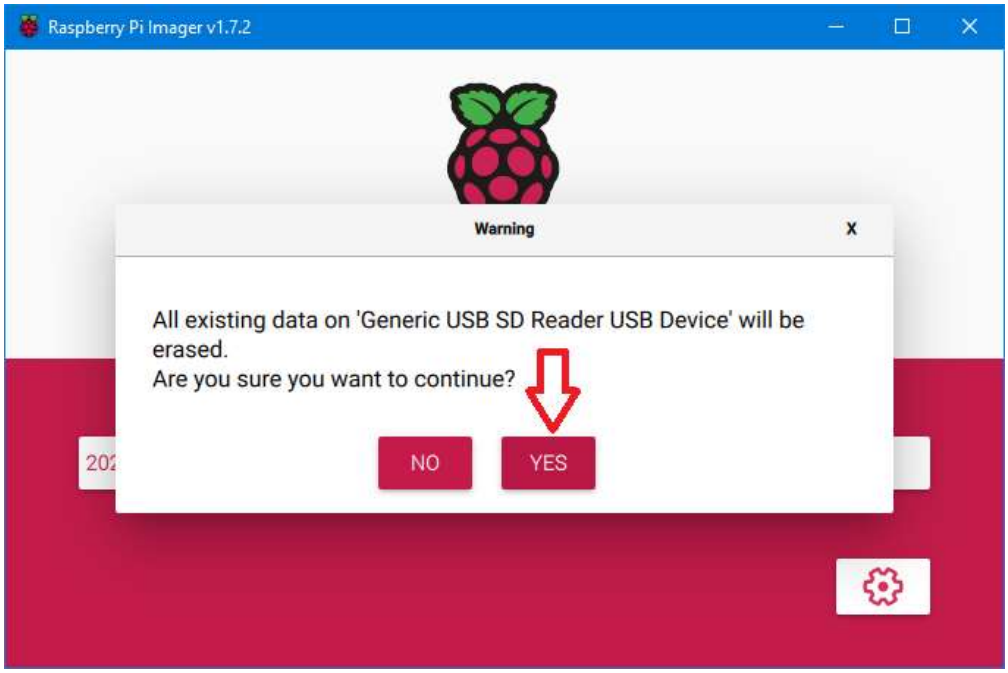
5. Click the **Gear** icon to open the Advanced options (NOTE: You can also use CTRL+SHIFT+X to open advanced settings). Make sure to select **For this session only** and set hostname, enable SSH, create pi username and password and lastly, set timezone.


Raspberry Pi Imager v1.7.2
Advanced options
X
Set hostname: pi400 .local
Enable SSH
Use password authentication
Allow public-key authentication only
Set authorized_keys for 'pi': IW1pPw== epascual@KAMI
Set username and password
Username: pi
SAVE

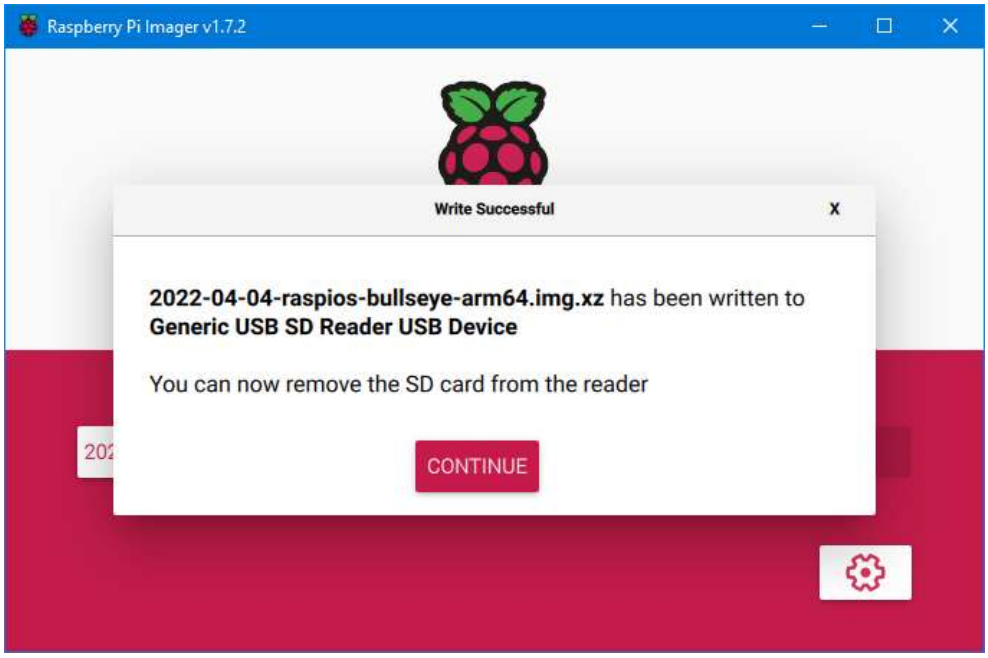

Raspberry Pi Imager v1.7.2
Advanced options
X
Set username and password
Username: pi
Password: ••••••••
Configure wireless LAN
SSID:
Hidden SSID
Password:
SAVE



