An User & Developer Perspective on Immutable OSes

Dario Faggioli
Virtualization SW. Eng. @ SUSE

✉️ dfaggioli@opensuse.org
# dariof
🐦 @DarioFaggioli

https://dariofaggioli.wordpress.com/
https://about.me/dario.faggioli
About Me

What I do

- Virtualization Specialist Sw. Eng. @ SUSE since 2018, working on Xen, KVM, QEMU, mostly about performance related stuff
- Daily activities ⇒ how and what for I use my workstation
  - Read and send emails (Evolution, git-send-email, stg mail, ...)
  - Write, build & test code (Xen, KVM, Libvirt, QEMU)
  - Work with the Open Build Service (OBS)
  - Browse Web
  - Test OSes in VMs
  - Meetings / Video calls / Online conferences
  - Chat, work and personal
  - Some 3D Printing
  - Occasionally play games
  - Occasional video-editing
  - Maybe scan / print some document
- Can all of the above be done with an immutable OS ?
Immutable OS: What?

Either:
- An OS that you cannot modify

Or, at least:
- An OS that you will have an hard time modifying

What do you mean “modify”?
- E.g., installing packages
- ⇒ An OS on which you cannot install packages
- ⇒ An OS on which you will have an hard time installing packages
Immutable OS: What?

Seriously?

dario@Wayrath:~> cat /etc/os-release | grep ID
ID="opensuse-microos"
ID_LIKE="suse opensuse opensuse-tumbleweed"
VERSION_ID="20210104"
dario@Wayrath:~> sudo zypper install git-core
This is a transactional-server, please use transactional-update to update or modify the system.
dario@Wayrath:~> 

[dario@localhost ~]$ cat /etc/os-release | grep ID
ID=fedora
VERSION_ID=33
PLATFORM_ID="platform:f33"
VARIANT_ID=silverblue
[dario@localhost ~]$ dnf install git-core
bash: dnf: command not found
[dario@localhost ~]$
Immutable OS: Why?

Because it will stay clean and hard to break

- Does this sound familiar?
  - Let’s install foo, and it’s dependency, libfoobar_1
  - Let’s install bar (depends from libfoobar_1, we have it already)
  - Actually, let’s add an external repo. It has libfoobar_2 that makes foo work better!
  - Oh no... libfoobar_2 would break bar!!

- Yeah. It happens. Even in the best families distros :-)
- Well, the fewer the packages ...
  ... the less likely this is to happen
- How about: only the base OS + the Desktop Environment
Decoupling OS and Apps

- Base OS + the Desktop Environment
  - Comes form “system packages” (RPMs, DEBs, ...):
    - E.g., on GNOME: no further than gnome-shell
  - Packaging these would be the main focus for distro developers
  - No apps!

- Apps?
  - Cross-distro application distribution solution
Mutable or Immutable?  
Tumbleweed + OpenQA

Tumbleweed == openSUSE’s pure rolling distro 
- A new snapshot (~= release) multiple times a week  
  - [https://software.opensuse.org/distributions/tumbleweed](https://software.opensuse.org/distributions/tumbleweed)  
  - [https://en.opensuse.org/Portal:Tumbleweed](https://en.opensuse.org/Portal:Tumbleweed)  
- Nevertheless, we want it stable and reliable
Mutable or Immutable?
Tumbleweed + OpenQA

OpenQA == OS Testing “The way they’re used”
- Passive Testing
- Active Testing
- Quality Control
- Quality Assurance

Some cool materials:
- [http://open.qa/docs/](http://open.qa/docs/)
OpenQA @ Fedora Project

How Fedora uses openQA

Welcome to openQA
Life is too short for manual testing!

fedora
- BuildFedora-Cloud-32-20201024.0 (about 8 hours ago)
- BuildFedora-Cloud-33-20201024.0 (about 10 hours ago)
- BuildFedora-Rawhide-20201023.n.1 (about 16 hours ago)

Fedora Updates
- BuildUpdate-FEDORA-2020-bbf208706c (about 9 hours ago)
- BuildUpdate-FEDORA-2020-57f9ebe50e (about 11 hours ago)
- BuildUpdate-FEDORA-2020-2a350b3959 (about 11 hours ago)
OpenQA @ QubesOS

https://openqa.qubes-os.org/
OpenQA @ EndlessOS

https://openqa.endlessm.com/

Welcome to openQA
Life is too short for manual testing!

eos
- Build201023-233207 (a day ago) 49 skipped
- Build201022-233827 (2 days ago) 49 skipped
- Build201021-233129 (3 days ago) 49 skipped

