

# LUKS Filesystems

When setting up a new Linux installation, I decided to take advantage of LUKS (Linux Unified Key System) to encrypt the data on my device. The setup is pretty turn-key, most Linux distributions offer an option for guided LUKS setup during installing. When rebooting my device you get a prompt to enter a password and the storage is decrypted.

I recently decided to retire my internal SSD to use it as external storage instead. Before formatting the SSD, I had hoped to mount it to a new Kubuntu 22.04 installation and copy over any files I wanted to keep. This was the first time I had ever tried mounting a LUKS filesystem manually, and it took a bit of messing around. Mostly I think this was because both SSDs were full installations of Kubuntu, so the partitions happened to be named the same - they were both created with guided setup for LUKS.

For some context, here's output of `vgdisplay` and all currently mapped devices on my system before making any modifications. It may be useful to compare this against later output to help see what's happening.

```
sudo vgdisplay
[sudo] password for kapper:

--- Volume group ---
VG Name          vgkubuntu
System ID
Format           lvm2
Metadata Areas    1
Metadata Sequence No  3
VG Access         read/write
VG Status         resizable
MAX LV           0
Cur LV           2
Open LV           2
Max PV            0
Cur PV           1
Act PV            1
VG Size           <1.82 TiB
PE Size           4.00 MiB
Total PE          476372
Alloc PE / Size   476372 / <1.82 TiB
```

```
Free PE / Size    0 / 0
VG UUID           uACwRN-syEc-S99E-tznk-3hYD-062d-oWVbKv
```

```
ls /dev/mapper/
```

```
control nvme0n1p3_crypt vgkubuntu-root vgkubuntu-swap_1
```

When initially plugging the device in, we run `lsblk` and check the output to find the name of the device we want to access is `sda3`. My system automatically mounted `sda2` which is `/boot` - not what we want, but understandable since the other partition was encrypted and not immediately available for mounting.

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
loop0	7:0	0	4K	1	loop	/snap/bare/5
loop1	7:1	0	61.9M	1	loop	/snap/core20/1405
loop2	7:2	0	55.6M	1	loop	/snap/core18/2538
loop3	7:3	0	62M	1	loop	/snap/core20/1587
loop4	7:4	0	163.3M	1	loop	/snap/firefox/1635
loop5	7:5	0	163.3M	1	loop	/snap/firefox/1670
loop6	7:6	0	400.8M	1	loop	/snap/gnome-3-38-2004/112
loop7	7:7	0	164.8M	1	loop	/snap/gnome-3-28-1804/161
loop8	7:8	0	248.8M	1	loop	/snap/gnome-3-38-2004/99
loop9	7:9	0	81.3M	1	loop	/snap/gtk-common-themes/1534
loop10	7:10	0	91.7M	1	loop	/snap/gtk-common-themes/1535
loop11	7:11	0	43.6M	1	loop	/snap/snapd/15177
loop12	7:12	0	47M	1	loop	/snap/snapd/16292
loop13	7:13	0	169.4M	1	loop	/snap/spotify/60
sda	8:0	0	931.5G	0	disk	
├─sda1	8:1	0	512M	0	part	
├─sda2	8:2	0	732M	0	part	/media/kapper/20e58d66-eea7-4c73-b40f-b293f9a468da
└─sda3	8:3	0	930.3G	0	part	
└─luks-f08f6bfc-fd1f-49cc-8882-c566f19189a3						
253:3	0	930.3G	0	crypt		
mmcblk0	179:0	0	3.7G	0	disk	
└─mmcblk0p1	179:1	0	3.7G	0	part	
nvme0n1	259:0	0	1.8T	0	disk	
├─nvme0n1p1	259:1	0	512M	0	part	/boot/efi
├─nvme0n1p2	259:2	0	1.7G	0	part	/boot

```

└─nvme0n1p3                259:3   0  1.8T 0 part
  └─nvme0n1p3_crypt         253:0   0  1.8T 0 crypt
    ├─vgkubuntu-root        253:1   0  1.8T 0 lvm  /
    └─vgkubuntu-swap_1      253:2   0  976M 0 lvm  [SWAP]

```

Now we can use `cryptsetup` to open the LUKS device and map the unencrypted data to a device. We choose the name of the mapped device - so you can change the `ssd` name below to be anything you want. The mapped device will be created in `/dev/mapper/`.

```

sudo cryptsetup luksOpen /dev/sda3 ssd-kubuntu
Enter passphrase for /dev/sda3:

```

Note that if your current system is using LUKS, some mappings may already exist - you should provide unique names for each device.

```

ls /dev/mapper/

control nvme0n1p3_crypt ssd-kubuntu ssd--kubuntu-root ssd--kubuntu-swap_1 vgkubuntu-root vgkubuntu-
swap_1

```

If you did not provide a unique name or if the devices default name happened to collide with an existing mapped device, you can change it with `vgrename`. This command needs to reference a UUID to rename the mapped device though, so we first use `vgdisplay` to get this information.

```

sudo vgdisplay
[sudo] password for kapper:

WARNING: VG name vgkubuntu is used by VGs 3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX and uACwRN-syEc-
S99E-tznk-3hYD-062d-oWVbKv.
Fix duplicate VG names with vgrename uuid, a device filter, or system IDs.

--- Volume group ---
VG Name          vgkubuntu
System ID
Format           lvm2
Metadata Areas    1
Metadata Sequence No 3
VG Access         read/write
VG Status         resizable
MAX LV           0
Cur LV           2