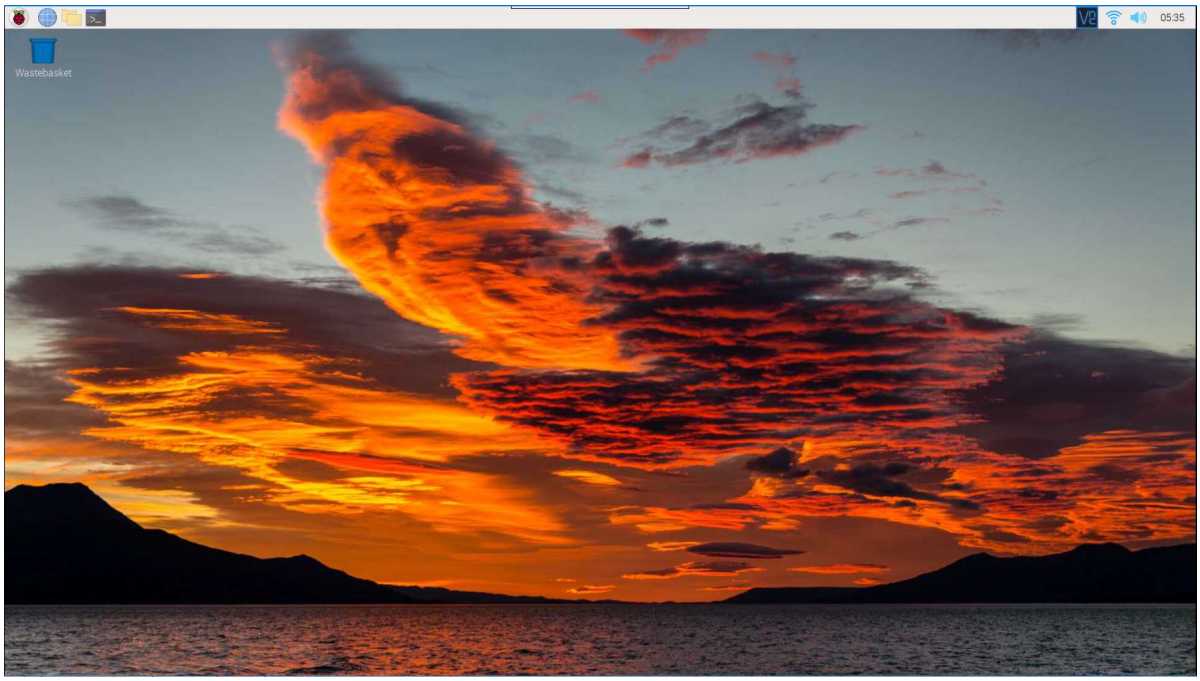
6. Once you are ready to write the image, click on **Write** and click **YES** to continue.



