✨ Member-only story

Using the VSCode flatpak distribution

Radu Zaharia · Follow

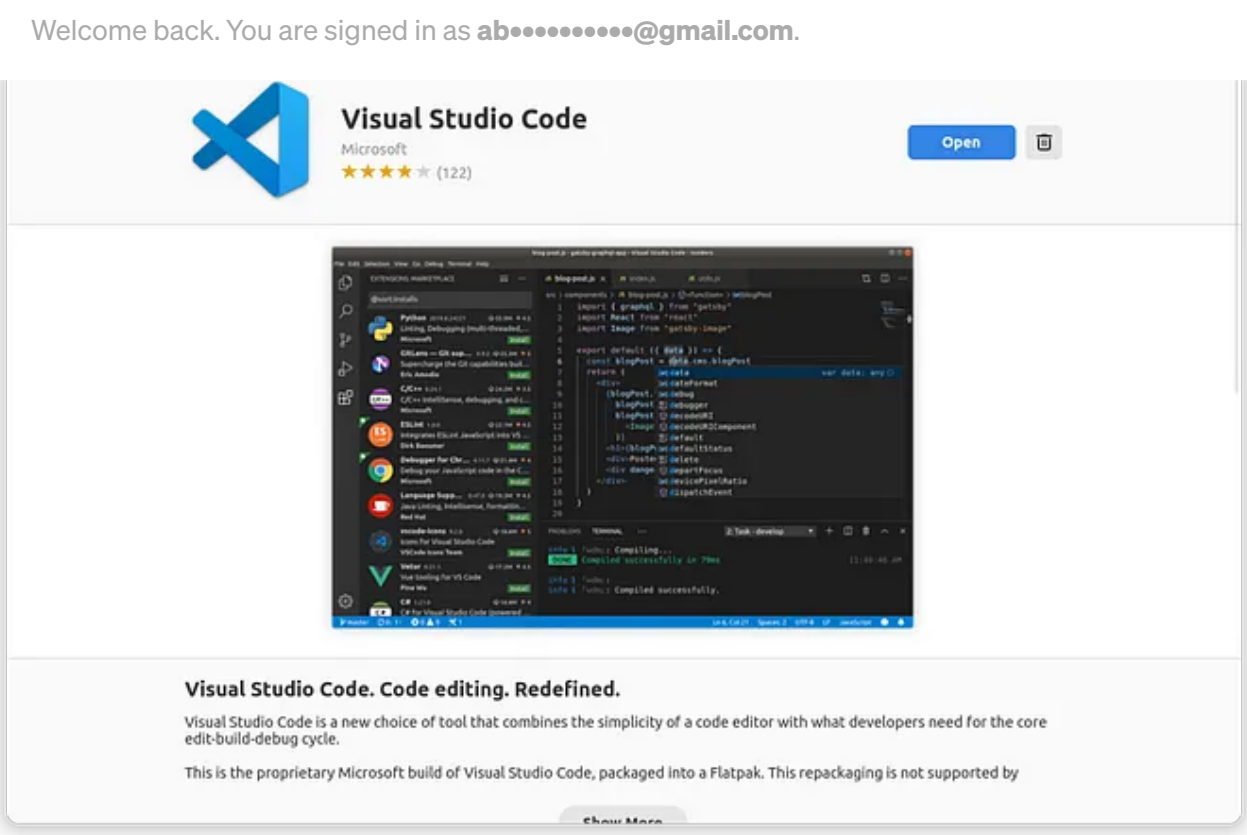
7 min read · Jun 23, 2022

Photo by Gary Meulemans on Unsplash

I am in the middle of an experiment trying to run all Fedora apps as flatpaks (don't ask). Everything works great so far but I did have a bit of trouble with VSCode and I wanted to share how I made everything work. It's not a problem with VSCode, nor with flatpak, it's just how flatpak works and expects things to work.

I will present a few ways in which you can make VSCode work as if it were a non-flatpak application. This requires a bit of configuration but once you do it everything works seamless. The configuration is also not really the fault of flatpak, nor of VSCode. It's just something you need to do when you are using an isolated app trying to make its way to the required development libraries. So let's start from the beginning.