Filter

no filter present, click to toggle filter form
OpenQA @ openSUSE (Tumbleweed)

Last Builds for openSUSE Tumbleweed

Responsible person: DimiTar

Automatic acceptance tests for openSUSE Tumbleweed. All tests in this job group have to pass for an automatic release of Tumbleweed to happen with the exception of the ignored issues. The process is controlled by the bot test-manager. Run testset-manager.py --dry --debug run Factory locally to check how its evaluation could look like. Call osc meta --blame prj openSUSE:Factory:ToTest | grep -A1 "pushed" to check if the repository is maybe already publish right now. You might need the following two lines in your ~/.osorc file:

```
[openSUSE:Factory]
staging=openSUSE:Factory/Staging
```

The ignored issues are taken from comments, but are stored in OSRT:IgnoredIssues. Only edit via osc as the webui destroys newlines.


The openSUSE:Factory dashboard gives an overview about the state of the current snapshot in building/testing/publishing.

Visit opensuse:factory for discussions and if you want to help.

cooio wrote about ayear.ago

pinned-description: Ignored issues

To see the list of issues @ttm ignores, run the following command: osc meta attribute openSUSE:Factory --attribute OSRT:IgnoredIssues

- **Build20201023** (a.d. ago)
  - 109 passed
  - 79 softfailed
  - 37 failed
- **Build20201022** (2.d. ago)
  - 110 passed
  - 90 softfailed
  - 28 failed
- **Build20201021** (3.d. ago)
  - 114 passed
  - 86 softfailed
  - 28 failed
- **Build20201020** (4.d. ago)
  - 105 passed
  - 78 softfailed
  - 41 failed
- **Build20201019** (5.d. ago)
  - 113 passed
  - 83 softfailed
  - 30 failed
- **Build20201018** (6.d. ago)
  - 108 passed
  - 66 softfailed
  - 44 failed
- **Build20201017** (7.d. ago)
  - 115 passed
  - 84 softfailed
  - 30 failed
- **Build20201016** (8.d. ago)
  - 101 passed
  - 70 softfailed
  - 38 failed
- **Build20201015** (9.d. ago)
  - 106 passed
  - 70 softfailed
  - 24 failed
# OpenQA: How It Works

<table>
<thead>
<tr>
<th>LibreOffice Manual Components</th>
<th>3m 4s</th>
<th>passed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LibreOffice Recent Documents</td>
<td>2m 37s</td>
<td>soft/failed</td>
<td>Soft Failed</td>
</tr>
<tr>
<td>LibreOffice Default Theme</td>
<td>1m 30s</td>
<td>passed</td>
<td></td>
</tr>
<tr>
<td>LibreOffice Double Click File</td>
<td>5m 52s</td>
<td>passed</td>
<td></td>
</tr>
<tr>
<td>LibreOffice Open Specified File</td>
<td>13m 7s</td>
<td>passed</td>
<td></td>
</tr>
</tbody>
</table>
OpenQA Can Test DEs and GUI Apps!
Mutable or Immutable? Tumbleweed + OpenQA

Tumbleweed is rock solid, thanks to OpenQA:
- We test the OS, we test the DE(s), we test the main apps
- We release only when (enough) green

But:
- We cannot test all the apps *you* use
- As soon as adding an additional repository, you may be out of tested territory!

So, to be 100% safe:
- You should not use RPMs of apps we don’t test
- You should not use external repositories (not even Packman / RPMFusion for CODECs, etc)

⇒ An Immutable OS gives you just that!
Immutable: Which? openSUSE MicroOS

- Immutable single purpose OS, based on Tumbleweed
  - [https://microos.opensuse.org/](https://microos.opensuse.org/)
  - [https://en.opensuse.org/Portal:MicroOS](https://en.opensuse.org/Portal:MicroOS)
- Based on Tumbleweed ⇒ It’s rolling
- Automatically (atomically) updates and reboot itself
- If update went wrong, automatically rollback to a working state
- Each install does only one thing:
  - One thing == Hosting containers (originally born for this)
  - One thing == Managing a K8S Cluster (Hey, that’s Kubic!)
  - One thing == Hosting VMs
  - One thing == Set Top Box
Immutable: Which ? openSUSE MicroOS

One thing == Your Desktop / Workstation
- Still early stage ~= ALPHA state
  - But usable already
  - It’s actually what I’m using since a few months
- Growing community of users
- Small community of developers
  - We need your help! :-) 

Stay in this very Devroom, for more, from Richard (at 14:45)
- openSUSE MicroOS, a platform for everything from containers, to IoT, and even the desktop
Immutable: Which Others?