7. Wait until the image write to SD card completes.

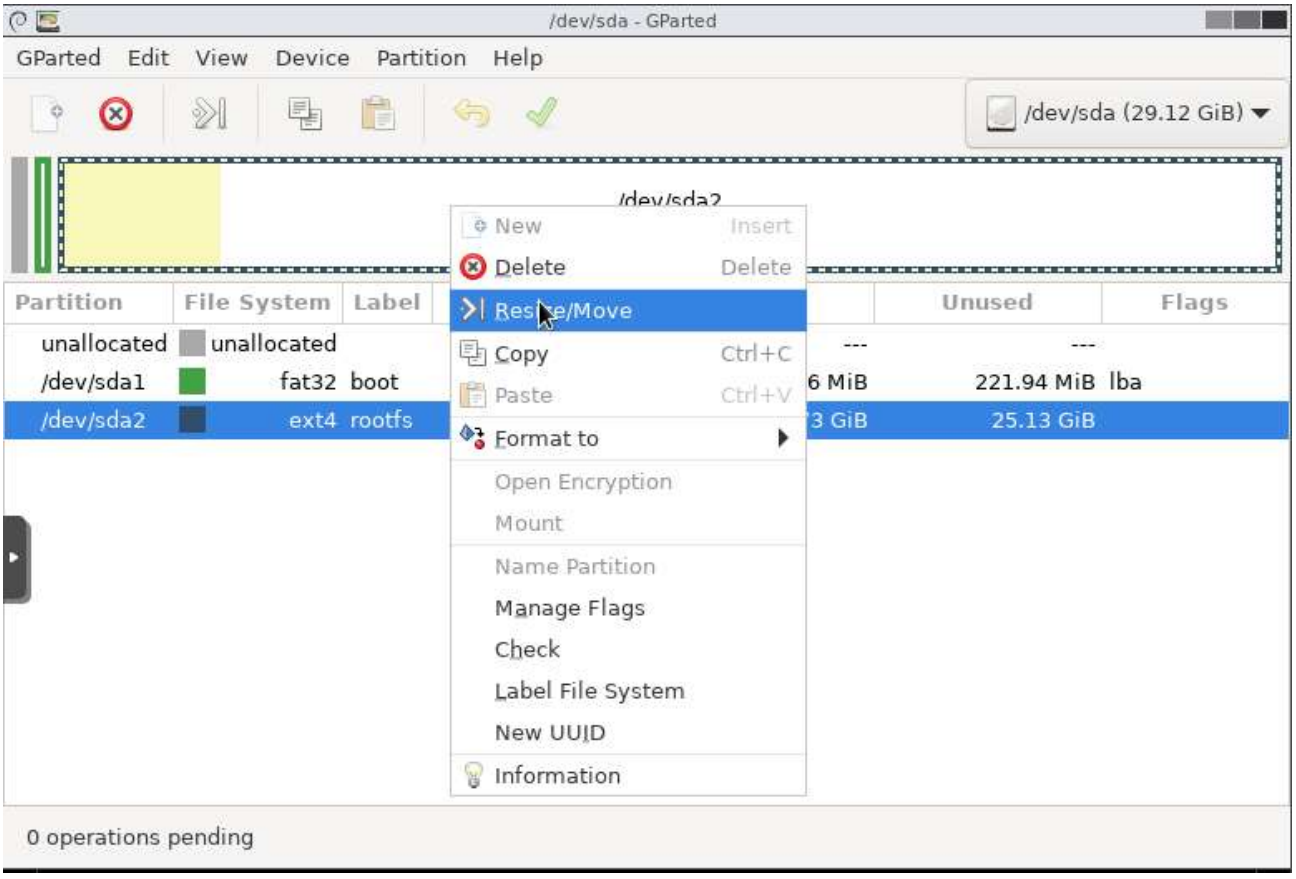


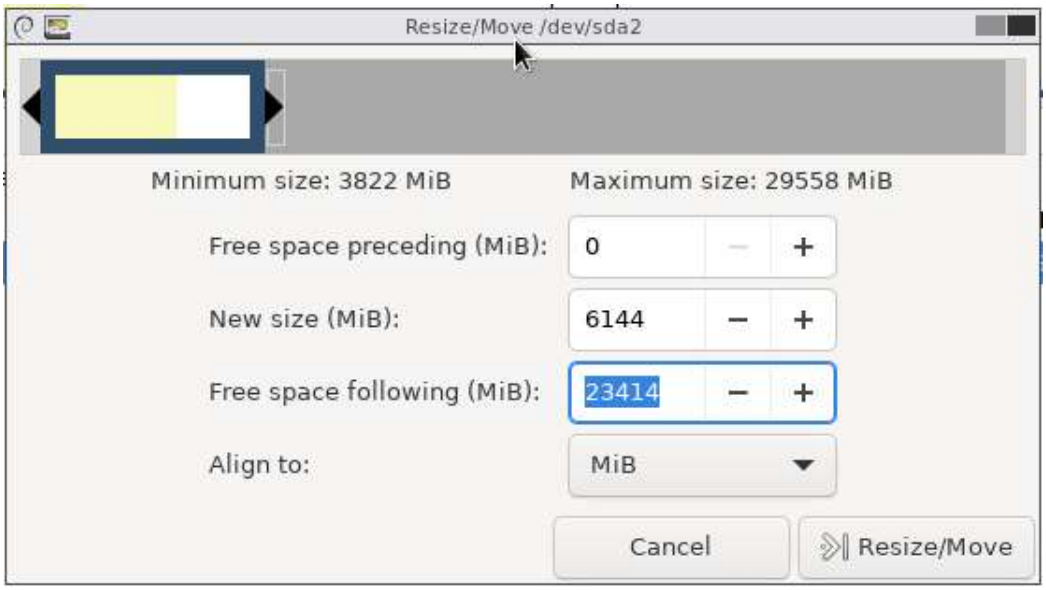
Put the SD card into your Pi and boot it up and let it finish expanding the 2nd partition to use the rest of the SD card. HINT: It should automatically login the pi user you created during Raspbian Pi Imager stage. Power off Pi then remove SD card and proceed to resize the SD card to create a new MSD partition.

