





Can MicroOS Desktop Be Your "Daily Driver" ? (SPOILER ALERT: Probably YES!)

Dario Faggioli, dfaggioli@suse.com





Can It Really Be ?

Proof that it can is:

dario@Wayrath:~> cat /etc/os-release NAME="openSUSE MicroOS" # VERSION="20201009" ID="opensuse-microos" ID_LIKE="suse opensuse opensuse-tumbleweed" VERSION_ID="20201009" PRETTY_NAME="openSUSE MicroOS" ANSI_COLOR="0;32" CPE_NAME="cpe:/o:opensuse:microos:20201009" BUG_REPORT_URL="https://bugs.opensuse.org" HOME_URL="https://bugs.opensuse.org/" DOCUMENTATION_URL="https://en.opensuse.org/Portal:MicroOS" LOGO="distributor-logo"

• The first part of this talk is also covered <u>by this blog post</u>

About Me What I do

- Virtualization Specialist Sw. Eng. @ SUSE since 2018, working on Xen, KVM, QEMU, mostly about performance related stuff
- Daily activities \Rightarrow 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)
 - o Browse Web
 - Meetings / Video calls / Online conferences
 - Chat, work and personal
 - Occasionally play games
 - Occasional video-editing
 - Maybe scan / print some document
- Can all of the above be done with MicroOS <u>already</u>?

What is MicroOS

- Immutable single purpose OS, based on Tumbleweed, born as container host but not limited to that use case
 - <u>https://microos.opensuse.org/</u>
 - <u>https://en.opensuse.org/Portal:MicroOS</u>
 - Richard's and Ish's talks!

openSUSE Conference 2019

openSUSE MicroOS in Production Deploying apps using Podman

Ish Sookur



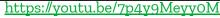
https://youtu.be/8gGjcKdOWIc

https://youtu.be/nIwqzGbX-oc

What is MicroOS as a Desktop

- MicroOS \Rightarrow Single purpose immutable OS
- Each install does only one thing:
 - One thing == Hosting containers
 - \circ One thing == Hosting VMs
 - \circ One thing == Set Top Box
 - <u>One thing == Your Desktop</u>
 - More talks from Richard
 - The latest one, yesterday!







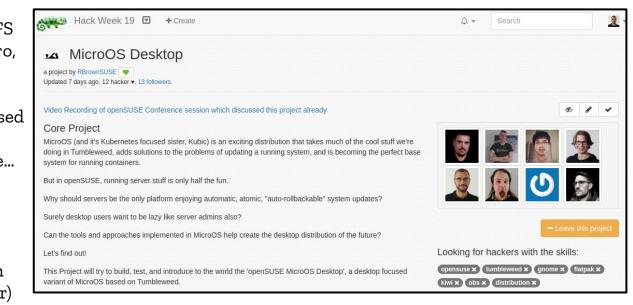
https://youtu.be/ASSkQH9kNa0



How I Got Involved

• SUSE Hack Week 19 (which happened in 2020)

- Chance for SUSE employees to work on do whatever they find cool
- MicroOS as a Desktop
 - Immutable, taking advantage of BTRFS
 - Base OS from distro, apps from other (proper?) sources
 - Rolling base, as based on Tumbleweed
 - Rolling but reliable...
 as based on
 Tumbleweed
- I found it cool! :-)
 - $\circ \quad \ \ {\rm Tried \ and \ tested \ it}$
 - Started hacking on toolbox (see later)



(https://hackweek.suse.com/projects/microos-desktop)

Why I Tried and Why I'm Liking it

- A relatively small and immutable base OS
 - \circ Stable and reliable
 - $\circ \quad \text{Immutable} \Rightarrow \text{much more difficult to mess-up}$
- Issues with package dependencies:

- \circ Fewer packages \Rightarrow a lot less likely to happen (in fact, never happened in months...)
- BTRFS at its finest:
 - Updates in non-running snapshots. Automatic rollback with <u>health-check</u>
- Apps from Flatpak/Flathub
 - Contributed to Flathub directly from upstream app developers
 - $\blacksquare \quad \Rightarrow$ Effort done once, multiple (all?) distro can profit from that
 - ⇒ Distro/OS developers can focus on OS, app developers can focus on apps
- Tumbleweed is rock solid, thanks to OpenQA, etc
 - \circ $\;$ As soon as you add an additional repository, this may change ...
 - Technically you're not using the distro that has been developed & tested any longer
 - (In practice, fine, especially for Packman, etc. But, still.)
 - $\circ \quad \text{Here you don't need } \textbf{any} \text{ additional repository !}$

Installing

- Just grab it: <u>https://microos.opensuse.org/</u>, and install it!
- Choose "MicroOS Desktop [GNOME] [ALPHA]"
- Choose "KDE Plasma" if you want, but I've never tested it. No idea if/how it works!

openSUSE -O- MicroOS		openSUSE HicroOS		
System Role	<text><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></text>	Installation Settings	<section-header><section-header><section-header><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item><list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></list-item></section-header></section-header></section-header>	Abot Back Intellige
Help Release Notes	Abo <u>r</u> t <u>B</u> ack <u>N</u> ext	Help Release Notes		Abo <u>r</u> t <u>B</u> ack <u>Install</u>

Immediately After Installing

• Add FlatHub as flatpak remote

- \$ flatpak remote-add --user flathub https://flathub.org/repo/flathub.flatpakrepo
- Some GNOME Software (black) magic:
 - \$ gsettings set org.gnome.software install-bundles-system-wide false
 - \$ gsettings set org.gnome.software allow-updates false
 - \$ gsettings set org.gnome.software download-updates false
 - \$ gsettings set org.gnome.software enable-repos-dialog false
 - \$ gsettings set org.gnome.software first-run true

• Some zypper (black) magic:

- o \$ sudo rm -Rf /var/cache/app-info
 - \$ sudo transactional-update shell
 - # rpm -e --nodeps libzypp-plugin-appdata
 - # zypper al libzypp-plugin-appdata
 - # exit
 - \$ sudo reboot
- Shouldn't this should all be done automatically?
 - Indeed ! Patches / SRs welcome :-P

Some More Customization

Should be done automatically too, IMO. Again, contributions welcome!

