forum.qubes-os.org /t/really-disposable-ram-based-qubes/21532/159

Really disposable (RAM based) qubes - Community Guides - Qubes OS Forum

12-16 minutes

Small side note: If you have not yet made the decision to put your work on Microsoft's Github. i.e. codeberg.org 2 or Gitlab would be a good non-big-tech alternative.

Yeah, it is difficult:

- On one hand, we distrust all infrastructure, on the other - just because privacy is not a goal for Qubes OS, doesn't mean deliberately choosing a well-known privacy abusing corporation is fine. Regardless of that, all Qubes OS's code is already on GitHub.
- Gitlab does not let me one the website without JS. Still, Tor Project uses it.
- GitHub is still usable without JS (if one has been lucky to register when that was possible without JS-dependent captcha)
- GitHub has a CLI tool
- codeberg.org 5 looks interesting. I will definitely check it out. Thanks.

This is only on the dom0 terminal any plans to of making it an option on the qube-manager create qube option?

That is up to the developer of Qube Manager.

To launch a disposable vm directly without using the Dom0 terminal, you can create a shortcut in the taskbar: copy the @qubist script into the `/usr/local/bin/` folder:

```
sudo cp *the_script* /usr/local/bin/*the_script*
```

check that it is executable:

```
sudo chmod +x /usr/local/bin/*the_script*
```

then, you add a new launcher in your panel with the command, for example:

```
*the_script* -c firefox -p template=debian-12-dvm -p netvm=sys-firewall -p memory=400 2G -p label=red
```

you will just have to click on it to launch without going through dom0.

Shouldn't the memory argument be `-p memory=400M 2G` (adding the `M` after `400`)?

Ah nvm the script doesn't like `M`; the command is correctly formed.

No it isn't, there is no `M`, the memory is in MB by default. I just try with it and it doesn't work :

`qvm-prefs: error: Failed to parse property value as int`

`-p memory=400 -s 2G`

However, I would use the opposite:

`-p memory=2000 -s 400M`

Reason: Firefox needs more RAM than storage, so no need to take so much from dom0 (`-s`) but better take from Xen (`-p memory`). You can use less than 400M if you configure your disposable template to mount ~/.mozilla on tmpfs, and thus offload even that to Xen.

Would it be possible to make a template in which its goal is to create ram disposable qubes?
As in this template when using the App DVM launches a ram disposable qube. Is this possible?

RAM-based disposables are AppVMs (not DispVMs) cloned to a RAM mount point. The fact that they are based on disposable templates (which is a strange term, as it is neither a TemplateVM, nor disposable in any way) is just for conformity to the concept of conventional disposables with the idea to prevent mistakes.

I'm not fond of that naming convention either, and I think it leads to confusion. Sometimes someone has to specify something like "the template of the disposable template" to mean the actual templateVM.

And I've seen both disposable templates and named disposables named *-dvm, so when you see something like that in a menu it's tricky to know what you're getting. In the first case you could get either the disposable template as a (normal) AppVM, or a numbered disposable, depending. In the second, you get a disposable with a specific name. I personally name my disposable templates *-dvmt and named disposables *-disp. That still leaves the issue of knowing whether clicking on a *-dvmt menu item will open a numbered disposable or the disposable template itself.

(Now you CAN tell them apart, if you know the somewhat arcane menu convention…but you shouldn't need to know it.)

Been playing around with the script a bit and there seem to be some problems:

```
./RAM-only_DVM.sh -c xfce4-terminal -p template=debian-12-dvm-low-mem -p
memory=1500 -s 500
```

will make it crash (`qvm-clone: error: Got empty response from qubesd.`), while

```
./RAM-only_DVM.sh -c xfce4-terminal -p template=debian-12-dvm-low-mem -p
memory=1500 -s 500M
```

works. So there's an inconsistency: `M` can't be added for the `-p memory` option, but has to be added for the `-s` option when trying to specify megabytes. This is because `-s` is handed off to the `mount` command, which uses the common convention (M designates megabytes), while `-p memory` is handed off to `qvm-prefs`, which defaults to MB…it's really `qvm-prefs` that should accept values ending in `m` or `M`, but then again, it's the script that takes values for different program's parameters, so it could be argued that it should uniformize the formats through processing the input better.