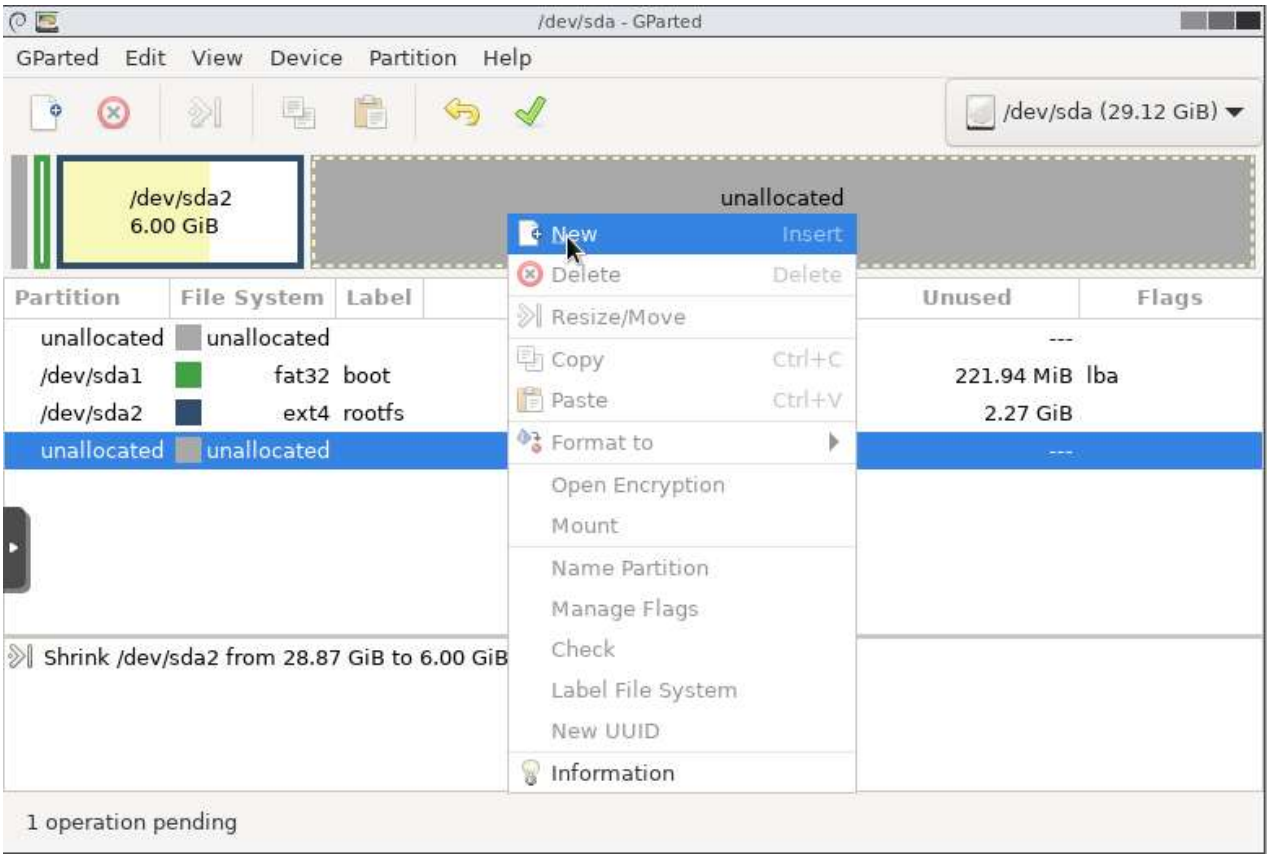


USE GPARTED ON A LINUX HOST TO RESIZE THE PI SD CARD

1. Resize partition 2 (/ root partition). In my example, I resized root partition to 6GB and then created 3rd partition for the remaining space for the MSD.

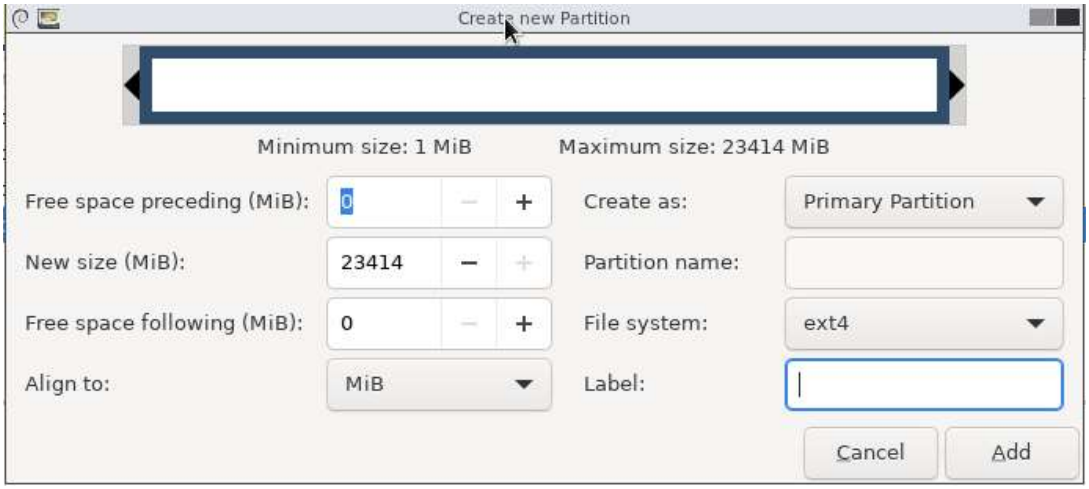



Resize/Move /dev/sda2
Minimum size: 3822 MiB
Maximum size: 29558 MiB
Free space preceding (MiB):
0
New size (MiB):
6144
Free space following (MiB):
23414
Align to:
MiB
Cancel
Resize/Move