- Fedora Silverblue
  
  ![Silverblue Logo]
  
  [https://silverblue.fedoraproject.org/](https://silverblue.fedoraproject.org/)
  
  “[...] unlike other operating systems, Silverblue is immutable. [...] Silverblue’s immutable design is intended to make it more stable, less prone to bugs, and easier to test and develop.”

- EndlessOS
  
  ![EndlessOS Logo]
  
  [https://endlessos.com/](https://endlessos.com/)
  
  “Endless is designed to feel natural and intuitive, making it easy to use even if you have little or no computer experience.”
Immutable OS: How?

Filesystems are read-only

- Silverblue & EndlessOS
  - Sort of: only /usr is + some **OSTree**'s “magic”

```bash
[dario@localhost ~]$ mount | grep 'boot\|luks'
/dev/mapper/luks-b7088271-39a6-46ae-addb-c3363adcffa8 on /sysroot type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7088271-39a6-46ae-addb-c3363adcffa8 on /type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7088271-39a6-46ae-addb-c3363adcffa8 on /usr type btrfs (ro,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7088271-39a6-46ae-addb-c3363adcffa8 on /var type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7088271-39a6-46ae-addb-c3363adcffa8 on /var/home type btrfs (rw,relatime,seclabel,space_cache,subvolid=256,subvol=/home)
/dev/vdal on /boot type ext4 (rw,relatime,seclabel)
[dario@localhost ~]$  
```

```
dario@endless:$ mount | grep vda
/dev/vda2 on /sysroot type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on / type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on /boot type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on /usr type ext4 (ro,relatime,errors=remount-ro)
/dev/vda2 on /var type ext4 (rw,relatime,errors=remount-ro)
dario@endless:$  
```
Immutable OS: How?

Filesystems are read-only

- **MicroOS**
  - Yes, all `/` is read-only + we even have `ro` subvolume property

```bash
dario@localhost:~> mount | grep cr_root
```

```bash
/dev/mapper/cr_root on / type btrfs (ro, relatime, space_cache, subvol=279, subvol=/@.snapshots/6/snapshot)
/dev/mapper/cr_root on /root type btrfs (rw, relatime, space_cache, subvol=261, subvol=/@/root)
/dev/mapper/cr_root on /var type btrfs (rw, relatime, space_cache, subvol=258, subvol=/@/var)
/dev/mapper/cr_root on /.snapshots type btrfs (rw, relatime, space_cache, subvol=267, subvol=/@/.snapshots)
/dev/mapper/cr_root on /home type btrfs (rw, relatime, space_cache, subvol=263, subvol=/@/home)
/dev/mapper/cr_root on /srv type btrfs (rw, relatime, space_cache, subvol=260, subvol=/@/srv)
/dev/mapper/cr_root on /boot/grub2/x86_64-efi type btrfs (rw, relatime, space_cache, subvol=265, subvol=/@/boot/grub2/x86_64-efi)
/dev/mapper/cr_root on /boot writable type btrfs (rw, relatime, space_cache, subvol=264, subvol=/@/boot/writable)
/dev/mapper/cr_root on /boot/grub2/i386-pc type btrfs (rw, relatime, space_cache, subvol=266, subvol=/@/boot/grub2/i386-pc)
/dev/mapper/cr_root on /usr/local type btrfs (rw, relatime, space_cache, subvol=259, subvol=/@/usr/local)
/dev/mapper/cr_root on /opt type btrfs (rw, relatime, space_cache, subvol=262, subvol=/@/opt)
```

```bash
dario@localhost:~> sudo btrfs property list /
```

- **ro**: read-only status of a subvolume
- **label**: label of the filesystem
- **compression**: compression algorithm for the file or directory

```bash
dario@localhost:~>  
```
Installation

- Silverblue
  - Grab the image [here](#) and install

- EndlessOS
  - Grab the image [here](#) and install

- Grab the image [here](#) and install
  - Choose one of the “MicroOS Desktop [ALPHA]” flavors
Installation With Encrypted Disk

A must have, e.g., for laptops

- **Silverblue**
  - Works just out of the box
  - `/boot` is not encrypted
  - Asks the password once (during boot)