Another example of input validation problems: if I put `-s 2.5` or `-s 2.5G`, then it will crash after having created the PID file and thus fail to start up again until the user manually deletes it. Again, one could argue that `mount` should handle these kinds of values better, but the script also should clean up after itself. I see that the cleanup function does include the `rm -f "${pidfile}"`, but since it's `mount` that exits with an error code it doesn't get called I think…probably something like `mount [...] || cleanup` would work here.

@Bearillo

There is this comment in the code:

```
# TODO: Validate size value
```

This means, the concerns regarding it have already been considered and will be addressed somehow. Meanwhile, simply don't use invalid input.

> something like `mount [...] || cleanup` would work here.

Perhaps using ERR trap might be more elegant as a whole.

And it's followed up with

```
# Show error message if size exceeds
# available memory
```

So I wasn't sure whether the number formatting issue was already known.

Perhaps, though `set -e` is still active at that point; I suggested `|| cleanup` due to it already being used at two other places in the `main` function.

I'd also suggest updating the command line example to reflect the new Whonix template naming convention (`workstation` is spelled out now).

Could I rather make a shortcut as a App VM entry? Would this Be possible?

> Could I rather make a shortcut as a App VM entry? Would this Be possible?

When the script creates the RAM-based AppVM, the qube will show up in your whisker menu just like regular AppVMs. When the qube is destroyed, it will disappear from there.

That's all I know.

@disp10 I think you can't but with the new appmenu, you could select to show the running online qubes on top of the applications. It's very useful to find it quickly

Hello, after running the script that creates **disposable (RAM-based) qubes**:

```
[user@dom0]$ bash disposable -c firefox -p template=os-dvm-qube -p
netvm=sys-firewall -p memory=2000 -s 400M -p label=purple
```

I have a block device (**qubes_dom0-swap**), as I understand it is a **swap file**:

```
[user@dom0]$ qvm-block
BACKEND:DEVID  DESCRIPTION      USED BY
dom0:dm-5      qubes_dom0-swap
```

After disabling **disposable (RAM based) qubes**, the block device (**swap file**) remains and disappears only after rebooting the computer.
I don't understand why it is created, because there is enough RAM in **dom0**.

Executed the command before launching firefox **disposable (RAM based) qubes**:

```
[user@dom0]$ xl info total_memory
65232
[user@dom0]$ xl info free_memory
43357
```

Executed the command after launching firefox **disposable (RAM based) qubes**:

```
[user@dom0]$ xl info free_memory
39339
```

Please tell me how to make the block device (**swap file**) disappear after **disposable (RAM based) qubes** is turned off?

The script only disables swapping to prevent disk writes. It does not create or manage the swap partition.

Yes indeed. I executed in the **dom0** terminal:

```
[user@dom0]$ sudo swapoff --all
```

The block device **qubes_dom0-swap** immediately appeared, I will try to create a new topic to clarify this behavior. **Thank you**.

Is there a guide on how to use this script?

what is the most recent version of this script? Is there a way to combine all the

different parts into a single shell script and post it as a complete script?

That way it can easily be copied into /usr/local/bin/

> what is the most recent version of this script?

It is in the OP. It will be that way until I find some time to put it in a versioning system.

> Is there a way to combine all the different parts into a single shell script and post it as a

> complete script?

What different parts? The different scripts are for different tasks.

BTW, all your posts appear with weird line breaks (when using the forum by email).

Is it possible to include the above mentioned scripts and modifications into unman’s

Github notes? I just have difficulty with following the different scripts that were added

in addition to the original script.

> Is it possible to include the above mentioned scripts and modifications into unman’s
>
> Github notes?

For that to happen, it would need @unman to do it. That would require him to agree to do that, to agree to update on potential updates to keep them up to date (to make sense of the whole thing), etc.