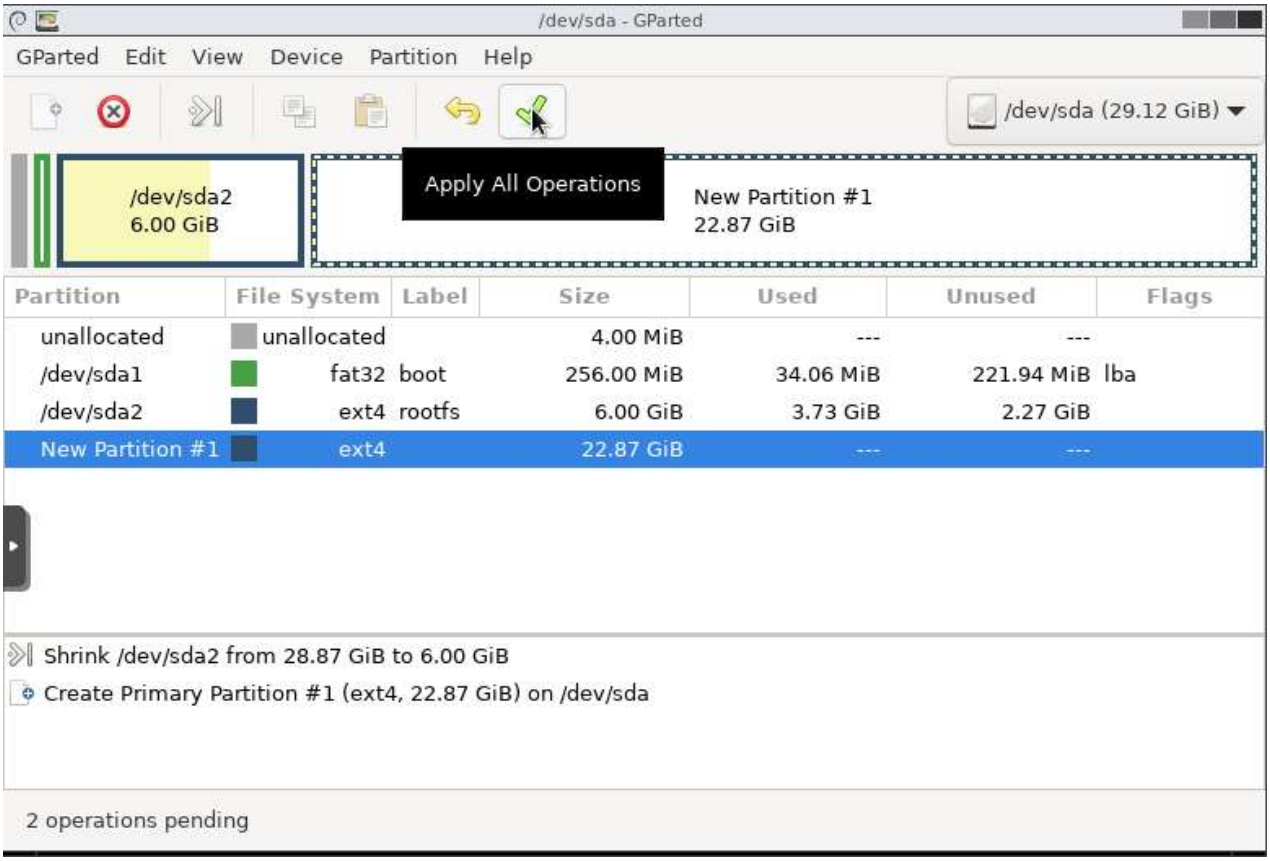

/dev/sda - GParted
GParted
Edit
View
Device
Partition
Help
/dev/sda (29.12 GiB)
/dev/sda2
6.00 GiB
unallocated
New
Insert
Delete
Delete
Resize/Move
Copy
Ctrl+C
Paste
Ctrl+V
Format to
Open Encryption
Mount
Name Partition
Manage Flags
Check
Label File System
New UUID
Information
Partition
File System
Label
Unused
Flags
unallocated
unallocated

/dev/sda1
fat32
boot
221.94 MiB
lba
/dev/sda2
ext4
rootfs
2.27 GiB
unallocated
unallocated