- For toolbox (see later) ---
 - o # echo "<yourusername>:100000:65536" > /etc/subuid # echo "<yourusername>:100000:65536" > /etc/subgid
- I want passwordless sudo
 - o # usermod -a -G wheel <yourusername>
 # echo "%wheel ALL = (root) NOPASSWD:ALL" > /etc/sudoers.d/wheel
- I want to disable automatic updating and rebooting
 - I will deal with updating (and rebooting) manually
 - \$ sudo systemctl disable --now transactional-update.timer \$ sudo systemctl disable --now rebootmgr.service
 - \circ Let's check:
 - \$ sudo rebootmgrctl is-active
 RebootMgr is dead
 \$ sudo rebootmgrctl status
 Error: The name org.opensuse.RebootMgr was not provided by any .service
 files

Additional Repositories & Packages

- Add repositories, e.g. Packman:
 - openSUSE Wiki: Additional package repositories
 - All of Packman:
 - zypper ar -cfp 90
 <u>http://ftp.gwdg.de/pub/linux/misc/packman/suse/openSUSE_Tumbleweed/</u> packman
 - \circ Install codecs
- Add <more repositories>
- Install <a lot of packages for whatever I need>

Right?

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

N0000000000000

Installing Packages

- No zypper (well, it's there but it's locked \Rightarrow try it, it won't work!)
- Transactional-update , directly:
 - \$ sudo transactional update pkg install wget unzip
 - \$ sudo reboot
- transactional-update , via shell:
 - \$ sudo transactional-update shell
 - # zypper ref
 - # zypper in wget unzip
 - # exit
 - \$ sudo reboot
- Multiple sessions:
 - o \$ sudo transactional -update pkg install wget <u>https://youtu.be/e3_X7v7aoHk</u>
 [...]
 \$ sudo transactional-update shell --continue
 # zypper in unzip
 # exit
 \$ sudo reboot
- <u>Reboot always necessary</u>, for seeing and using new packages: <u>The Transactional Update Guide</u>



Are We Constantly Rebooting?

- Nah!
 - For instance, I haven't rebooted this workstation since 3 days and 16 hours (and counting!)
 dario@Wayrath:~> uptime

08:51:42 up 3 days 16:33,

- How so?
 - For apps:
 - Flatpak (from Flathub, <u>https://flathub.org/</u>)

dario@Wayrath:~>

- \circ For troubleshooting / debugging:
 - toolbox
- For development
 - toolbox
- For "development & apps":
 - toolbox
- Installing/removing activities RPMs on the base OS tends to zero

Flatpak

- It will be our main install source, for all applications
- Via GNOME Software
 - \circ Once configured as shown
- Via cli
 - flatpak install org.gnome.gedit
 alias gedit='flatpak run org.gnome.gedit'

dario@Wayra	ath:~>	flatpak ps	
Instance	PID	Application	Runtime
2416651609	17472	org.vim.Vim	org.freedesktop.Platform
4132321407	17296	org.libreoffice.LibreOffice	org.freedesktop.Platform
1453829993	16762	org.mozilla.firefox	org.freedesktop.Platform
588396198	16391	chat.rocket.RocketChat	org.freedesktop.Platform
729209139	16107	org.gnome.Evolution	org.gnome.Platform
2888792739	3792	org.gnome.gedit	org.gnome.Platform
4074027177	2547	org.telegram.desktop	org.kde.Platform
1538198007	2541	com.github.debauchee.barrier	org.kde.Platform
3656091025	2542	im.pidgin.Pidgin	org.gnome.Platform
3380805255	2545	me.kozec.syncthingtk	org.gnome.Platform

Remove

Remove

Remove

Remove

Remove 5,5148 Remove 26,548

Remove

Remove

Remove

Remove

Explore Installed

Free and open source 3D creation suite
Port of IvanMathy's Boos to GTK, a scriptable scratchead for

Calendar for GNOME

A contacts manager for GNOME

Manage Drives and Media

A graphical tool for editing the dconf database

Check folder sizes and available disk space

A document manager application for GNOME

Barrier - Share mouse and keyboard over the local network

Take photos and videos with your webcam, with fun graphical

Barrier

a Blender

Boop-GTK

Calendar

@ Contacts

Dconf Editor

Disks

Disk Usage Analyzer

٩	Explore Installed	= _ n ×
Featured Applications		
		Steam Manage and play games distributed by Steam
Categories		
Ja Audio & Video	Communication & News	C Productivity
🛱 Games	Ø Graphics & Photography	Z Developer Tools
Education & Science	🕼 Utilities	
Recent Releases		
Postman	Cozy Marker	ProtonMail Brid_