- **EndlessOS**
  - Didn’t check ;-P

- **MicroOS**
  - Works just out of the box
  - Everything is encrypted, including `/boot`
  - Asks the password twice
    - But, can be fixed:
      - [Avoiding typing the passphrase twice](#)
Filesystem & Filesystem Layout

EndlessOS

- Typical layout of an OSTree managed system
- Filesystem is ext4, by default

```
dario@endless:~$ mount | grep vda
/dev/vda2 on /sysroot type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on / type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on /boot type ext4 (rw,relatime,errors=remount-ro)
/dev/vda2 on /usr type ext4 (ro,relatime,errors=remount-ro)
/dev/vda2 on /var type ext4 (rw,relatime,errors=remount-ro)
dario@endless:~$  
```
Filesystem & Filesystem Layout

Silverblue

- BTRFS, / and /home subvolumes
- No properties, all COW

[dario@localhost usr]$ mount | grep btrfs
/dev/mapper/luks-b7eb90271-39a6-46ae-addb-c3363adc0a8 on /sysroot type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7eb90271-39a6-46ae-addb-c3363adc0a8 on / type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7eb90271-39a6-46ae-addb-c3363adc0a8 on /usr type btrfs (ro,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7eb90271-39a6-46ae-addb-c3363adc0a8 on /var type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/root)
/dev/mapper/luks-b7eb90271-39a6-46ae-addb-c3363adc0a8 on /var/home type btrfs (rw,relatime,seclabel,space_cache,subvolid=258,subvol=/home)

[dario@localhost usr]$ sudo btrfs subvolume list /
ID 256 gen 3082 top level 5 path home
ID 258 gen 3110 top level 5 path root
[dario@localhost usr]$
[dario@localhost usr]$ sudo btrfs property get /
label=fedora_fedora
[dario@localhost usr]$ sudo btrfs property get /home
MicroOS

- BTRFS, with the classic openSUSE subvolumes layout
- / has ro set, the others don’t

```bash
/dev/mapper/cr_root on /var type btrfs (rw,relatime,space_cache,subvolid=258,subvol=/var)
/dev/mapper/cr_root on /boot/writable type btrfs (rw,relatime,space_cache,subvolid=264,subvol=/boot/writable)
/dev/mapper/cr_root on /snapshots type btrfs (rw,relatime,space_cache,subvolid=267,subvol=/snapshots)
/dev/mapper/cr_root on /boot/grub2/x86_64-efi type btrfs (rw,relatime,space_cache,subvolid=265,subvol=/boot/grub2/x86_64-efi)
/dev/mapper/cr_root on /boot/grub2/1386-pc type btrfs (rw,relatime,space_cache,subvolid=266,subvol=/boot/grub2/1386-pc)
/dev/mapper/cr_root on /home type btrfs (rw,relatime,space_cache,subvolid=263,subvol=/home)
/dev/mapper/cr_root on /opt type btrfs (rw,relatime,space_cache,subvolid=262,subvol=/opt)
/dev/mapper/cr_root on /srv type btrfs (rw,relatime,space_cache,subvolid=260,subvol=/srv)
/dev/mapper/cr_root on /usr/local type btrfs (rw,relatime,space_cache,subvolid=259,subvol=/usr/local)

dario@localhost:~> sudo btrfs property get /
ro=true
label=
dario@localhost:~> sudo btrfs property get /home
ro=false
dario@localhost:~> sudo btrfs property get /var
ro=false
```
## Filesystem & Filesystem Layout

### MicroOS

- More on subvolumes...

```bash
dario@localhost:~> sudo btrfs subvolume list /
ID 256 gen 33 top level 5 path @
ID 258 gen 906 top level 256 path @/var
ID 259 gen 841 top level 256 path @/usr/local
ID 260 gen 36 top level 256 path @/srv
ID 261 gen 658 top level 256 path @/root
ID 262 gen 36 top level 256 path @/opt
ID 263 gen 902 top level 256 path @/home
ID 264 gen 24 top level 256 path @/boot/writable
ID 265 gen 26 top level 256 path @/boot/grub2/x86_64-efi
ID 266 gen 58 top level 256 path @/boot/grub2/i386-pc

At install time

ID 267 gen 872 top level 256 path @/.snapshots
ID 268 gen 68 top level 256 path @/.snapshots/1/snapshot
ID 271 gen 55 top level 256 path @/.snapshots/2/snapshot
ID 272 gen 322 top level 256 path @/.snapshots/3/snapshot
ID 274 gen 462 top level 256 path @/.snapshots/4/snapshot
ID 278 gen 469 top level 256 path @/.snapshots/5/snapshot
ID 279 gen 478 top level 256 path @/.snapshots/6/snapshot
ID 281 gen 657 top level 256 path @/var/lib/containers/storage/btrfs/subvolumes/0d6854eefed9b168f925
ID 282 gen 658 top level 256 path @/var/lib/containers/storage/btrfs/subvolumes/2c8491680004e9ablede
ID 283 gen 827 top level 256 path @/var/lib/containers/storage/btrfs/subvolumes/31c962f14282eab0bef7

During usage
```
MicroOS