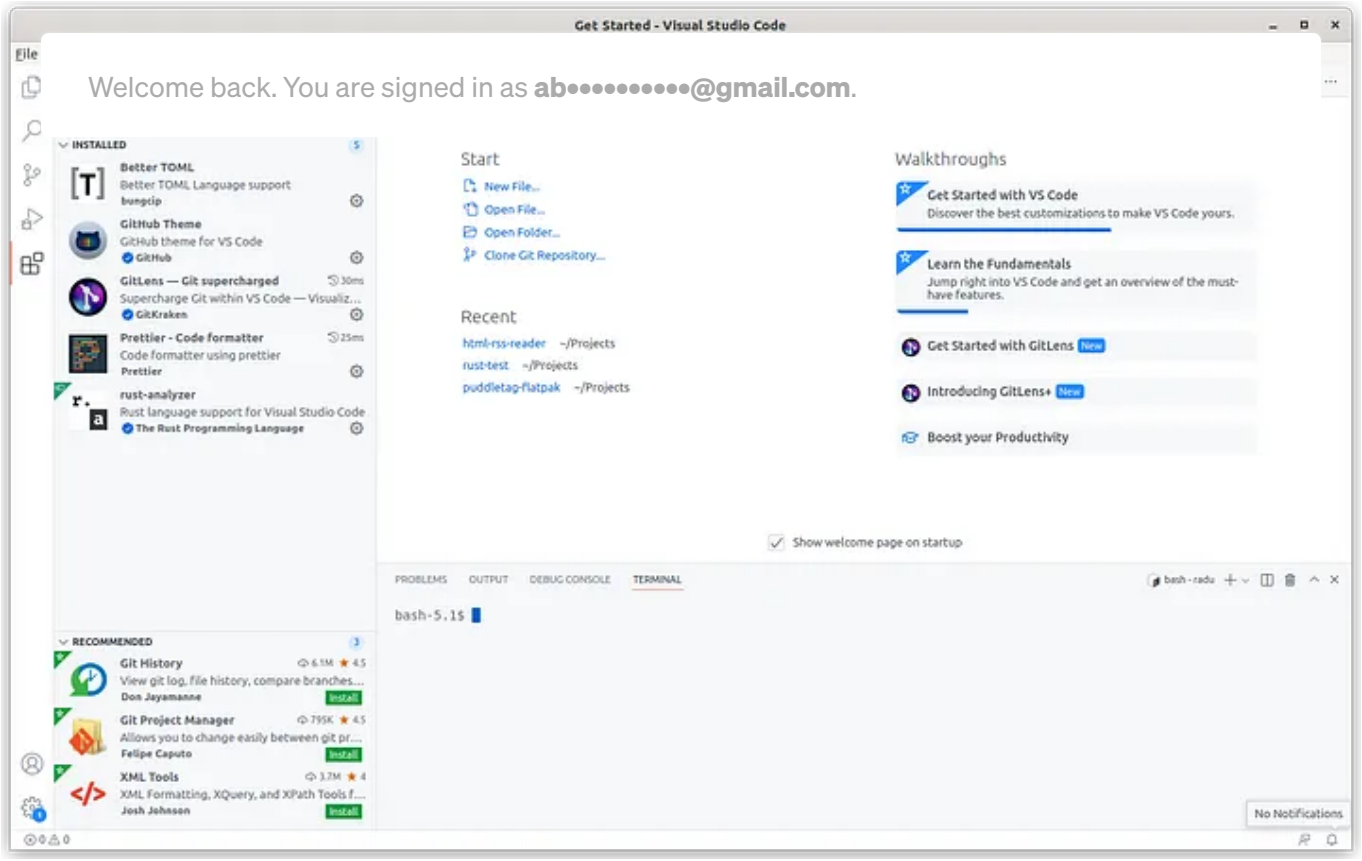
Installing the VSCode flatpak



The VSCode flatpak page in Gnome Software

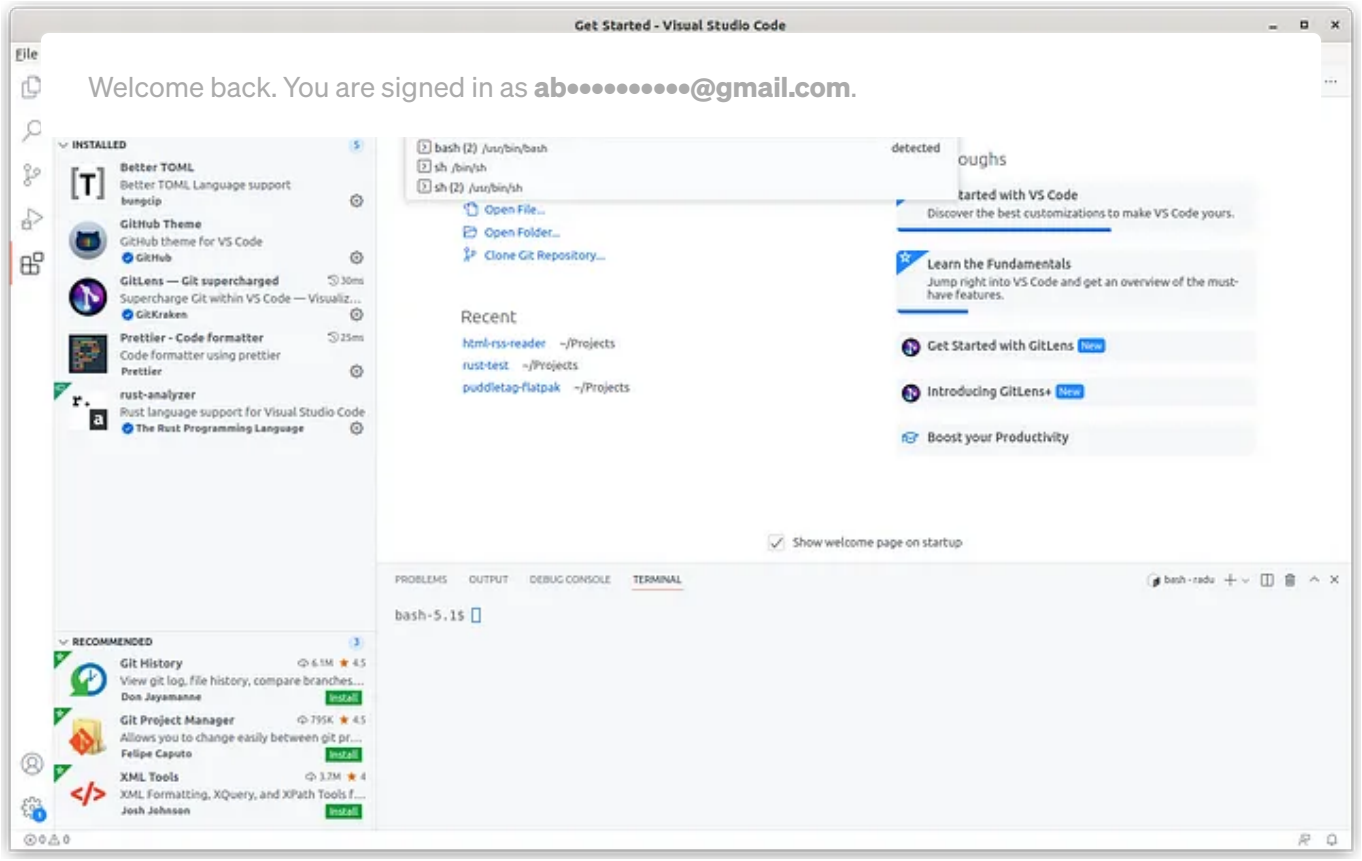
Installing the VSCode flatpak is easy: it's available from the Fedora Flathub Selection repository and can be found by searching "code" in Gnome Software without hassle. Next we click on the blue Install button and we are done.

Working with Rust a lot, I usually install some extensions helping me with that: `rust-analyzer`, `prettier` just in case, `Better Toml` to handle the `Cargo.toml` configuration file and `GitLens`. I usually set my VSCode theme to `Github Light` as that's how I got used to.