Toolbox

- A shell script that launches a privileged container
 - Check: <u>https://kubic.opensuse.org/blog/2019-10-22-toolbox/</u>
 - Most other immutable OSes has something similar (e.g., <u>Silverblue</u>)
 - The host file system will be visible/accessible while inside the container (bind mounts, etc)
- The container can run:
 - As root
 - You may or may not have your regular user in the toolbox container
 - When you are root in the toolbox container run as root, you're kind of root on the host
 - \circ As your regular user
 - Thanks to <u>"rootless podman"</u>
 - You have your regular user in the toolbox container
 - Even when you are root in the toolbox container, you are not root on the host
- BEWARE: <u>"privileged container"</u> & <u>"can run as root"</u>
 - $\circ \quad \ \ {\rm It's} \ \, {\rm {\bf not}} \ \, {\rm a} \ {\rm security} \ {\rm enhancing} \ {\rm tool}$
 - I.e.: <<I can do whatever I want, I'm in a container, I won't affect or disrupt the base OS, right?>>
 - No, this is not the right mindset
 - \circ ~ You're not less secure or safe than when you're working directly on the base OS
 - You're not more secure or safe either!

Different Kind of Toolbox-es

• Creating and entering a toolbox that <u>runs as your user</u>, and <u>be your own user while inside it</u>:

- Useful for using toolbox as your user / developer environment
- \$ toolbox -u
 \$ u ⇒ you will have your user, your /home, etc
 \$ you're inside the toolbox already!
 \$ toolbox -u -t foo # -t ⇒ to give this toolbox a name ('t' for 'tag')
 \$ you're now inside the toolbox tagged 'foo'
 \$ sudo su
 # you're becoming root in container. But, e.g., you still
 # won't be able to touch files owned by root on the host!
- Creating and entering a toolbox that <u>runs as your user</u>, but has <u>only root user inside it</u>:
 - Useful for using toolbox as a debugging and troubleshooting environment
 - \$ toolbox # no -u ⇒ no user except root, nothing in /home
 - #> # your are root already. But root in toolbox
 - #> # does not map on root on the host

Different Kind of Toolbox-es

• Creating and entering a toolbox that <u>run as root</u>, and be <u>your own user while inside it</u>:

- Useful for using toolbox as your user / developer environment (that needs "special powers")
- Creating and entering a toolbox that <u>runs as root</u>, and has <u>only root user root inside it</u>:
 - Useful for using toolbox as a debugging/troubleshooting environment (with "special powers")
 - \$ toolbox -r # -r \Rightarrow the toolbox run as root on the host
 - # > # no -u \Rightarrow no user except root, nothing in /home. Also,
 - #> # your are root already, and that does map with root on the host

Managing Your Toolbox-es

• Toolbox is **stateful**:

- <u>Yesterday</u> you created a toolbox, and you install stuff, change configs, etc
- <u>Today</u> you stop the toolbox, you turn off the PC and take the day off
- <u>Tomorrow</u> toolbox will still have all the software and all the config changes you made

• Listing toolbox-es running as user:

o \$ podman ps

```
CONTAINER IDIMAGECOMMANDCREATEDSTATUSNAMES5cb19ade1fb1[...]toolbox:latestsleep +Inf 3 weeks agoUp 3 hours agotoolbox-dario-user
```

- Listing all toolbox-es created as user (running ot not):
 - \$ posman ps --all

```
CONTAINER IDIMAGECOMMANDCREATEDSTATUSNAMES5cb19ade1fb1[...]toolbox:latestsleep +Inf3 weeks agoUp3 hourstoolbox-dario-user502722d98390[...]toolbox:latestsleep +Inf3 weeks agoExitedtoolbox-dario-user-dev
```

• For toolbox-es created as root:

- \$ sudo podman ps # list the running ones
- Sudo podman ps --all # list all of them

• Removing toolbox-es:

- 0 \$ podman rm <toolbox_name/ID> # for a toolbox running as user
- \$ sudo podman rm <toolbox_name/ID> # for a toolbox running as root

Toolbox For TroubleShooting

Toolbox is super handy for debugging and troubleshooting

- Example: you need to do a strace ls
 - You can try... but strace is not installed!
 - Install it with transactional-update pkg inand then reboot ?!?
 - No!
 - \$ toolbox
 # runs as your user on the host (no -r)
 #> zypper in strace
 # you're root in toolbox, but that
 # does not map to root on the host
 # here you go your strace!
- Example, you need to nmap some host
 - Again, nmap is not there, and you don't want to reboot!
 - Nmap needs "real root", to scan low ports
 - \$ toolbox -r
 # runs as root on the host (-r)
 #> zypper install nmap
 # we can add packages, no problem
 # nmap -sS 192.168.0.2
 # you're root in toolbox and that
 <...>
 # does map to root on the host

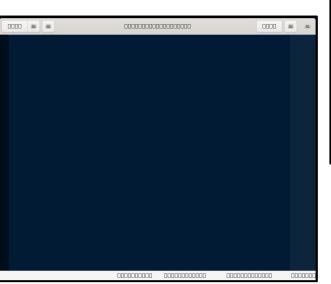
Toolbox Config File

- Some tweaking possible (and more possibilities of tweaking being worked on ;-P)
- Config file:
 - > \$ cat ~/.toolboxrc REGISTRY=registry.opensuse.org IMAGE=opensuse/toolbox:latest TOOLBOX_NAME=special-debug-container TOOLBOX_SHELL="/bin/bash"
- TOOLBOX NAME: allows to tweak the basename of the toolbox-es
- REGISTRY + IMAGE allows to use a different image for your toolbox-es
 - toolbox/latestis based on Tumbleweed
 - You can have Leap toolbox-es
 - You can make toolbox-es from your (<u>Kiwi</u> / <u>OBS</u> built) images
 - You can have toolbox-es based on different distros!
 - (possible already, but needs a little more work for dealing well with -u)