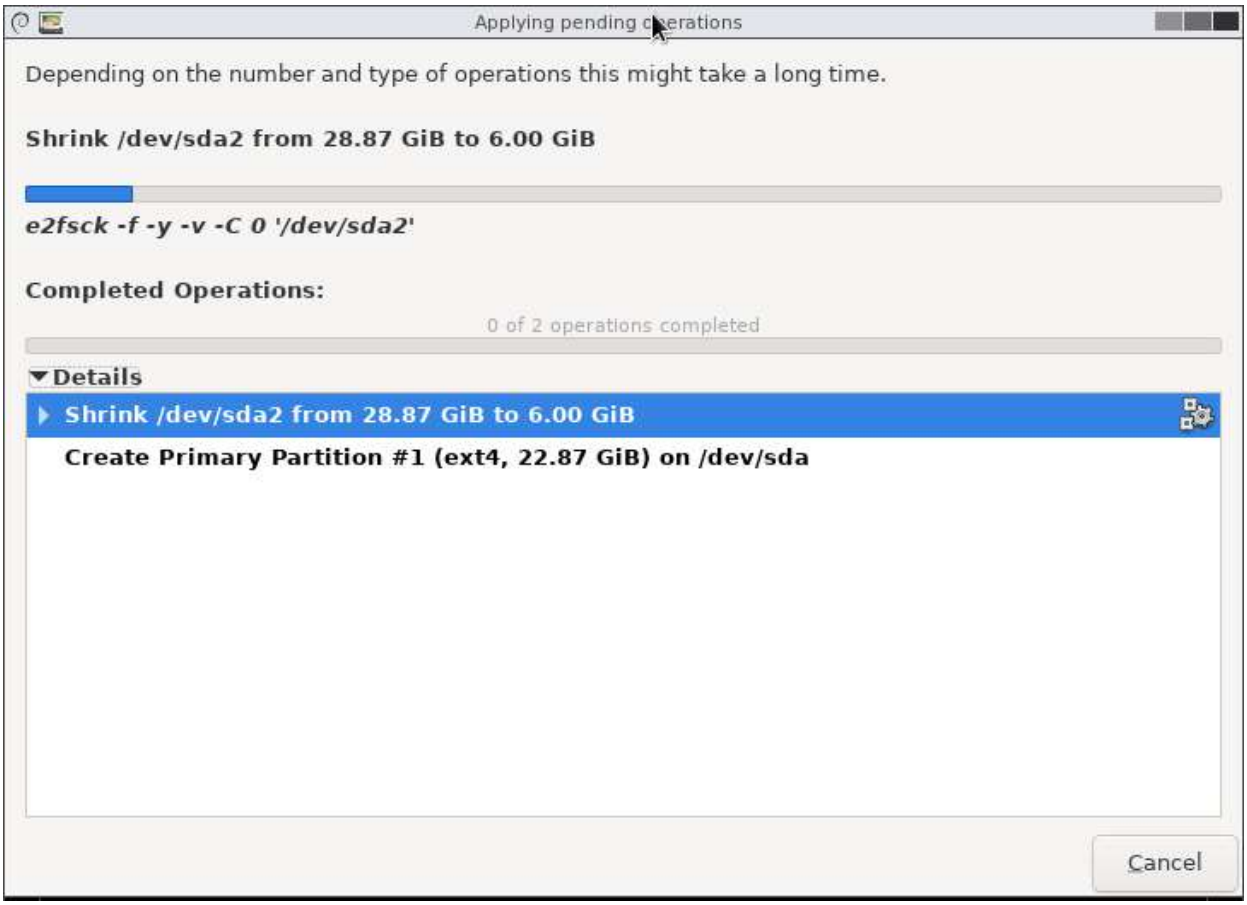
Shrink /dev/sda2 from 28.87 GiB to 6.00 GiB
1 operation pending