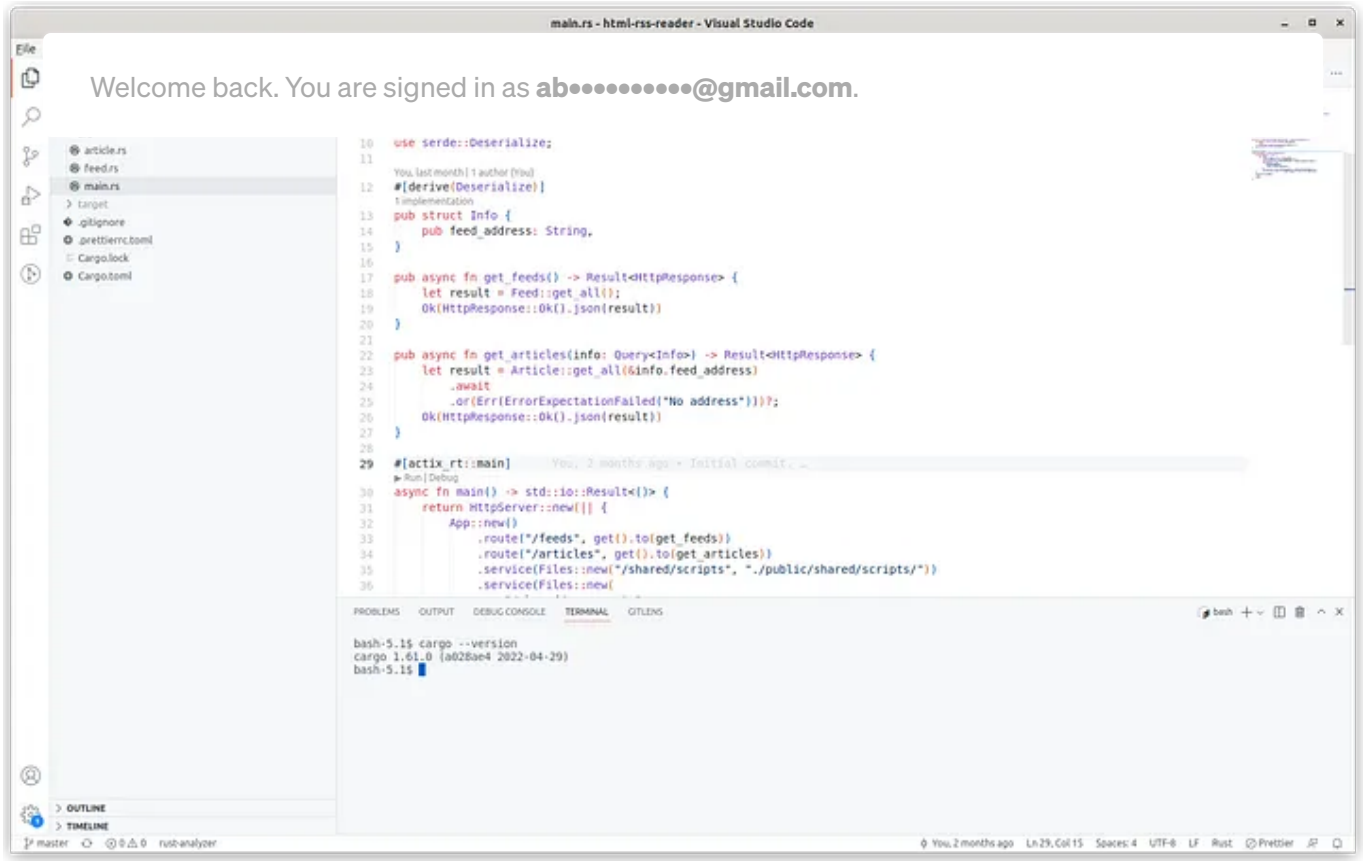
Running the VSCode flatpak with some extensions installed

If you look at the terminal screen, you will see that it does not have the usual terminal appearance. By default it will use `sh` instead of `bash` but that can be easily fixed: just go to the open terminal drop-down (the plus icon at the top right of the terminal tab) and click on “Select Default Profile”:



Making bash the default terminal shell

Developing using flatpak SDKs

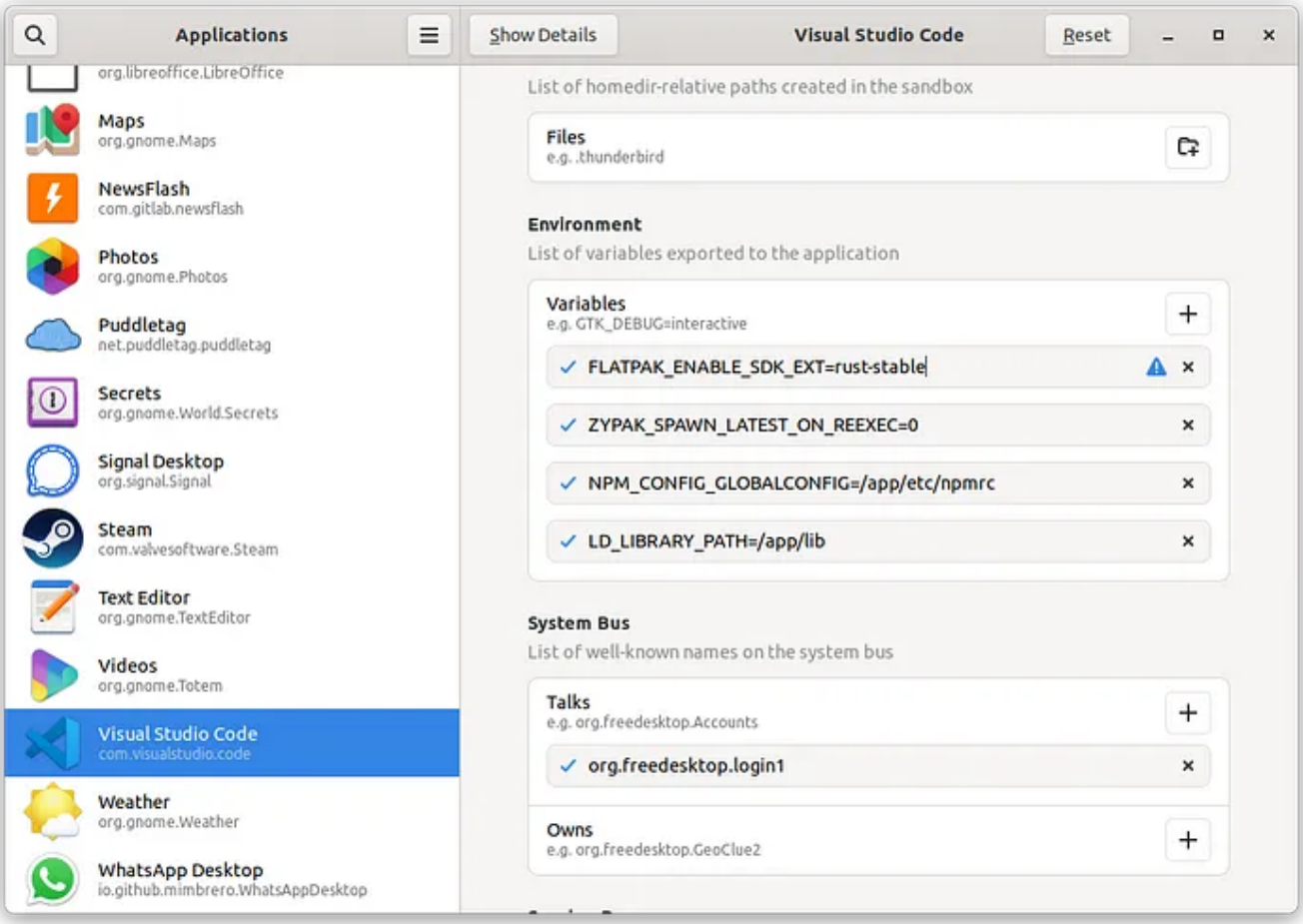


The HTML5 RSS Reader app loaded in VSCode flatpak

Now, for example if you installed `cargo` with `sudo dnf install cargo` in your usual Linux terminal, you will notice that calling `cargo` in the VSCode terminal will not work. It will simply complain that `cargo` does not exist. Sure enough, if we look for `cargo` in `/usr/bin` in the VSCode terminal it will not be there. That's because the flatpak `/usr/bin` contains sandboxed binaries rather than the full `/usr/bin` found in the host system.