Toolbox for Graphical Apps

- They work too! \Rightarrow No need installing them in base OS
- \$ toolbot -u
 - > sudo zypper in gedit virt-manager
 - > gedit
 - > virt-manager





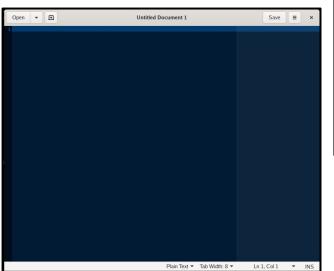
	Vi	rtual Ma	hine N	Virtual Machine Manager (on toolbox-dario-user-dev)						
0000	0000	0000	000	1						
0000							6]
		1000001 1000001								

Toolbox for Graphical Apps

- They work too! \Rightarrow No need installing them in base OS
- \$ toolbot -u
 - > sudo zypper in gedit virt-manager
 - > sudo zypper in xorg-x11-fonts-core
 - > sudo zypper in adwaita-icon-theme
 - > gedit
 - > virt-manager

Ok, now we're Talking

(are we missing some deps somewhere, maybe?)



Shutoff Vm2 Shutoff Shutoff Vm3 Shutoff Vm4 Vm4 Shutoff Vm4 Shutoff	Vame QEMU/KVM: 10.156.208.94 vm1		Memory usag
Shutoff vm4 Shutoff QEMU/KVM: 192.168.0.30 Tumbleweed	Shutoff		
✓ QEMU/KVM: 192.168.0.30 Tumbleweed	Shutoff vm4		
	▼ QEMU/KVM: 192.168.0.30		
		1 M	

Toolbox for "GL" Graphical Apps

• Kernelshark as an example:

O \$ toolbox -u

> 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...

- O \$ toolbox
 - > sudo zypper addrepo https://download.nvidia.com/opensuse/tumblewee dNVIDIA
 - > sudo zypper ref
 - > sudo zypper in x11-video-nvidiaG05
- It installs stuff like:
 - o kernel-default-devel,nvidia-gfxG05-kmp-default,nvidia-glG05 ...
 - \circ ... Inside the container ?

Toolbox for "GL" Graphical Apps

			Ker	nel Sha	ark (trace.dat) ((on toolbox-dario-user-dev)	×	
File	Filter Plots Tools	Help						
Poir	nter: 377678.945173	CRASSE.					Г	
<	. + >	++	Marke	r A		Marker B A,B Delta:	-	
	377678.945123			- 16 	37	7678.945995 37	77678.946867	Well, it works!
CP	01							
	arch: Column #	▼ Cont		•		Next Prev Graph follows	•	led
ŧ	CPU Time Stamp	Task	PID	Latency	and the second	Info	▼	led
# 0	CPU Time Stamp 132 377678.94512	Task 5 <idle></idle>	PID 0	Latency .N	funcgraph_entry	Info 2.258 us sched_ttwu_pending();	×	led
# 0 1	CPU Time Stamp 132 377678.94512 132 377678.94512	Task 5 <idle> 7 <idle></idle></idle>	PID 0 0	Latency .N .N	funcgraph_entry funcgraph_exit	Info 2.258 us sched_ttwu_pending(); 2.258 us }	×	led
# 0 1 2	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514	Task 5 <idle> 7 <idle> 2 <></idle></idle>	PID 0 0 104728	Latency .N .N d	funcgraph_entry funcgraph_exit funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() {	•	led
# 0 1 2 3	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <></idle></idle>	PID 0 0 104728 104728	Latency .N .N d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() {	•	led
# 0 1 2 3 4	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <> 5 <></idle></idle>	PID 0 0 104728 104728 104728	Latency .N .N d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054		led veedNVIDIA
# 0 1 2 3 4 5	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <> 5 <> 5 <></idle></idle>	PID 0 0 104728 104728 104728 104728	Latency .N d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us ttwu_queue_wakelist(); }		
# 0 1 2 3 4 5 6	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <> 5 <> 6 <> 7 <></idle></idle>	PID 0 0 104728 104728 104728 104728 104728 104728	Latency .N d d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry funcgraph_exit	Info 2.258 us sched_ttwu_pending(); 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us ttwu_queue_wakelist(); 0.385 us }		
# 0 1 2 3 4 5 6 7	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <> 5 <> 6 <> 7 <> 7 <></idle></idle>	PID 0 0 104728 104728 104728 104728	Latency .N N d d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry funcgraph_exit funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us ttwu_queue_wakelist(); 0.385 us } ttwu_do_activate() {		
# 0 1 2 3 4 5 6 7 8	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 3 <> 5 <> 6 <> 7 <> 7 <> 9 <></idle></idle>	PID 0 0 104728 104728 104728 104728 104728 104728	Latency .N d d d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us ttwu_queue_wakelist(); 0.385 us } ttwu_do_activate() {		
Sea # 0 1 2 3 4 5 6 7 8 9 10	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 8 <> 5 <> 6 <> 7 <> 9 <> 9 <> 9 <></idle></idle>	PID 0 0 104728 104728 104728 104728 104728 104728 104728	Latency .N d d d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry funcgraph_exit funcgraph_entry	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us twu_queue_wakelist(); 0.385 us } ttwu_do_activate() { 0.674 us ttwu_do_wakeup(); }		
# 0 1 2 3 4 5 6 7 7 8 9	CPU Time Stamp 132 377678.94512 132 377678.94512 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514 140 377678.94514	Task 5 <idle> 7 <idle> 2 <> 5 <> 6 <> 7 <> 9 <> 0 <> 0 <></idle></idle>	PID 0 0 104728 104728 104728 104728 104728 104728 104728 104728	Latency .N d d d d d d d	funcgraph_entry funcgraph_exit funcgraph_entry funcgraph_entry sched_waking funcgraph_entry funcgraph_entry funcgraph_entry funcgraph_exit	Info 2.258 us sched_ttwu_pending(); 2.258 us } wake_up_process() { try_to_wake_up() { comm=kworker/u292:2 pid=104368 prio=120 target_cpu=054 0.385 us ttwu_queue_wakelist(); 0.385 us } ttwu_do_activate() { 0.674 us ttwu_do_wakeup(); 0.674 us }		