- Some COW, some noCOW
- Personally, I think /home should become COW

```
dario@localhost:~$ sudo lsattr / 2> /dev/null
----------C---- /etc
----------C---- /boot
----------C---- /home
----------C---- /opt
----------C---- /root
----------C---- /usr
----------C---- /var
----------C---- /lib
----------C---- /bin
----------C---- /lib64
----------C---- /mnt
----------C---- /sbin
```
Post Installation Configuration

- **Silverblue**
  - Add Flathub
    - (more on this later...)

- **EndlessOS**
  - Nothing, you’re all set

- **MicroOS (Hey, we’re still in ALPHA!)**
  - Add Flathub
  - For toolbox to work:
    # echo "dario:100000:65536" > /etc/subuid
    # echo "dario:100000:65536" > /etc/subgid
  - For controlling update/reboot yourself
    $ sudo systemctl disable --now transactional-update.timer
    $ sudo systemctl disable --now rebootmgr.service

$ flatpak remote-add --user flathub \\https://flathub.org/repo/flathub.flatpakrepo
Adding Packages (packages, not flatpaks)

You shouldn’t! But what if you really want/need?

- **EndlessOS**
  - Truly and fully immutable
  - You can add (or remove) anything

- **Silverblue**
  - Possible. It’s called “layering”
  - Handled via `rpm-ostree` (E.g., Adding Layered Packages)
  - **Always reboot** ASAP after layering something
  - Changes visible after the reboot
Adding Packages (packages, not flatpaks)

- **MicroOS**
  - Possible. Handled via `transactional-update`
    - `$ sudo transactional-update pkg install git-core`
    - `$ sudo transactional-update shell`
      - `#> sudo zypper ref`
      - `#> sudo zypper install git-core`
  - Check `transactional-update shell --continue`
  - **Always reboot** ASAP after using `transactional-update`
  - Changes visible after the reboot

- **So, transactional-update =~ rpm-ostree?**
  - Technically, not at all (although usage and “effects” are similar)
  - Transactional update leverage BTRFS
Transactional Updates

BTRFS is awesome
- We can modify the system (e.g., install packages) without affecting the instance of it that is running
- We do not need anything more than BTRFS itself

Implementation:
- The FS and subvolumes are really read-only
- Request to modify ⇒ we create a snapshot and do that in there
- At next reboot, we boot in the new snapshot

Update / Install / ...

```
$ sudo reboot
```
Transactional Updates

Materials:

- The Transactional Update Guide
Altering The Root Filesystem

For instance:
- Running NVIDIA’s `.run` driver installer
- Adding a wireless card’s hex file in `/lib/firmware`
- Whatever...

EnlessOS & Silverblue
- No way
  - EndlessOS: no way to alter the FS at all
  - Silverblue: layering. **But** it’s only for packages

MicroOS
- `$ sudo transactional-update shell`
  - `#> <do_what_you_need>`
  - `#> exit`
- `$ systemctl reboot`
Shall We Alter The Base OS A Lot Then?

NO, GOD! NO, GOD, PLEASE NO! NO! NO!

NOOOOOOOOOOOOOO!
Are We Constantly Rebooting?

This is my MicroOS workstation. Judge yourself:

```
dario@Wayrath:~> uptime
 22:34:38 up 7 days 5:40, 2 users, load average: 3.30, 2.95, 2.37
```

How so?

- For apps:
  - Flatpak
- For troubleshooting or debugging:
  - toolbox
- For development or “non-Flatpaked” apps:
  - toolbox

Installing/removing packages on the base OS tends to zero
Flatpak

Flatpak, [https://flatpak.org/](https://flatpak.org/)
- Application are self-contained
- There’s some sharing (via Runtimes)
- All needed file installed either in `/var` or `$HOME`

App repositories (remotes)
- **MicroOS**
  - Use Flathub “official” remote. To be added manually
- **Silverblue**
  - Has its own (preconfigured) remote but not many apps
  - You most surely need Flathub. To be added manually
- **EndlessOS**
  - Flathub + their own (both preconfigured)
Flatpak

What’s there, just immediately after install?