This also cannot change. If we want to install `cargo` by running `sudo dnf install cargo` in the VSCode terminal, it will complain that `sudo` does not exist. Same for `dnf`. There is simply no way to install `cargo` and Rust this way. What we need to do is install the flatpak Rust SDK in the normal Linux terminal:

This will prompt us to select the Rust version we want (I just pick the latest) and install the Rust SDK flatpak, which will also make cargo available for the VSCode flatpak. The SDK will be installed at `/usr/lib/sdk/rust-stable` just so you know how different the location is related to a normal `cargo` install. We need to enable the Rust SDK extension for VSCode:



Enabling the Rust SDK extension in Flatseal

Basically we add `FLATPAK_ENABLE_SDK_EXT=rust-stable` to the Enviroment Variables section in Flatseal. Now if we type `cargo` in the VSCode flatpak terminal, it will finally run `cargo`.

This is great and all but largely insufficient. Yes, there are SDKs for Rust .N

which may not exist for your development language. And there's more: you got Rust to work, how about the necessary libraries to write GTK applications? They require the `libadwaita-devel` package which you cannot install with flatpak.

Developing using a toolbox container

Photo by MealPro on Unsplash

For more involved development you will need to rely on containers. A `toolbox` container will allow us to install whatever libraries we want and use them from the VSCode flatpak. To create a toolbox container we simply use `toolbox create my-container-name` and to enter it we use `toolbox enter my-container-name`.

Once in the container, we can install everything from `cargo` `libadwaita-`

de a container for each development language or a container for each app we want to develop. The way we set this up is limited only by our requirements and imagination.

After the container is created, using the VSCode flatpak we can connect to it in two ways: we either setup an SSH server in the container and use the Remote SSH extension to connect to it, or we use the Remote Containers extension to enter the container directly.

Connecting VSCode to the container using SSH

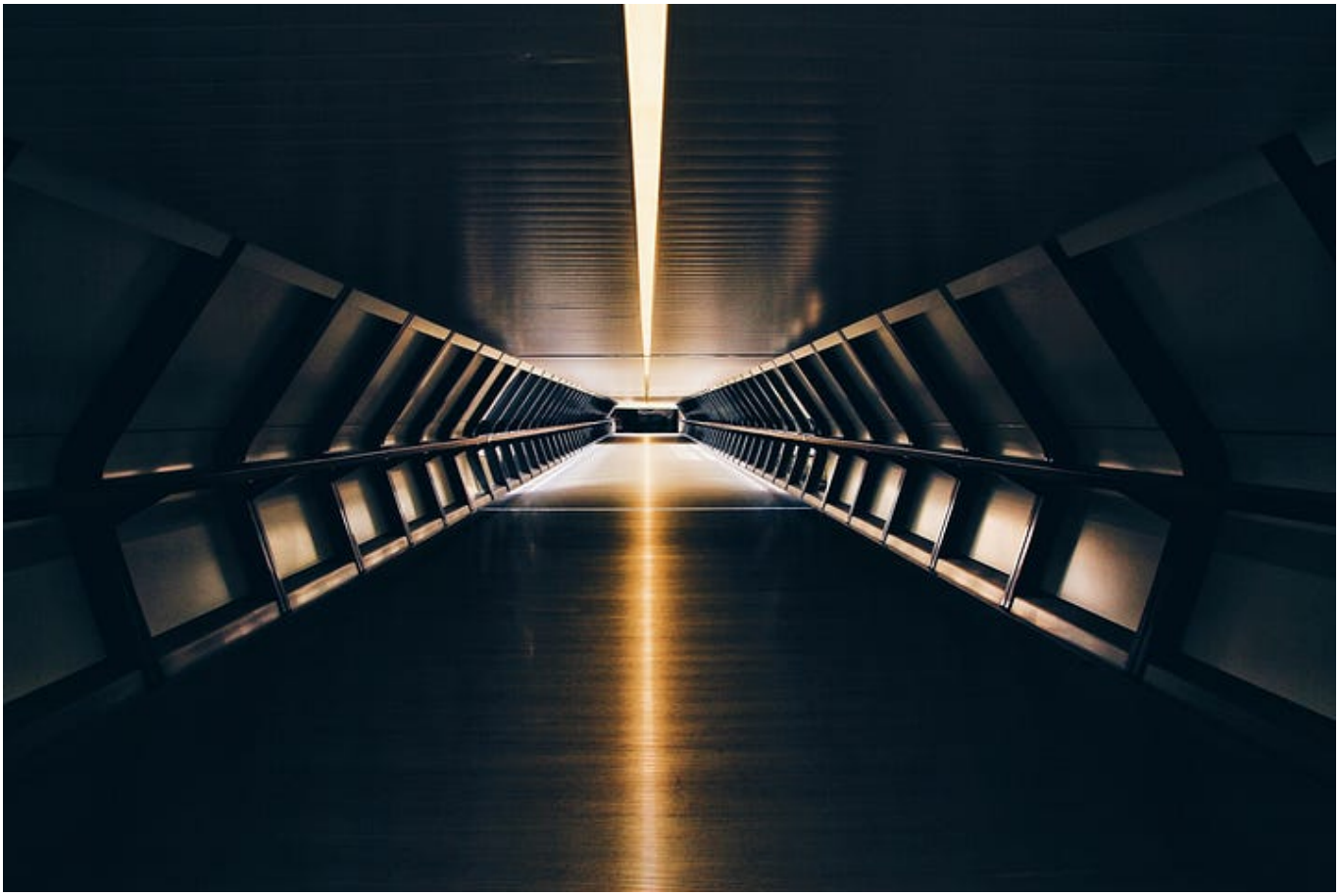


Photo by MontyLov on Unsplash

We need to install and run the SSH server in the container first:

```
my-container-name# sudo /usr/libexec/openssh/sshd-keygen rsa
my-container-name# sudo /usr/libexec/openssh/sshd-keygen ecdsa
my-container-name# sudo /usr/libexec/openssh/sshd-keygen ed25519
```

We configure the SSH server by editing the usual `sshd_config` file in `/etc/ssh`:

```
Port 10000
ListenAddress localhost
PermitEmptyPasswords yes
PermitUserEnvironment yes
X11Forwarding yes
```