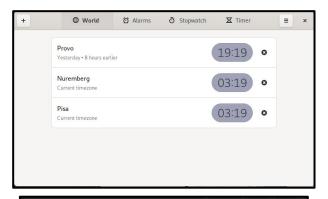
 \circ ... Inside the container ?

Remember this?

- Virtualization Specialist Sw. Eng. (a) SUSE since 2018, working on Xen, KVM, QEMU, mostly about performance related stuff
- Daily activities \Rightarrow 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)
 - o Browse Web
 - Meetings / Video calls / Online conferences
 - Chat, work and personal
 - Occasionally play games
 - Occasional video-editing
 - Maybe scan / print some document
- Can all of the above be done with MicroOS <u>already</u>?

Email, Calendaring, IM & Office Apps

- Mail, calendaring, contacts, ...
 - Evolution, <u>org.gnome.Evolution</u>
 - Calendar, <u>org.gnome.Calendar</u>
 - Contacts, <u>org.gnome.Contacts</u>
 - GNOME Clocks, <u>org.gnome.clocks</u>
 - Weather, <u>org.gnome.Weather</u>
- Documents
 - Evince, <u>org.gnome.Evince</u>
 - GNOME Documents, <u>org.gnome.Documents</u>
 - LibreOffice, <u>org.libreoffice.LibreOffice</u>
- Messaging
 - RocketChat, <u>chat.rocket.RocketChat</u>
 - Pidgin, <u>im.pidgin.Pidgin</u>
 - Telegram, <u>org.telegram.desktop</u>
 - Signal, <u>org.signal.Signal</u>

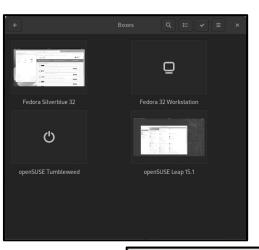


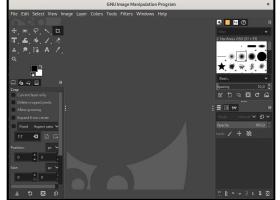


Editors, Tools, Graphics

• Editors:

- Vim, <u>org.vim.Vim</u>
- Gedit, <u>org.gnome.gedit</u>
- Setzer, <u>org.cvfosammm.Setzer</u>
- Eclipse, <u>org.eclipse.Java</u>
- Graphics
 - GIMP, <u>org.gimp.GIMP</u>
 - Krita, <u>org.kde.krita</u>
 - Blender, <u>org.blender.Blender</u>
- VMs:
 - GNOME Boxes, <u>org.gnome.Boxes</u>
- Tools:
 - Regex Tester, com.github.artemanufrij.regextester
 - Meld, <u>org.gnome.meld</u>
 - Boop-GTK,<u>uk.co.mrbenshef.Boop-GTK</u>



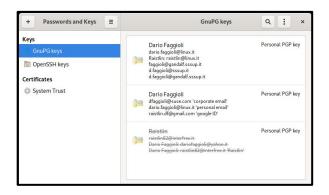


Utilities, Configuration

• Misc utilities:

- SyncThing, <u>me.kozec.syncthingtk</u>
- Barrier, <u>com.github.debauchee.barrier</u>
- Seahorse, <u>org.gnome.seahorse.Application</u>
- Config:
 - Dconf Editor, <u>ca.desrt.dconf-editor</u>
 - Flatseal, com.github.tchx84.Flatseal
 - GPU-Viewer, io.github.arunsivaramanneo.GPUViewer

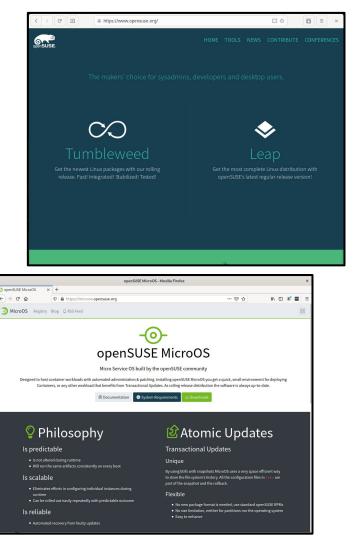




	Applications	■ Show Details	Flatseal		×
 S D S S	Dconf Editor ca.desrt.dconf-editor Rocket.Chat chatrocket.RocketChat Regex Tester com.github.artemanufrij.regexte Barrier com.github.adebauchee barrier	Share	Flatseal Martin Abente Lahaye Version 1.6.2 Last Updated agosto 28 Runtime org.gnom	5, 2020 e.Platform/x86_64/3.3	36
5_	EasySSH com.github.muriloventuroso.eas	Network share=network			
0	Flatseal com.github.tchx84.Flatseal	Inter-process con share=ipc	nmunications		D
4	Microsoft Teams com.microsoft.Teams Skype	Socket List of well-known s	ockets available in the sandbo	Х	
5	com.skype.Client Steam	X11 windowing sy socket=x11	stem		C
	com.valvesoftware.Steam Remotely de.haeckerfelix.Remotely	Wayland window socket=wayland	ing system		D
		Fallback to X11w	indowing system	_	