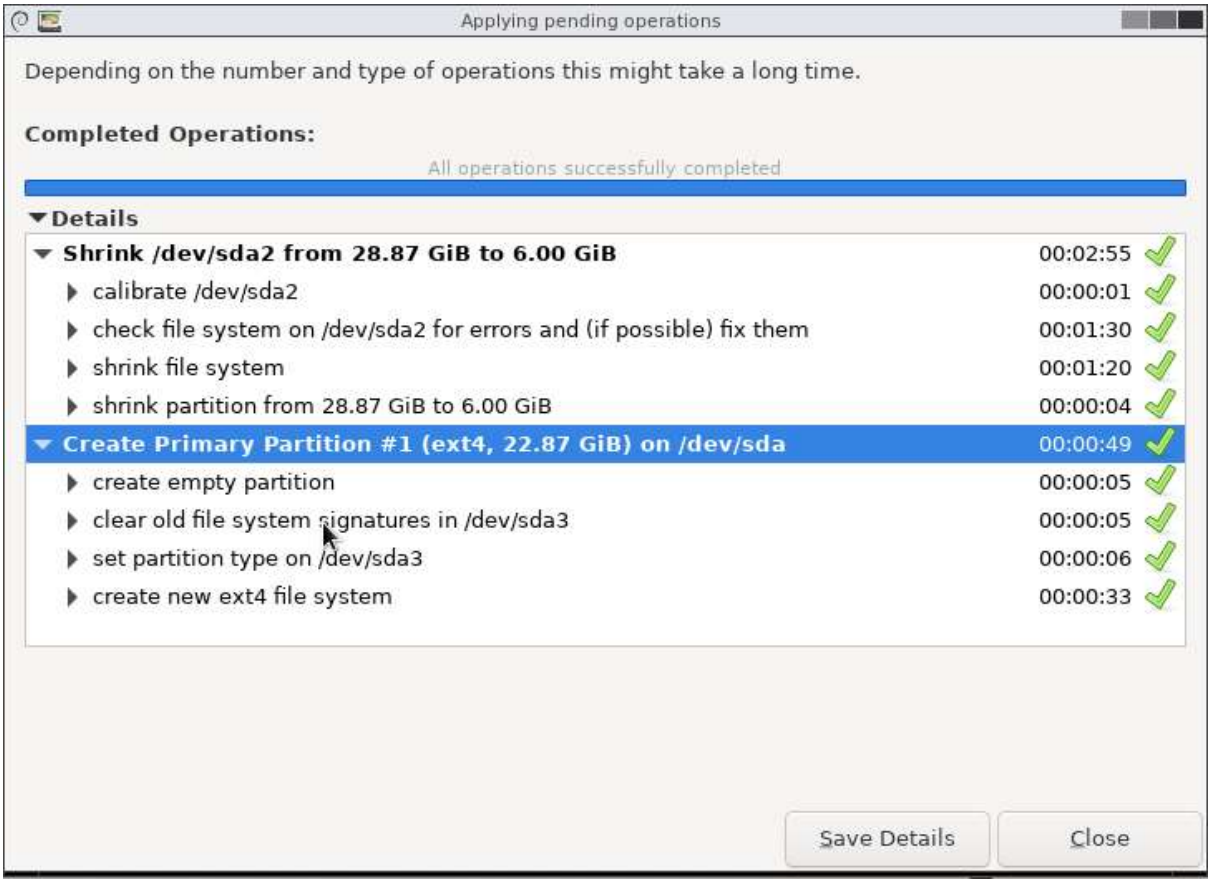


Right-click the New Parition #1 and **format to ext4** then **Apply All Operations**.



DO NOT INTERRUPT THE PENDING OPERATIONS UNTIL IT IS ALL SUCCESSFUL.


Applying pending operations
Depending on the number and type of operations this might take a long time.
Shrink /dev/sda2 from 28.87 GiB to 6.00 GiB
e2fsck -f -y -v -C 0 '/dev/sda2'
Completed Operations:
0 of 2 operations completed
Details
Shrink /dev/sda2 from 28.87 GiB to 6.00 GiB
Create Primary Partition #1 (ext4, 22.87 GiB) on /dev/sda
Cancel


Applying pending operations
Depending on the number and type of operations this might take a long time.
Completed Operations:
All operations successfully completed
Details
Shrink /dev/sda2 from 28.87 GiB to 6.00 GiB 00:02:55
calibrate /dev/sda2 00:00:01
check file system on /dev/sda2 for errors and (if possible) fix them 00:01:30
shrink file system 00:01:20
shrink partition from 28.87 GiB to 6.00 GiB 00:00:04
Create Primary Partition #1 (ext4, 22.87 GiB) on /dev/sda 00:00:49
create empty partition 00:00:05
clear old file system signatures in /dev/sda3 00:00:05
set partition type on /dev/sda3 00:00:06
create new ext4 file system 00:00:33
Save Details
Close

Remove SD card from the Linux system you used above and boot the Pi with the newly resized/partitioned SD card. Your root partition should now be smaller than before.

Next step is to create /var/lib/kvmd/msd mountpoint and update /etc/fstab so that it mounts automatically. Add the following entry in /etc/fstab.

```
/dev/mmcblk0p3  /var/lib/kvmd/msd   ext4  nodev,nosuid,noexec,ro,errors=remount-ro,data=journal,X-kvmd.otgmsd-root=/var/lib/kvmd/msd,X-kvmd.otgmsd-user=kvmd  0  0
```

```
root@pi400:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       5.9G  3.4G  2.3G  61% /
devtmpfs        1.7G     0  1.7G   0% /dev
tmpfs           1.8G  1.5M  1.8G   1% /dev/shm
tmpfs           724M  1.4M  723M   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.8G  8.0K  1.8G   1% /var/lib/misc
/dev/mmcblk0p3   23G   19G  3.1G  86% /var/lib/kvmd/msd
/dev/mmcblk0p1  253M   31M  222M  12% /boot
tmpfs           362M   20K  362M   1% /run/user/1000
tmpfs           362M   12K  362M   1% /run/user/0
```

INSTALL PIKVM RASPBIAN USING DEB PACKAGES

Janus (h264/webrtc), ustreamer (vid capture), kvmd-webterm (ttyd), kvmd-platform and main code kvmd 3.47

1. Update running kernel to 5.15.x by running **rpi-update** to use the 5.15.x kernel. When asked to reboot, please do so.