- **EndlessOS**
  - Lots of flatpaks pre-installed
  - System already usable for its intended use-case

- **Silverblue**
  - Some flatpaks pre-installed (from Fedora’s own remote)
  - Browser (Firefox) installed from RPMs
  - System is ok for basic usage, need Flathub for more apps

- **MicroOS**
  - No flatpak pre-installed
  - Do add Flathub and pick apps as first thing!
    - To Be Done: preinstall something. Not trivial, though
  - No browser
    - To Be Done: not clear
Free? Not Free? Let’s Have a Beer!
(Wait, was it like that?)

Flathub has:
- Free Software
- Free Software built with CODECs for “patent encumbered” formats (e.g., VLC, OpenShot)
- Proprietary Software

IANAL but, most likely:
- Flathub as a remote cannot be enabled by default in MicroOS
- We cannot pre-install stuff from Flathub in MicroOS
- I don’t think we fancy pre-installing stuff we don’t build

Silverblue:
- Firefox is “layered”
  ⇒ Even if a Flatpak exist?
- They build some flatpaks their own
  ⇒ Only a handful, Flathub needed anyway
  ⇒ Duplication with same apps from Flathub

To Be Done: think about it...
Installing & Updating Flatpaks

Installing / Updating flatpaks:
- Via CLI
  - EnlessOS / Silverblue / MicroOS:
    - Flatpak update
- Via GUI
  - EndlessOS:
    - Via GNOME Software
    - (with their customized UI)
  - Silverblue:
    - Via GNOME Software
  - MicroOS:
    - Via GNOME Software or Discover
- To Be Done: They work, but need some tweaking
Email, Calendaring, IM & Office Apps

- Mail, calendaring, contacts, ...
  - Evolution, org.gnome.Evolution
  - Calendar, org.gnome.Calendar
  - Contacts, org.gnome.Contacts
  - GNOME Clocks, org.gnome.clocks
  - Weather, org.gnome.Weather

- Documents
  - Evince, org.gnome.Evince
  - GNOME Documents, org.gnome.Documents
  - LibreOffice, org.libreoffice.LibreOffice

- Messaging
  - RocketChat, chat.rocket.RocketChat
  - Pidgin, im.pidgin.Pidgin
  - Telegram, org.telegram.desktop
  - Signal, org.signal.Signal
Editors, Tools, Graphics

- Editors:
  - Vim, [org.vim.Vim](https://www.vim.org)
  - Gedit, [org.gnome.gedit](https://git.gnome.org/pub/gnome/sources/gedit)
  - Setzer, [org.cvfosammmm.Setzer](https://github.com/cvfosammmm)
  - Eclipse, [org.eclipse.Java](https://www.eclipse.org)

- Graphics
  - GIMP, [org.gimp.GIMP](https://gimp.org)
  - Krita, [org.kde.krita](https://kontact.kde.org)
  - Blender, [org.blender.Blender](https://www.blender.org)

- VMs:
  - GNOME Boxes, [org.gnome.Boxes](https://valpol.org)

- Tools:
  - Regex Tester, [com.github.artemanufrij.regextester](https://github.com/artemanufrij/regextester)
  - Meld, [org.gnome.meld](https://meldmerge.org)
  - Boop-GTK, [uk.co.mrbenshef.Boop-GTK](https://github.com/benshef/Boop-GTK)
Utilities, Configuration

- Misc utilities:
  - SyncThing, me.kozec.syncthingtk
  - Barrier, com.github.debauchee.barrier
  - Seahorse, org.gnome.seahorse.Application

- Config:
  - Dconf Editor, ca.desrt.dconf-editor
  - Flatseal, com.github.tchx84.Flatseal
  - GPU-Viewer, io.github.arunsivaramanneo.GPUViewer
Browsing

● Firefox, org.mozilla.firefox
  ○ Works great, including video codecs
● Epiphany, org.gnome.Epiphany
● Chromium, org.chromium.Chromium
  ○ Works great, including video codecs
● Google Chrome, com.google.Chrome
  ○ Available in flathub-beta
  ○ Tested briefly, seems to work fine

NB: GNOME Shell Extension can’t be installed from a “Flatpak-ed” browser yet:
● Browser installed from RPM
● An application for managing them
  (MicroOS: we’re investigating this second solution)
Gaming