Browsing

- Firefox, from the Flatpak (<u>org.mozilla.firefox</u>)
 - Works great, including video codecs (and without having to add Packman repos)
- Epiphany (GNOME Web, <u>org.gnome.Epiphany</u>)
- Chrome[ium]
 - There is no Flatpak for that yes (but no, but <u>it's being worked on</u>)
 - Installed in the base OS, with
 Transactional-update(and reboot)
- NB: GNOME Shell Extension can't be installed from a "Flatpak-ed" browser yet
 - You probably need at least one browser in the base OS (I have Chrome)



Gaming

- Steam, <u>com.valvesoftware.Steam</u>
 - Works great, even SteamPlay/Proton

• NVIDIA Drivers

- \$ sudo transactional-update shell
 - # zypper ar --refresh <u>https://download.nvidia.com/opensuse/tumbleweed</u> NVIDIA
 - # 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 org.freedesktop.Platform.GL32.nvidia-450-66





Video: Viewing, Editing & Codecs

- Remember: we did not add Packman
- VLC, <u>org.videolan.VLC</u>
 - \circ Has the proper codecs
- Pitivi, <u>org.pitivi.Pitivi</u>
 - \circ Has the proper codecs
- Openshot, <u>org.openshot.OpenShot</u>
 - \circ Has the proper codecs
- Cheese, <u>org.gnome.Cheese</u>
 - Works well with my webcam

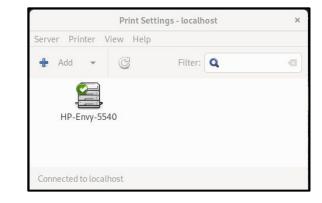


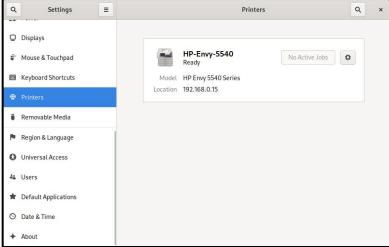


Printing & Scanning

• Printing

- By default: no cups, no PPDs, ...
- Tried installing (transactional-update)
- It works!
- OBS request <u>840921</u>
- Should just work for everyone now
- Scanning
 - By default: no sane packages
 - Tried installing (transactional-update)
 - Flatpak apps (e.g., Paper) don't work yet
 - Still working on it
 - (yeah, most scanners, e.g., from all-in-one printers, have Web-ish interface. But still)





Remember this?^2

- Virtualization Specialist Sw. Eng. (a) SUSE since 2018, working on Xen, KVM, QEMU, mostly about performance related stuff
- Daily activities \Rightarrow how and what for I use my workstation
 - Read and send emails (Evolution, git-send-email, stg mail, ...) **Check**
 - Write, build & test code (Xen, KVM, Libvirt, QEMU)
 - Work with the Open Build Service (OBS)
 - \circ Browse Web
 - Meetings / Video calls / Online conferences
 - Chat, work and personal
 - Occasionally play games
 - Occasional video-editing
 - Maybe scan / print some document

Check Check Check Check Check Check

• Can all of the above be done with MicroOS <u>already</u>?

Hacking On, E.g., QEMU

• Dependencies for building <u>QEMU</u> from sources:

- O bc bison bluez-devel brlapi-devel bzip2 ccache clang cyrus-sasl-devel flex gcc gcc-c++ gettext-tools git glib2-devel glusterfs-devel gtk3-devel gtkglext-devel gzip hostname libSDL2-deve libaio-devel libasan4 libcap-devel libcap-ng-devel libcurl-devel libfdt-devel libgcrypt-devel libgnutls-devel libjpeg62-devel libnettle-devel libnuma-devel libpixman-1-0-devel libpng16-devel librbd-devel libseccomp-devel libspice-server-devel libssh-devel libssh2-devel libtasn1-devel libudev-devel libxml2-devel lzo-devel make makeinfo multipath-tools-devel ncurses-devel perl pkg-config python3 python3-PyYAML python3-Sphinx rdma-core-devel snappy-devel sparse tar usbredir-devel virglrenderer-devel vte-devel which xen-devel zlib-devel
- You don't want to install them with transactional-update and reboot
- Oh, you forgot one / there is a new one needed:
 - Install with transactional-update and reboot again?
- Do try! I promise that it **won't** be funny :-/
- Toolbox to the rescue:
 - O \$ toolbox -u -t dev # -r may or may not be needed. Generally not for building
 > sudo zypper in <all the dependencies above>
 - > cd <your QEMU sources directory in your home (it's there in the toolbox!)>
 - > <do your changes>
 - > <build it>

Working With OBS

Requires installing packages, using VMs for building, etc.