2. Download and run the installer only after running

```
sudo su -

wget https://kvmnerds.com/PiKVM/TESTING/pikvm-raspbian.sh

chmod +x pikvm-raspbian.sh

./pikvm-raspbian.sh
```

SAMPLE OUTPUT:

```
root@pi400:~ # ./pikvm-raspbian.sh
Running new @srepac installer version 1.3 that uses deb packages

*** ONLY Show commands to run
+ OSID [ debian ] is supported by installer.
+ Kernel version 5.15.32-v8+ ... OK
+ /etc/apt/sources.list.d/pikvm-raspbian.list already exists.

-> Getting list of available/installed janus, kvmd, and ustreamer packages...
No janus, kvmd, or ustreamer packages currently installed.

Auto setting platform for Pi 400

Choose installed oled screen:

  1 - 128x32 (default)
  2 - 128x32 flipped 180 degrees
```

```
  3 - 128x64
  4 - none
Please type [1-4]: 4

Platform selected -> kvmd-platform-v2-hdmiusb-rpi4

USB HDMI Capture device selected is supported.

-> Install instructions:
 platform:  kvmd-platform-v2-hdmiusb-rpi4
 oled:      none
 fan:       none
 model:     400
 board:     rpi4
 ARCH:      aarch64
 OS bits:   64bit

-> Copy/Paste below commands to install PiKVM on your Debian-based system manually.

apt install -y ustreamer-64bit kvmd-webterm-64bit kvmd-platform-v2-hdmiusb-rpi4 janus-64bit  kvmd-
raspbian

*** NOTE:  If you want the script to run the install command, then add -f option.
```

3. When you are ready to install, copy/paste the command generated by the script.

```
root@pi400:~ # apt install -y ustreamer-64bit kvmd-webterm-64bit kvmd-platform-v2-hdmiusb-rpi4
janus-64bit  kvmd-raspbian

[...]

Job for kvmd-otg.service failed because the control process exited with error code.
See "systemctl status kvmd-otg.service" and "journalctl -xe" for details.
lrwxrwxrwx 1 root root 6 Apr 26 04:56 /dev/kvmd-video -> video0

You should see devices for keyboard, mouse, and video.

Point a browser to https://pi400
If it doesn't work, then reboot one last time.
Please make sure kvmd services are running after reboot.

  var-lib-kvmd-msd.mount
loaded active mounted   /var/lib/kvmd/msd
  kvmd-janus-static.service
loaded active running   PiKVM - Janus WebRTC Gateway (Static Config)
  kvmd-nginx.service
loaded active running   PiKVM - HTTP entrypoint
● kvmd-otg.service
loaded failed failed    PiKVM - OTG setup
  kvmd-webterm.service
loaded active running   Pi-KVM - Web terminal (ttyd)
  kvmd.service
loaded active running   PiKVM - The main daemon
+ rw
/usr/bin/kvmd-gencert: line 50: rw: command not found
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for dbus (1.12.20-2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rpi1+deb11u2) ...
```

As you can see from the excerpt above, the kvmd-otg service failed because that piece requires reboot in order for the changes to /boot/config.txt to allow OTG peripherals.

4. Reboot your raspbian pikvm in order to resolve pending issues. At next boot, point a browser to https://<hostname>. In my example, I called my pikvm pi400, so my webui is located at https://pi400/

5. Lastly, run **pikvm-info** script to see useful troubleshooting information/packages installed.

```
root@pi400:~ # pikvm-info
 05:59:54 up 26 min,  4 users,  load average: 0.14, 0.15, 0.17
Host OS:  Debian GNU/Linux 5.15.32-v8+  aarch64
# Raspberry Pi 400 Rev 1.0 4GB

CPU temp: 37.48'C
GPU temp: 38.4'C

Throttled flags: 0x0

Throttled now:  no
Throttled past: no

Undervoltage now:  no
Undervoltage past: no

Frequency capped now:  no
Frequency capped past: no

Version                 Package-Name
----------------------  ----------------------------
1.2rpt8                 bluez-firmware
1:20210315rpt5          firmware-atheros
1:20210315rpt5          firmware-brcm80211
1:20210315rpt5          firmware-libertas
1:20210315rpt5          firmware-misc-nonfree
1:20210315rpt5          firmware-realtek
0.12.0                  janus-64bit
3.47                    kvmd-platform-v2-hdmiusb-rpi4
3.47                    kvmd-raspbian
1.6.3                   kvmd-webterm-64bit
1.18.0-6.1              nginx
1.18.0-6.1              nginx-common
1.18.0-6.1              nginx-core
2021.1.1+rpt1           raspberrypi-archive-keyring
1:1.20220331-1          raspberrypi-bootloader
1:1.20220331-1          raspberrypi-kernel
1.3.3                   raspberrypi-net-mods
20220224                raspberrypi-sys-mods
1.20220302              raspberrypi-ui-mods
5.3                     ustreamer-64bit
2020.04.29-2            wireless-regdb
30~pre9-13.1            wireless-tools
```

Want read-only filesystem? Please follow this article below.

https://medium.com/swlh/make-your-raspberry-pi-file-system-read-only-raspbian-buster-c558694de79