- Steam, [com.valvesoftware.Steam](https://steam.com)
  - Works great, even SteamPlay/Proton
- NVIDIA Drivers
  - `$ sudo transactional-update shell`
    - `zypper ar https://download.nvidia.com/opensuse/tumbleweed NVIDIA`
    - `zypper ref`
    - `zypper in nvidia-glG05 x11-video-nvidiaG05`
    - `exit`
  - `$ sudo reboot`
  - Brings in gcc and some development packages (not ideal... Thanks NVIDIA, I guess :-/)
  - NB flatpak picked up automatically:
    - [org.freedesktop.Platform.GL.nvidia-450-66](https://github.com)
    - [org.freedesktop.Platform.GL32.nvidia-450-66](https://github.com)
Video: Viewing, Editing & Codecs

Remember: no ffmpeg and/or CODEC RPM (e.g., from Packman/RPMFusion) installed on system

- VLC, \texttt{org.videolan.VLC}
  - Has the proper codecs
- Pitivi, \texttt{org.pitivi.Pitivi}
  - Has the proper codecs
- Openshot, \texttt{org.openshot.OpenShot}
  - Has the proper codecs
- Shotcut, \texttt{org.shotcut.Shotcut}
  - Has the proper codecs
- Cheese, \texttt{org.gnome.Cheese}
  - Works well with my webcam
[3D] Printing & Scanning

- **3D Printing**
  - Prusa Slicer, `com.prusa3d.PrusaSlicer`

- **Printing**
  - Drivers must be installed on the system (RPMs ⇒ PPDs)
  - The ones for the most common Ones are (on all the 3 OSes)
  - You may need to add drivers with (e.g., with `transactional-update`)

- **Scanning**
  - Paperwork, `work.openpaper.Paperwork`
    - Not working yet... Still not sure why
    - (Yeah, well, most scanners, e.g., from All-in-one printers, have Web-ish interface)
Toolbox

An easy way to start a read-write environment (in a podman container):
- With your user configured
- You have your home there, in its usual place
- Your files have the proper owner, group, permissions
- You reach your SSH agent (running on the host)
- You can launch graphical apps
- You have `sudo`
- You can install and remove packages

Sounds pretty handy:
- For installing apps not available/not working as Flatpaks
- For doing development inside it
- For troubleshooting and debugging the immutable OS

Check this other talk (tomorrow): “By The Power of toolbox!”
Toolbox

A launcher for a privileged podman container

- Silverblue & EndlessOS
  - github.com/containers/toolbox
  - Was Bash, now Go (EndlessOS still using the old bash version)

- MicroOS
  - github.com/kubic-project/microos-toolbox
  - Bash

BEWARE:
- It’s for convenience
- It’s not a security enhancing tool!
Project You Go, toolbox You Find

UI is (almost) compatible, at least!

- Create a toolbox
  - Silverblue: `toolbox create`
  - MicroOS
    - Either: `toolbox -u`
    - Or: `toolbox create`

- Entering a toolbox:
  - Silverblue:
    - `toolbox enter`
  - MicroOS
    - Either: `toolbox -u`  # creates it, if doesn’t exist
    - Or: `toolbox enter`

- Create (and enter) a toolbox as root:
  - Silverblue: `sudo toolbox create && sudo toolbox enter`
  - MicroOS:
    - Either: `toolbox -u -r`
    - Or: `toolbox create -r && toolbox enter -r`
Toolbox for Development: Building Xen

● Dependencies for building Xen from sources:

● Toolbox to the rescue:
  ○ $ toolbox enter
    $> sudo zypper in <all_the_dependencies_above>
    $> git clone git://xenbits.xen.org/xen.git xen.git
    $> cd xen.git
    $> ./configure
    $> make -j8 xen tools
Toolbox for Development: Working With OBS

Requires installing packages, using VMs for building, etc.

- I need a `-r` toolbox, for mounting filesystems in the build VM (I think)

```
$ toolbox create -r
$ toolbox enter -r
$> zypper ar https://download.opensuse.org/ [...] /openSUSE:Tools.repo OBS
$> zypper in cpio osc build [...] 
$> osc mkpac / co / vc
$> [...] 
$> osc vc
$> osc build --vm-type=kvm
$> osc commit
```
Toolbox for Graphical Apps

- They work too! ⇒ No need installing them in base OS
- $ toolbox enter
  $> sudo zypper in gedit virt-manager
  $> gedit
  $> virt-manager

On MicroOS also do, inside the toolbox:
$> sudo zypper in xorg-x11-fonts-core
$> sudo zypper in adwaita-icon-theme
Toolbox for “GL” Graphical Apps

- Kernelshark as an example:
  - $ toolbox enter
    - $> kernelshark
      libGL error: No matching fbConfigs or visuals found
      libGL error: failed to load driver: swrast
      QOpenGLWidget: Failed to create context
      QOpenGLWidget: Failed to create context
      qt.qpa.backingstore: composeAndFlush: QOpenGLContext creation failed
      qt.qpa.backingstore: composeAndFlush: makeCurrent() failed
      ...