- toolbox, what else ?!
- I need a -r one, for mounting filesystems in the build VM (I think)

552 dario

20

```
$ toolbox -u -r -t dev
 > zypper ar <a href="https://download.opensuse.org/[...]/openSUSE Tumbleweed/openSUSE:Tools.repo">https://download.opensuse.org/[...]/openSUSE Tumbleweed/openSUSE:Tools.repo</a>
 > zypper in cpio osc build [...]
 > osc mkpac / co / vc
 > [...]
 > osc vc
 > osc build --vm-type=kvm
                                                Swp
 > osc commit
                                                PID USER
Building outside of VMs
                                                1150 demu
                                                             20
                                                                  0 8589M 4149M 23564 R 87.7 13.1 2:32.26 /usr/bin/gemu-system-x86_64 -machine accel=kvm
                                                1151 gemu
                                                             20
                                                                  0 8589M 4149M 23564 R 83.9 13.1 2:48.71 /usr/bin/qemu-system-x86_64 -machine accel=kvm -nodefa
currently not working
                                                             20 0 8589M 4149M 23564 R 83.3 13.1 2:57.36 /usr/bin/gemu-system-x86_64 -machine accel=kvm -nodefa
                                                1147 gemu
                                               1149 gemu
                                                                  0 8589M 4149M 23564 R 82.6 13.1 2:31.65 /usr/bin/gemu-system-x86_64 -machine accel=kvm -nodefa
                                                             20
        (but it's better to
                                                1148 gemu
                                                             20
                                                                 0 8589M 4149M 23564 R 80.1 13.1 2:30.10 /usr/bin/gemu-system-x86_64 -machine accel=kvm -nodefa
                                               1146 gemu
                                                             20
                                                                 0 8589M 4149M 23564 R 80.1 13.1 2:34.15 /usr/bin/gemu-system-x86 64 -machine accel=kvm -nodefa
        build In VMs anyway...)
                                               1158 gemu
                                                                 0 8589M 4149M 23564 R 79.5 13.1 2:32.78 /usr/bin/gemu-system-x86_64 -machine accel=kvm -nodefa
                                                             20
                                               1157 gemu
                                                                  0 8589M 4149M 23564 R 73.2 13.1 2:50.52 /usr/bin/gemu-system-x86_64 -machine accel=kvm -nodefa
                                                             20
                                                                               228M S 32.2 1.3 58:07.55 /opt/google/chrome/chrome --type=renderer --field-tria
                                               6197 dario
                                                             20
                                                                  0 4933M
```

0 659M 196M 94584 S 12.6 0.6 20:17.32 /opt/google/chrome/chrome --type=gpu-process --field-t

Working on Libvirt and QEMU

Real scenario:

- I make a change in QEMU
- I make a change in Libvirt
- I want to build **and also test** my changes

How it works for me:

- 1. I work on the changes themselves inside my development toolbox
- 2. Still in there, I start my modified libvirtd, make it listed on TCP (no socket activation)

```
$ toolbox -r -u -t dev
$> <work on QEMU> && <build QEMU> && <install my QEMU>
$> <work on libvirt> && <build libvirt> && <install my libvirt>
$> sudo ./build/src/virtlogd &
```

- \$> sudo ./build/src/libvirtd -v -l
- 3. From (either the same or a different) toolbox I start virsh and/or virt-manager and connect to my modified libvirtd

Working on Libvirt and QEMU

	Qualere libuirt ait		
	(a)sichibvire.gie	Virtual Machine Manager (on toolbox-dario-us	er) ×
Tumbleweed on QEMU/KVM: 192.168.0.30 (on toolbox-dario-user) Tumbleweed on QEMU/KVM: 192.168.0.30 (on toolbox-dario-user) File Virtual Machine View Send Key C	<pre>@:-/src/libvirt.git X E_TIMEOUT: timer=2 interval=5000 _HANDLE: watch=10 events=1 ATCH_HANDLE: watch=10 events=1 LCLIENT_MSG_RX: client=0x7fa3dc0 E_HANDLE: watch=10 events=1 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>	File Edit View Help	z-manager running in my
VM started by virt-manager in the `toolbox -u`. It's actually running inside `toolbox -r -u -t dev, using my modified Libvirt and QEMU	ATCH_HANDLE: watch=10 events=2 c HANDLE: watch=10 events=1 ATCH_HANDLE: watch=10 events=1 LANDLE: watch=10 events=1 LIMEOUT: timer=2 interval=5000 HANDLE: watch=10 events=1 	vm-leap04 `toc	blbox -u` and connecting libvirtd in the other box
20 2020-10-15 09:45:39.423+0000: 12142: 1nto : vir0bjectRef:403 : OBJECT_REF: obj=0x7ta33 2020-10-15 09:45:39.423+0000: 12142: info : vir0bjectRef:403 : OBJECT_UNREF: obj=0x7ta33 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:403 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:238 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:238 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:239 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:238 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:381 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:330 : OBJECT_REF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:331 : OBJECT_UNREF: obj=0x7ta32 2020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:432 : OBJECT_UNREF: obj=0x7ta32 0020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:432 : OBJECT_UNREF: obj=0x7ta32 0020-10-15 09:45:39.424+0000: 12142: info : vir0bjectRef:432 : OBJ	Lerato Lerato	vm-leap11 Shutoff ↓ OEMU/KVM.192168.0.30 ↓ DEMU/KVM.192168.0.30 ↓ Tumbleweed Running t=0x7fa3dc013170 len=440 prog=536903814 vers=1 proc=344 type= b=0x7fa42b278600 opaque=0x561767e308d0	=l status=0 serjal=311

A Day in the Life of a Developer who Uses MicroOS as Workstation...