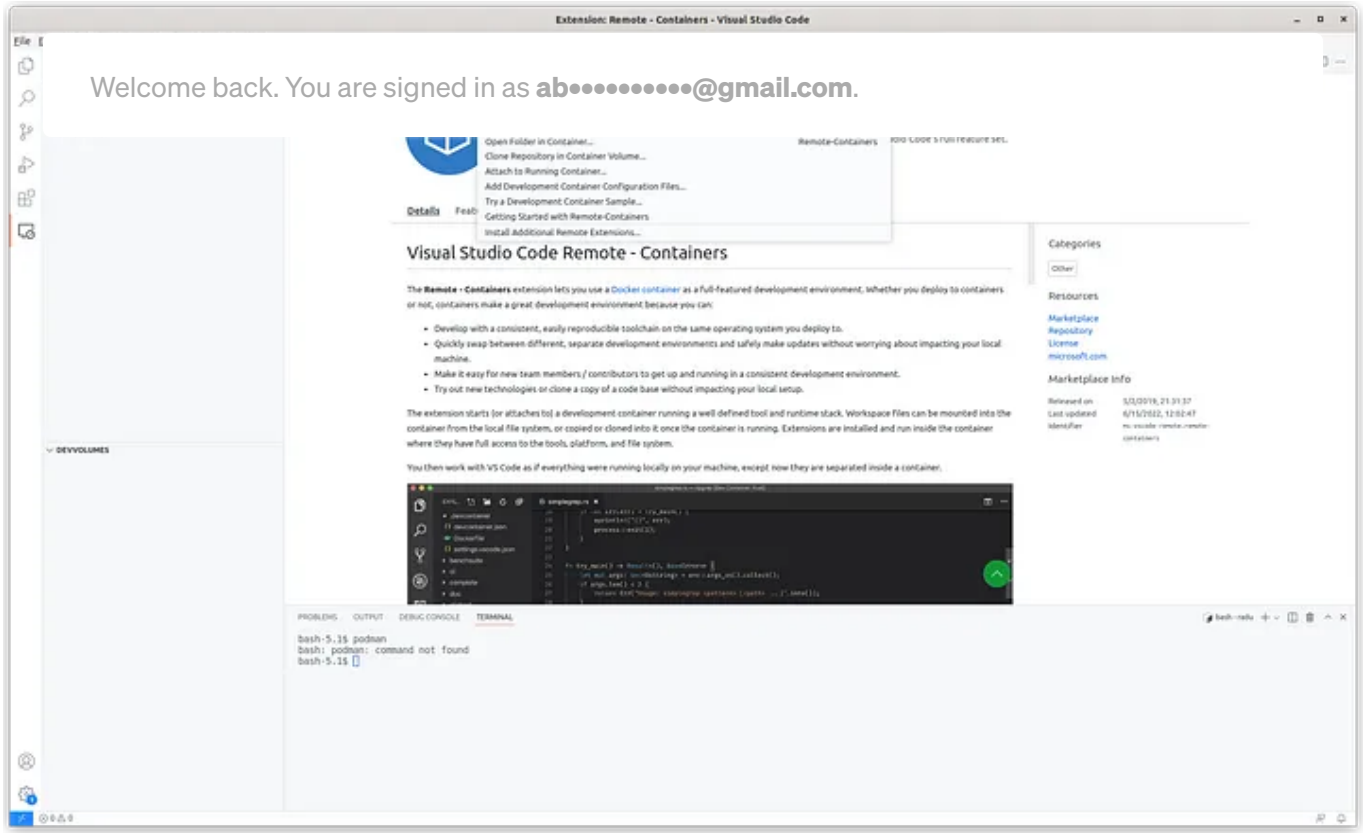
We also need to set the default display in `.ssh/environment`:

```
DISPLAY=:0
```

And then we start the SSH server:

```
my-container-name# sudo /usr/bin/sshd
```

Now that the SSH server is running, we can exit the container, start VSCode and configure the Remote SSH extension to connect to our container SSH server. We do this by clicking on the bottom left blue Remote SSH button and selecting “Open SSH configuration file…”:



Configuring the Remote SSH VSCode extension

We open the default SSH configuration file and we add our container SSH settings:

```
Host my-container-name
 HostName localhost
 Port 10000
```

Now we can connect to it by clicking the bottom left Remote SSH button and selecting “Connect to host…” and then picking `my-container-name` from the list.