- I have NVIDIA with proprietary drivers here. What if...
  - $ toolbox enter
    - $> sudo zypper addrepo https://download.nvidia.com.opensuse/tumbleweeNVIDIA
    - $> sudo zypper ref
    - $> sudo zypper in x11-video-nvidiaG05

- It installs stuff like:
  - kernel-default-devel, nvidia-gfxG05-kmp-default, nvidia-g1G05 Inside the container ? :-O
Well, it works!
Toolbox for Troubleshooting

E.g., I need to do an \texttt{nmap}

- It’s not installed
- I don’t want to reboot now!

$\ \texttt{toolbox\ enter\ -r}\ \ $\ \texttt{# runs as root on the host (necessary for scanning “low ports”)}

$\ \texttt{zypper\ install\ nmap}\ \ $\ \texttt{# we can add packages, no problem}

$\ \texttt{toolbox\ enter\ -r}\ \ $\ \texttt{# runs as root on the host (necessary for scanning “low ports”)}

On Silverblue, it would be:

$\ \texttt{sudo\ toolbox\ enter}$
Some Stats

● RPM Packages
  ○ When I was using Tumbleweed: more than 4000 packages
  ○ On my MicroOS Desktop: less than 1000 packages
    ■ Inside a development toolbox there: ~1300 packages
      ● (has GUI apps & libs too)
  ○ On a stock Fedora Silverblue: ~1200 packages

● Flatpaks
  ○ Apps installed: 112
  ○ All flatpaks (Apps + Runtimes + Locales): 202
  ○ Disk space: ~50 GB
Update the OS Packages

Silverblue & EndlessOS
- Handled by OSTree
- Integrated in GNOME Software
  - It handles both packages via OSTree and flatpaks

MicroOS
- By default: completely automatic
- Every day the OS checks, installs updates and reboots itself
- Super cool... But not necessarily for a Desktop:
  - we want to be in control of the reboots
  - I personally have this disabled
  - TBD: Integrate updates in GNOME Software or similar

All 3 ⇒ Reboot required after an update
How About: Rebooting Even Less?

Can we “containerize more”?

- That would mean having to reboot even less

- Current situation (all 3):
  - Base system + Desktop Environment come from packages

- @fcrozet super-cool (WIP!) idea:
  - Desktop Environment in a container
    - GDM Containers
    - Updating DE packages means no reboot!
Rollback After a Bad Update

Silverblue
- At boot, pick a previous version
- Once booted: `rpm-ostree rollback`

MicroOS
- At boot, pick a previous snapshot
- Once booted: `transactional-update rollback`
Remember this?

- Daily activities ⇒ how and what for I use my workstation
  - Read and send emails (Evolution, git-send-email, ...)
  - Write, build & test code (Xen, KVM, Libvirt, QEMU)
  - Work with the Open Build Service (OBS)
  - Browse Web
  - Test OSes in VMs
  - Meetings / Video calls / Online conferences
  - Chat, work and personal
  - Some 3D Printing
  - Occasionally play games
  - Occasional video-editing
  - Maybe scan / print some document
Remember this?

- Daily activities ⇒ how and what for I use my workstation

- Read and send emails (Evolution, git-send-email, ...)
- Write, build & test code (Xen, KVM, Libvirt, QEMU)
- Work with the Open Build Service (OBS)
- Browse Web
- Test OSes in VMs
- Meetings / Video calls / Online conferences
- Chat, work and personal
- Some 3D Printing
- Occasionally play games
- Occasional video-editing
- Maybe scan / print some document
About Myself

- Ph.D on Real-Time Scheduling @ ReTiS Lab, SCHED DEADLINE
- 2011, Sr. Software Engineer @ Citrix
  The Xen-Project, hypervisor internals, NUMA-aware scheduler, Credit2 scheduler, Xen scheduler maintainer
- 2018, Virtualization Software Engineer @ SUSE
  Still Xen, but also KVM, QEMU, Libvirt;
  Scheduling, VM’s virtual topology, performance evaluation & tuning