- I hacked on toolbox in such a way that:
 - With toolbox -u and/or toolbox -r -u:
 - You have your user inside the toolbox
 - You have your home, 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
- Also:
 - With -t, you can have multiple toolbox-es, e.g.:
 - One per each project you're working on?
 - One for work projects and one for home projects?
 - One for
- IOW: It's a quite cool development environment
 - I adopted it even on Tumbleweed, <u>before</u> moving to MicroOS!

A Day in the Life of a Developer who Uses MicroOS as Workstation...

My morning routine:

- 1. Wake-up / wake-up the kids / have breakfast with them / bring them to school ;-P
- 2. Brew some more coffee
- 3. Open gnome-terminal
- 4. Enter a toolbox -r -u -t dev (brings me inside toolbox-dario-user-dev)
- 5. Start tmux inside that toolbox
 - a. all panes will be inside the toolbox already!
 - b. Stay in there until end of day
- 6. Maybe, enter my toolbox -u (brings me inside toolbox-dario-user)
 - a. Use some apps from there that I need but don't want to install in the base OS
- 7. <<Hey network to the office seems slow!>>
 - a. \$ toolbox -r
 - *#>* zypper in traceroute
 - #> traceroute <u>www.suse.com</u>

 Image: constraint of the second sec

Some Stats

• RPM Packages

- On my MicroOS Desktop: ~1000
 - But I've done a few experiments, added stuff, ...
- In a development toolbox on my MicroOS Desktop: ~1300
 - No Desktop Environment packages
 - But with some GUI apps & libs
- On a stock Fedora Silverblue: ~1200
- \circ $\,$ On a Tumbleweed box I also have: ~3500 $\,$
 - Not used for development (so no -devel pkgs)
 - A few apps as flatpak there as well
- Flatpaks
 - Apps installed: 68
 - All flatpaks (including runtimes): 110
 - Disk space: 12 GB

Example: Nautilus, Trash, USB Keys, From "not working" to "it works!"

Music
 Pictures
 Videos
 Trash
 Sources

🛛 efi

P etc

home

P i386-pc

D local

O opt

o root

■ srv

O tmp

var

writable

x86_64-efi

+ Other Locations

.snapshots

🖸 Filesystem root 🔺

۰

۲

.

.

.

۰

.

.

.

۸

.

.

.

Problem:
 Nautilus was looking weird (showing all BTRFS subvolumes, etc)
 Trash was not working
 Files going in .local/share/Trash
 Not being shown when clicking on "Trash" icon
 USB keys not being (auto)mounted, /run/media/<user>not appearing</user>
Let's try something
 Mounting USB keys in /run/<user>/<volume>⇒ it's udisks2</volume></user>
• On a Tumbleweed:
o ps aux grep udisk ⇒
/usr/libexec/gvfs/gvfs-udisks2-volume-monitor
/usr/libexec/udisks2/udisksd
o rpm -qf⇒
gvfs-backends-1.44.1-2.4.x86_64
udisks2-2.8.4-1.3.x86_64

Example: Nautilus, Trash, USB Keys

- Let's fix it!
 - \$ sudo transactional-update pkg in gvfs-backends udisks2
 - \$ sudo reboot
- It works!
- OBS request <u>840921</u>
- Should just work for everyone now

< >	■ 2,1 GB Volume 🔻		
⊘ Recent			
🗙 Starred			
🔂 Home	SUSEcon_Poster_	SUSEcon_Poster_	SUSEcon_Poster_
Documents	Bring_Me_the_Data.	Flap_Neck_Jazz_	Paint_it_Green_1.pdf
↓ Downloads	pdf 2,2 MB	Quartet.pdf 4,3 MB	3,0 MB 16:16
🛛 Music	16:16	16:15	
Pictures			
Videos			
🗑 Trash			
🚍 2,1 GB Volume 🔺			
👤 raistlin.df@gmail			
✿ SOURCES			
+ Other Locations			

Conclusions

• Using MicroOS as a Desktop / Workstation is already possible, IME

- Requires some manual fiddling with configurations, but it's mostly something done right after install and then forgotten
- It's pretty comfortable to use
 - In fact, I started using it just as an experiment. But I'm definitely staying!
- It pushes you to do things properly
 - No quick-&-dirty hacks, like symlinking that library to make that other app work
 - Results is a much cleaner and stable system
- It's not perfect yet:
- It asks for a password too many times, post install manual config steps should be done automatically, we may want to have a GUI way for updating the base OS (like Silverblue does), etc.
- It needs you! As a user, as a tester, as a contributor, as an "evangelist", as... Well, <u>whatever you want to do, you're welcome</u>!

About Myself

- Ph.D on Real-Time Scheduling, <u>SCHED_DEADLINE</u>
- 2011, Sr. Software Engineer (a) Citrix <u>The Xen-Project</u>, hypervisor internals, NUMA-aware scheduler, Credit2 scheduler, Xen scheduler maintainer
- 2018, Virtualization Software Engineer @ <u>SUSE</u> Still Xen, but also <u>KVM</u>, <u>QEMU</u>, <u>Libvirt</u>; Scheduling, VM's virtual topology, performance evaluation & tuning









All text and image content in this document is licensed under the Creative Commons Attribution-Share Alike 4.0 License (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these thereof is subject to trademark policy.