www.qubes-os.org /doc/secondary-storage/

Secondary storage

4-4 minutes ⋮ 10/17/2024

Suppose you have a fast but small primary SSD and a large but slow secondary HDD. You want to store a subset of your app qubes on the HDD.

Instructions

Qubes 4.0 is more flexible than earlier versions about placing different VMs on different disks. For example, you can keep templates on one disk and app qubes on another, without messy symlinks.

These steps assume you have already created a separate volume group and thin pool (not thin volume) for your HDD. See also this example if you would like to create an encrypted LVM pool (but note you can use a single logical volume if preferred, and to use the `-T` option on `lvcreate` to specify it is thin). You can find the commands for this example applied to Qubes at the bottom of this R4.0 section.

First, collect some information in a dom0 terminal:

Take note of the VG and thin pool names for your HDD, then register it with Qubes:

```
# <pool_name> is a freely chosen pool name
# <vg_name> is LVM volume group name
# <thin_pool_name> is LVM thin pool name
qvm-pool --add <pool_name> lvm_thin -o
volume_group=<vg_name>,thin_pool=<thin_pool_name>,revisions_to_keep=2
```

Now, you can create qubes in that pool:

```
qvm-create -P <pool_name> --label red <vmname>
```

It isn't possible to directly migrate an existing qube to the new pool, but you can clone it there, then remove the old one:

```
qvm-clone -P <pool_name> <sourceVMname> <cloneVMname>
qvm-remove <sourceVMname>
```

If that was a template, or other qube referenced elsewhere (NetVM or such), you will need to adjust those references manually after moving. For example:

```
qvm-prefs <appvmname_based_on_old_template> template <new_template_name>
```

In theory, you can still use file-based disk images ("file" pool driver), but it lacks some features such as you won't be able to do backups without shutting down the qube.

Example HDD setup

Assuming the secondary hard disk is at /dev/sdb (it will be completely erased), you can set it up for encryption by doing in a dom0 terminal (use the same passphrase as the main Qubes disk to avoid a second password prompt at boot):

```
sudo cryptsetup luksFormat --hash=sha512 --key-size=512 --cipher=aes-xts-
plain64 --verify-passphrase /dev/sdb
sudo blkid /dev/sdb
```

Note the device's UUID (in this example "b209…"), we will use it as its luks name for auto-mounting at boot, by doing:

And adding this line (change both "b209…" for your device's UUID from blkid) to crypttab:

```
luks-b20975aa-8318-433d-8508-6c23982c6cde
UUID=b20975aa-8318-433d-8508-6c23982c6cde none
```

Reboot the computer so the new luks device appears at /dev/mapper/luks-b209… and we can then create its pool, by doing this on a dom0 terminal (substitute the b209… UUIDs with yours):

First create the physical volume

```
sudo pvcreate /dev/mapper/luks-b20975aa-8318-433d-8508-6c23982c6cde
```

Then create the LVM volume group, we will use for example "qubes" as the :

```
sudo vgcreate qubes /dev/mapper/luks-b20975aa-8318-433d-8508-6c23982c6cde
```

And then use "poolhd0" as the (LVM thin pool name):

```
sudo lvcreate -T -n poolhd0 -l +100%FREE qubes
```

Finally we will tell Qubes to add a new pool on the just created thin pool

```
qvm-pool --add poolhd0_qubes lvm_thin -o
volume_group=qubes,thin_pool=poolhd0,revisions_to_keep=2
```

By default VMs will be created on the main Qubes disk (i.e. a small SSD), to create them on this secondary HDD do the following on a dom0 terminal:

```
qvm-create -P poolhd0_qubes --label red unstrusted-hdd
```