```

```
Open LV          0
Max PV           0
Cur PV          1
Act PV           1
VG Size          <930.28 GiB
PE Size          4.00 MiB
Total PE         238151
Alloc PE / Size  238151 / <930.28 GiB
Free PE / Size   0 / 0
VG UUID          3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX
```

--- Volume group ---

```
VG Name          vgkubuntu
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          2
Max PV           0
Cur PV          1
Act PV           1
VG Size          <1.82 TiB
PE Size          4.00 MiB
Total PE         476372
Alloc PE / Size  476372 / <1.82 TiB
Free PE / Size   0 / 0
VG UUID          uACwRN-syEc-S99E-tznk-3hYD-062d-oWVbKv
```

The two devices above may have the same name but they provide unique UUIDs. The device I want to rename is 1TB, so here I'll use the `3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX` UUID to rename it. You may get a warning here. The warning is referring to the device we just renamed - notice the matching UUIDs. We just need to update our VG devices with the new name, which we will do in the next step.

```
sudo vgrename 3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX ssd-kubuntu
```

WARNING: VG name vgkubuntu is used by VGs 3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX and uACwRN-syEc-

```
S99E-tznk-3hYD-062d-oWVbKv.
```

Fix duplicate VG names with `vgrename` `uuid`, a device filter, or system IDs.

Processing VG `vgkubuntu` because of matching UUID `3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX`

Volume group `"3Ab6YC-AsQ0-BKfF-F5QA-OXb3-Hvlu-hsLuyX"` successfully renamed to `"ssd-kubuntu"`

Activate the devices - this will resolve the warning from the previous step. If you still get a warning here, you do have two devices with the same name. Check the output of `vgdisplay` to determine which device needs to be renamed.

```
sudo vgchange -ay
```

```
2 logical volume(s) in volume group "ssd-kubuntu" now active
```

```
2 logical volume(s) in volume group "vgkubuntu" now active
```

Check the devices were activated successfully

```
sudo lvscan
```

```
ACTIVE      '/dev/ssd-kubuntu/root' [929.32 GiB] inherit
ACTIVE      '/dev/ssd-kubuntu/swap_1' [976.00 MiB] inherit
ACTIVE      '/dev/vgkubuntu/root' [<1.82 TiB] inherit
ACTIVE      '/dev/vgkubuntu/swap_1' [976.00 MiB] inherit
```

See information for the activated VG devices -

```
sudo lvdisplay
```

```
--- Logical volume ---
```

```
LV Path      /dev/ssd-kubuntu/root
LV Name      root
VG Name      ssd-kubuntu
LV UUID      Vgijki-nRap-tE3q-etn0-HKJz-2g6V-9TXg19
LV Write Access   read/write
LV Creation host, time kubuntu, 2021-12-06 09:26:51 -0500
LV Status     available
# open       0
LV Size      929.32 GiB
Current LE   237907
Segments     1
Allocation   inherit
Read ahead sectors   auto
```

- currently set to 256  
Block device 253:4

--- Logical volume ---

LV Path /dev/ssd-kubuntu/swap\_1  
LV Name swap\_1  
VG Name ssd-kubuntu  
LV UUID feQi9r-QVBH-ukjv-sE6P-jgzX-x46p-ahxPz9  
LV Write Access read/write  
LV Creation host, time kubuntu, 2021-12-06 09:26:52 -0500  
LV Status available  
# open 0  
LV Size 976.00 MiB  
Current LE 244  
Segments 1  
Allocation inherit  
Read ahead sectors auto  
- currently set to 256  
Block device 253:5

--- Logical volume ---

LV Path /dev/vgkubuntu/root  
LV Name root  
VG Name vgkubuntu  
LV UUID 00Zi9e-JF5h-WNZn-527p-Tfqq-RGc2-kRdtds  
LV Write Access read/write  
LV Creation host, time kubuntu, 2022-07-29 19:47:38 -0400  
LV Status available  
# open 1  
LV Size <1.82 TiB  
Current LE 476128  
Segments 1  
Allocation inherit  
Read ahead sectors auto  
- currently set to 256  
Block device 253:1

--- Logical volume ---

LV Path /dev/vgkubuntu/swap\_1  
LV Name swap\_1

VG Name	vgkubuntu
LV UUID	ATtx1E-9CDY-R349-pzqJ-f6id-RGd0-Zz136a
LV Write Access	read/write
LV Creation host, time	kubuntu, 2022-07-29 19:47:38 -0400
LV Status	available
# open	2
LV Size	976.00 MiB
Current LE	244
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	256
Block device	253:2

Finally, create the directory where you want to mount the device if it doesn't exist already, then mount the device.

It's likely that you can skip this step. If for some reason your device isn't automatically mounted, this is an example of how to mount a mapped device. On Kubuntu 22.04, my device was automatically mounted at `/media/kapper/174fdc5d-0e9b-4be2-aeaa-1c2fbfd65c28` and available to browse in Dolphin.

```
mkdir /mnt/ssd
sudo mount /dev/ssd-kubuntu/root /mnt/ssd/
ls /mnt/ssd/

bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
```

Done! You can now access the filesystem from your last LUKS installation and transfer any files you want to keep.

---

Revision #6

Created 13 July 2019 14:02:49 by Shaun Reed

Updated 14 August 2022 16:02:55 by Shaun Reed