I don’t quite see why he would waste time with all that. Devs are busy people.

What exactly is the problem you are trying to solve with that question?

It would be great to have this integrated into Qubes OS directly instead of a third-party script. People with threat models higher than average might be worse off if they do not manage to find this thread (or don’t know that this could help them). I for one need protection against forensics and this is integral to it, and I hope it is merged into Qubes OS as soon as the devs can.

Maybe just the simplicity of having @unman 's GUI install it, but I don’t see how that could be solved since this is a standalone script and not salted. Suppose if anything @ben-grande might consider working a varient up for Qusal.

Agree, the new laptop generation with DDR5 and up to 96GB RAM shows that *RAM should not be any issue for Qubes OS anymore* (in the upcoming years).

Maybe it is worth to summarize some cons and pros of *really disposable qubes* vs. *default disposable qubes* ?

Attribute	Disposable RAM qube	Default disposable qube	Comment
…	…	…	…
…	…	…	…

It would be nice if devs put in some thoughts and also comment on a possible implementation effort.

> I for one need protection against forensics and this is integral to it, and I hope it is merged into Qubes OS as soon as the devs can.

This tool was never aimed to be anti-forensic and it can’t be relied on as such. Details were explained previously in the thread.

It should, in theory. I haven’t managed to comment that proves it otherwise (sorry, would be great if you could point towards it), but having a VM which is completely ephemeral, and run on encrypted, plausibly deniable storage for assets it needs access to, *will* make them a part of good security posture for individuals who need them.

It’s essentially akin to the live-images of most desktop distros and how they don’t leave any trace after shutting them down.

> It should, in theory.

Well, it actually does in practice but that is a *partial* side effect, not essential functionality, i.e. a non-goal. Partial = it removes the domU itself but traces of its existence remain in dom0’s logs due to how the whole system works. Additionally, the safety of domU’s erasure depends on how the RAM-based disposable is shut down. If you initiate a system reboot/shutdown while the qube is running, the cleanup part won’t work (hence one of the additional scripts).

Anti-forensics on Qubes is not easy. Especially with our suboptimal logging system.

Thank you for that, it is very relevant to know that.

@qubist i just see that some qubes that i’ve deleting are still in ~/.local/share/qubes-appmenus/ .
Do you have the same problem like me?
It’s ok for rdispxxxx but not for the others…

edit: in the vm’s folder, i still have “apps” “apps.icons” “apps.tempicons” and “apps.templates” folders.
edit 2: forget it, i try to create other vm, them remove it and all is allright
i think it’s when i restore Dom0 without use your script

No. I don’t see any rdisp* or named RAM-based disposable names in there.

Have you tried deleting the manually and checking if they would reappear?

Nice script. I took interest in the example:

```
EXAMPLE:
Launch Tor browser in a RAM based whonix disposable:
${0##*/} -p template=whonix-ws-16-dvm -p netvm=sys-whonix -c torbrowser
```

Tor protocol encapsulation is handled inside the sys-whonix gateway, not the workstation. Meaning that sys-whonix could potentially leak cleartext request/response data (page content, passwords, etc. if the connection isn’t using TLS, SNI leakage for TLS) from the disposable to disk through swap or coredumps. I’m not sure if Whonix gateway retains other data like logs, cache files, etc.

The Whonix wiki explicitly discourages 1 using a disposable gateway, so I hesitate mitigating the leak that way. Is disabling swap and coredumps in sys-whonix enough?

```
qvm-run -p sys-whonix "sudo swapoff --all; sudo sysctl -w
fs.suid_dumpable=0; sudo sysctl -w kernel.core_pattern='|/bin/false'"
```

> Is disabling swap and coredumps in sys-whonix enough?

There is no such disabling.

The script disables swap **in dom0** to prevent disk writes. It touches nothing inside any other qube. It will not put your netvm in RAM, only the AppVM you are creating resides there.

Like as i say in my edit 2 : i try it and they disappear all is ok

> Like as i say in my edit 2 […]

Sorry, I don’t receive edits. I use the forum by email.