Connecting VSCode to the container directly

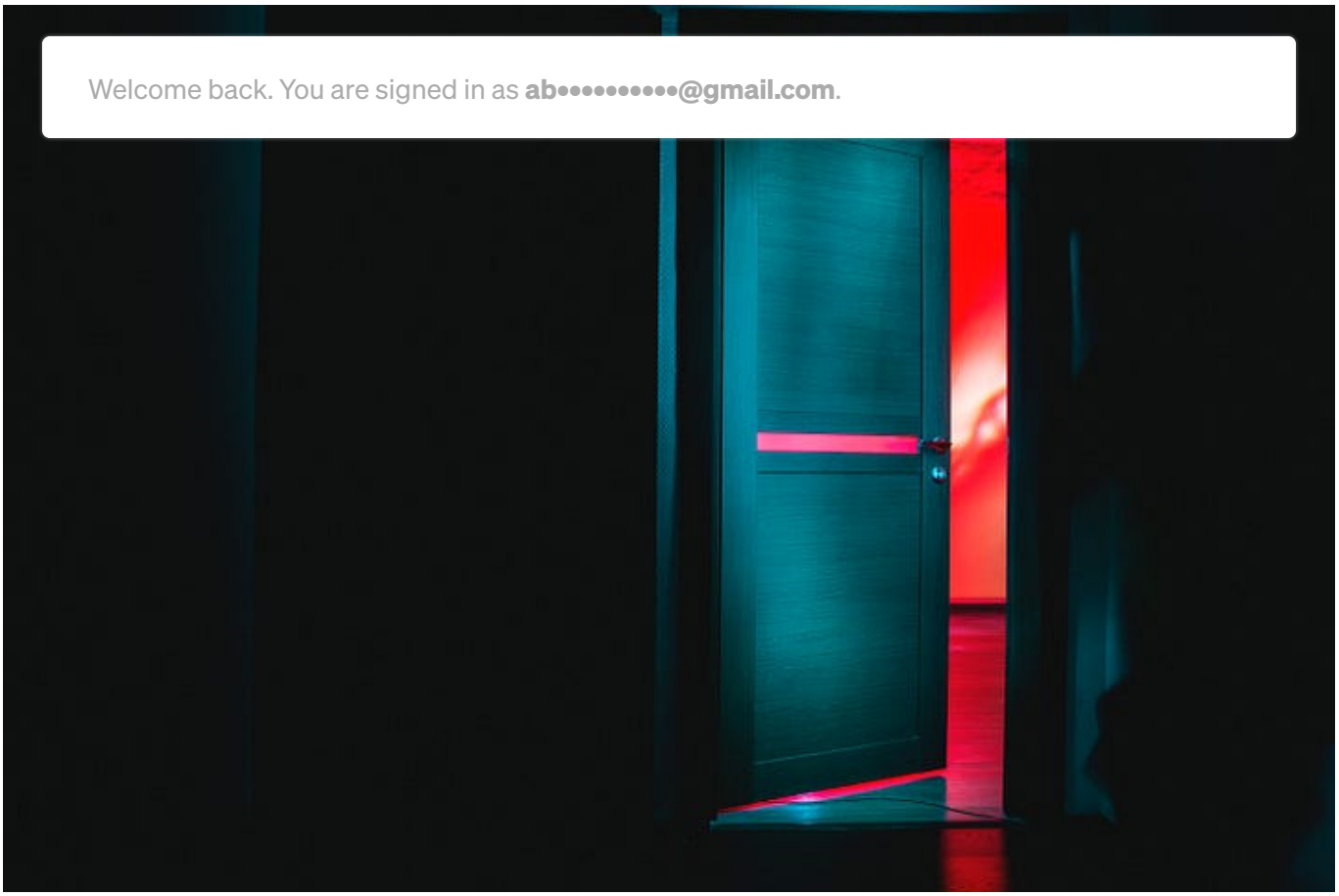
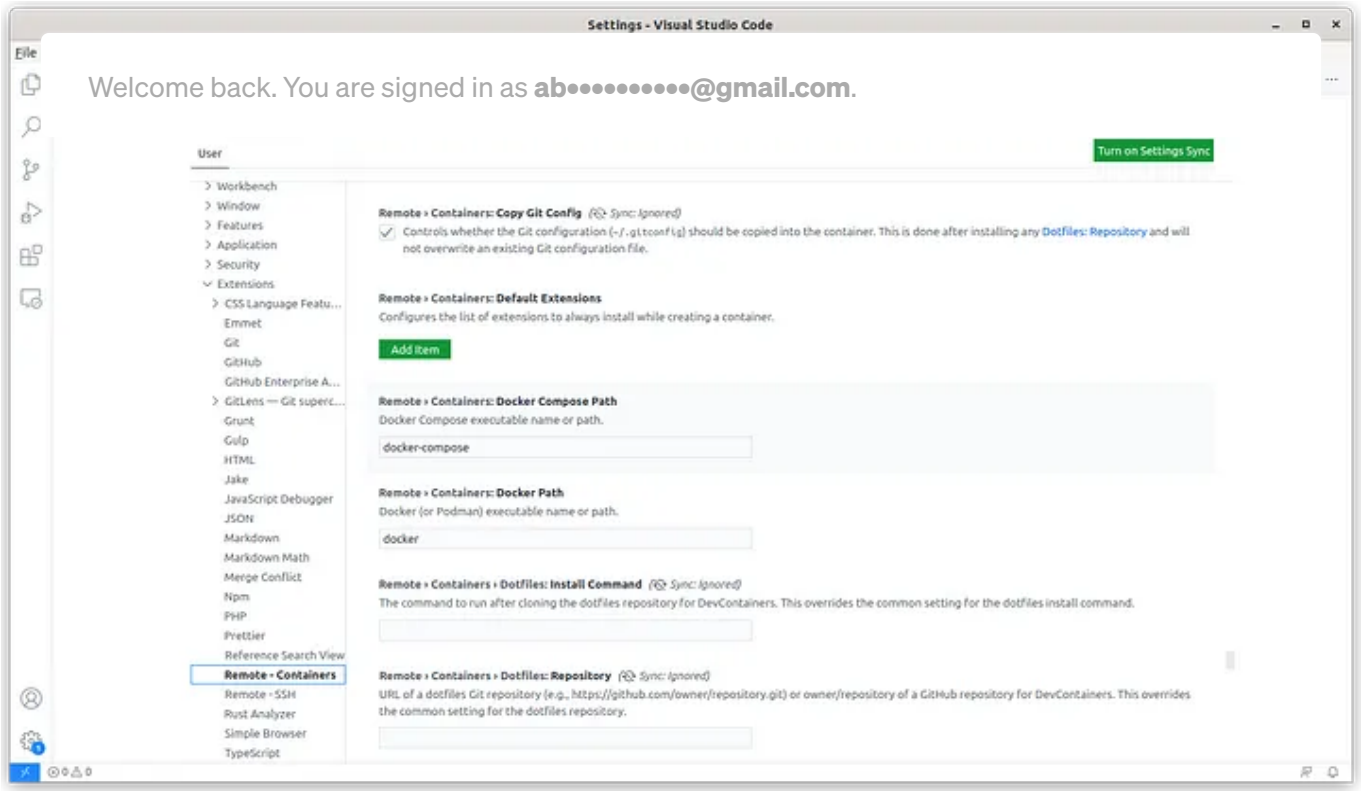


Photo by Dima Pechurin on Unsplash

Instead of relying on an SSH connection we can enter the container directly using the Remote Container extension. To do this, we must first configure the extension to make it use `podman` (`toolbox` is a wrapper for `podman`). We open VSCode Settings and go to the Remote Container extension and search for Docker Path:

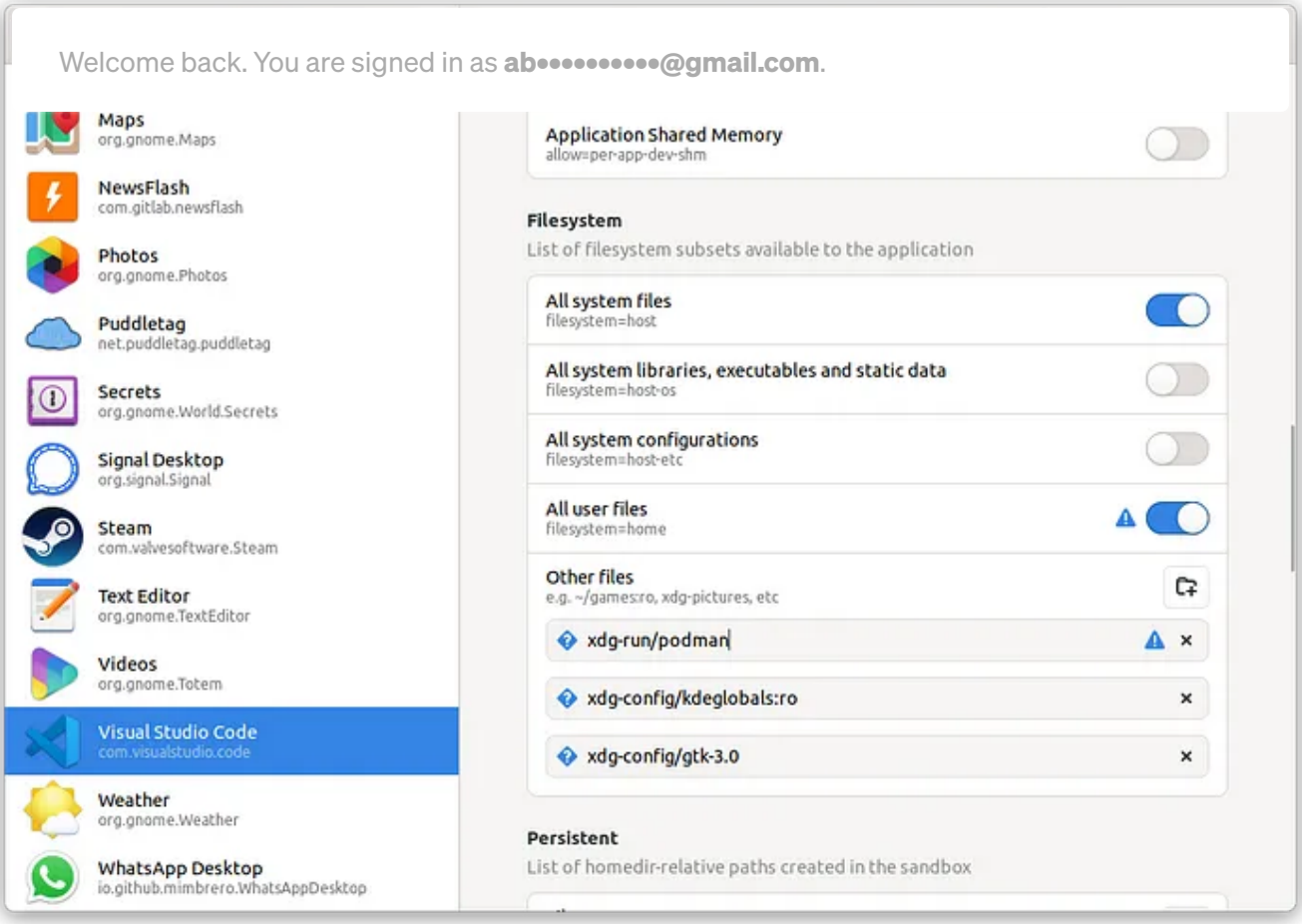


Configuring the Remote Container extension to use podman

We replace the `docker` executable with `/app/tools/podman/bin/podman-remote` and close the Settings page. We need to install the `podman` flatpak:

```
#flatpak install flathub com.visualstudio.code.tool.podman
```

And make it accessible to the VSCode flatpak by adding it to the Filesystem settings in Flatseal:

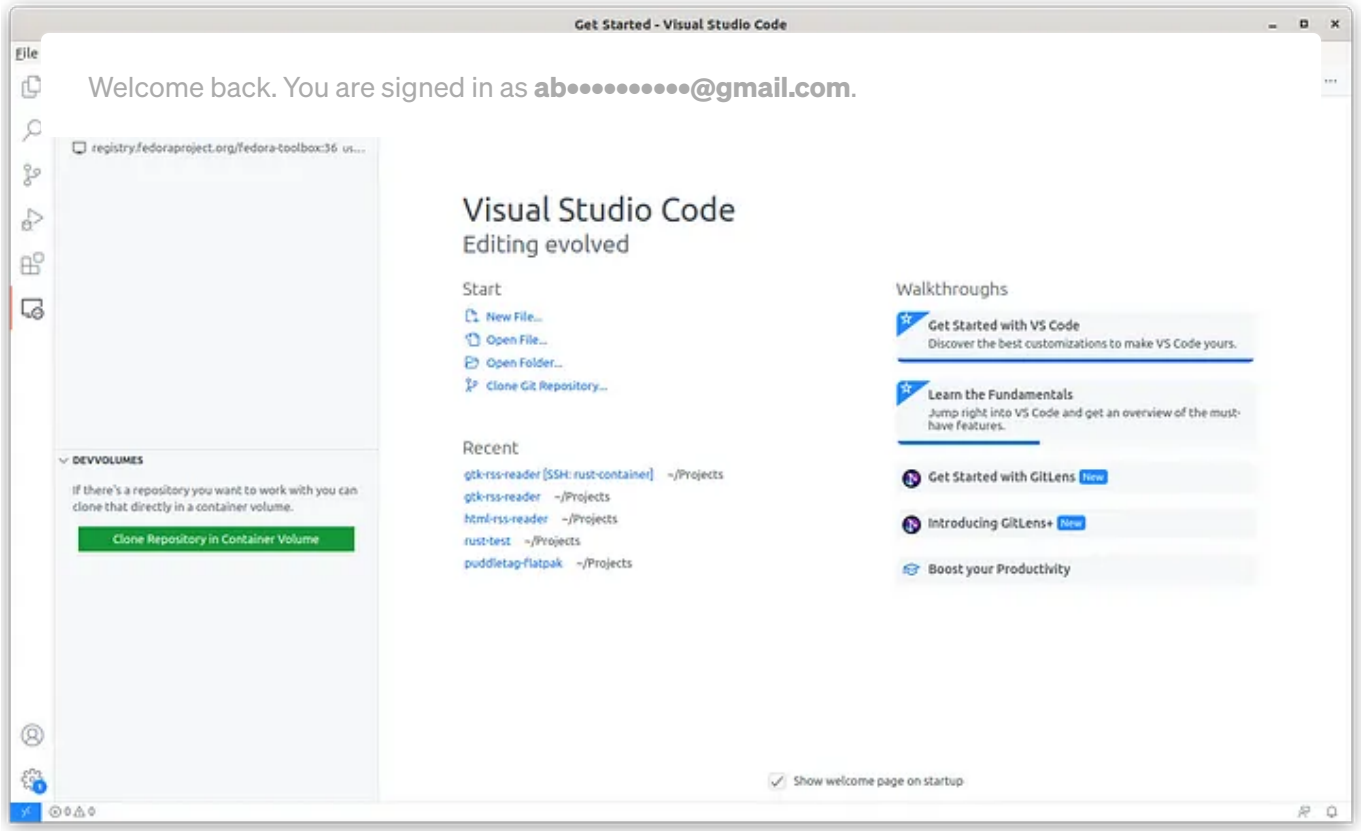


Publishing podman to the VSCode flatpak

Basically we add `xdg-run/podman` to the Filesystem settings in the Other files section. We enable the `podman` socket:

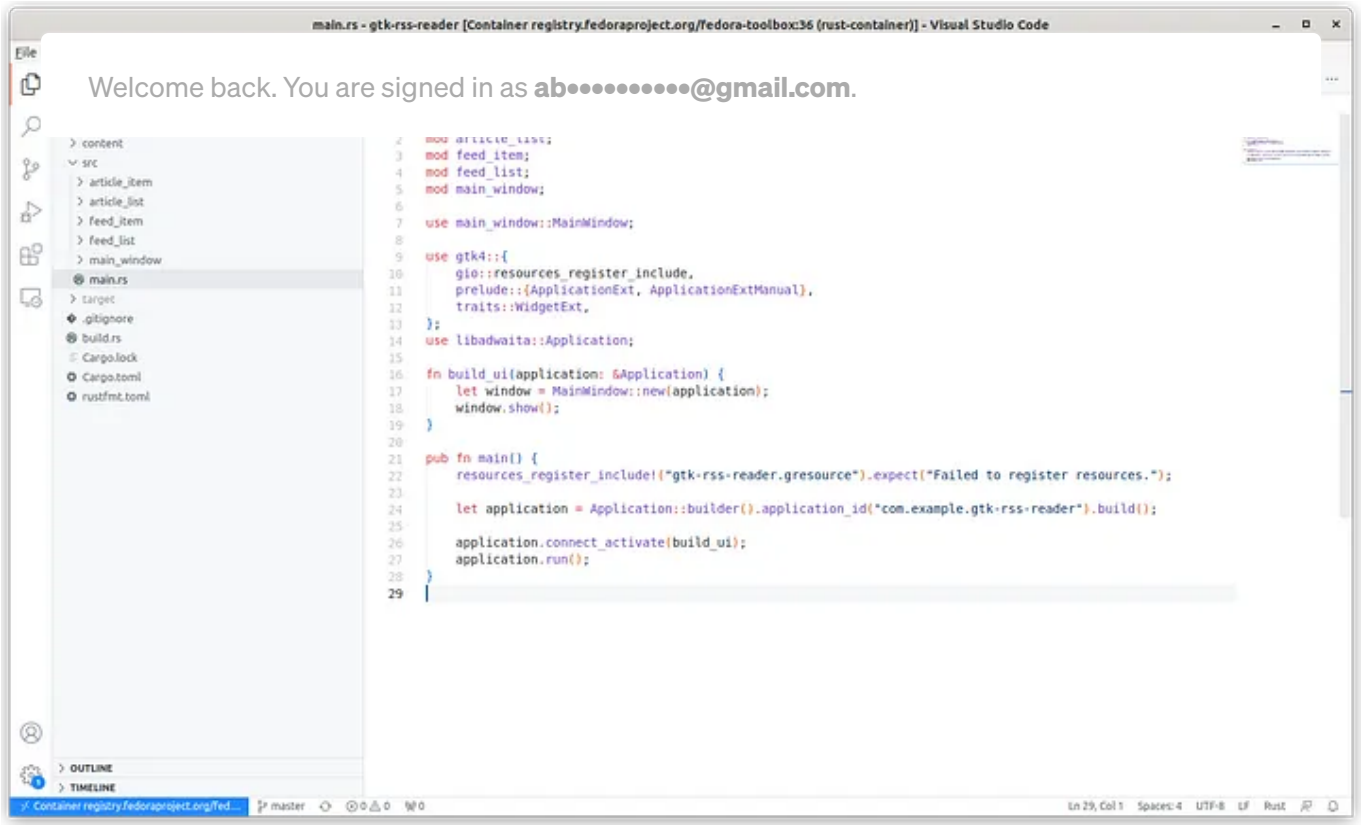
```
#systemctl --user enable podman.socket --now
```

Now if we start the VSCode flatpak and open the Remote Explorer tab from the left sidebar, we will see our container made available:



The Remote Explorer sidebar with our container listed

Let's click on the "Attach to container" icon, prompting us to allow arbitrary code execution. Once attached, we can open a folder from our `home` path: the `home` path is accessible to the container so all git projects there are available by default:



Opening a Rust project in the container

We are ready to write code using the VSCode flatpak. The two ways of connecting to a `toolbox` container presented here are seamless and require little setup. Yes, you need to convince the flatpak app to resolve the `toolbox` executable, but that's business as usual for flatpaks.

Once the configuration is done, you will have a development container at your disposal which is an excellent way to isolate development libraries from the main system. Not to mention if you want to use Fedora Silverblue, where flatpaks are your only option. I hope the article will be of use to you and see you next time!

Written by Radu Zaharia

356 Followers

Still planning that trip to the Moon.

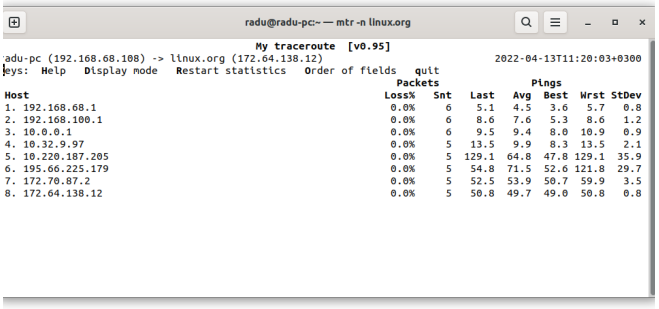
More from Radu Zaharia

Radu Zaharia

When the Linux single-purpose philosophy fails: NFS

There is a long-standing unchallenged axiom in the open-source world: the best...

8 min read · Oct 21

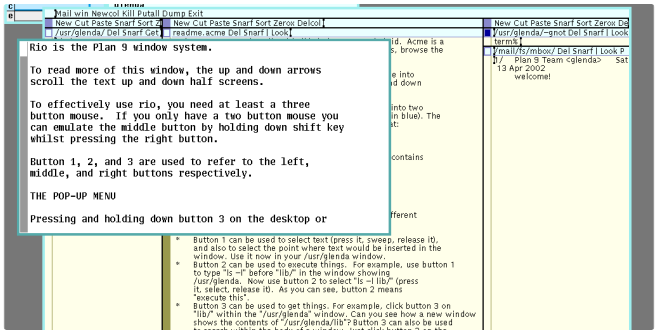


Radu Zaharia

Network health overview with mtr, ss, lsof and iperf3

We said many times in the past that personal and home network security is a habit. It's not...

8 min read · Sep 17



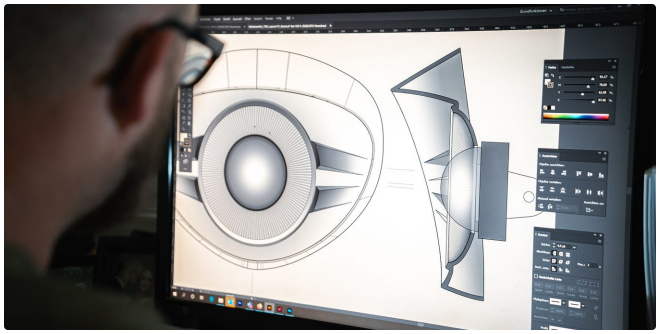
Radu Zaharia

Playing with the Plan 9 operating system

I think I first learned about Plan 9 from Wikipedia. Yes, I am like that. I may have...

8 min read · Oct 5

9



Radu Zaharia

MS-DOS applications: AutoCAD

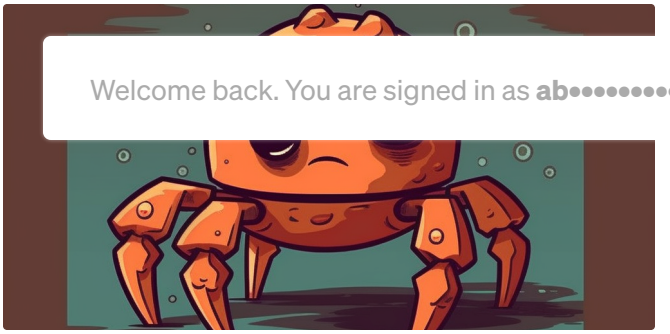
Feeling that I am getting all theoretical and idealistic lately, I thought to bring you...

6 min read · 3 days ago

18 2

See all from Radu Zaharia

Recommended from Medium



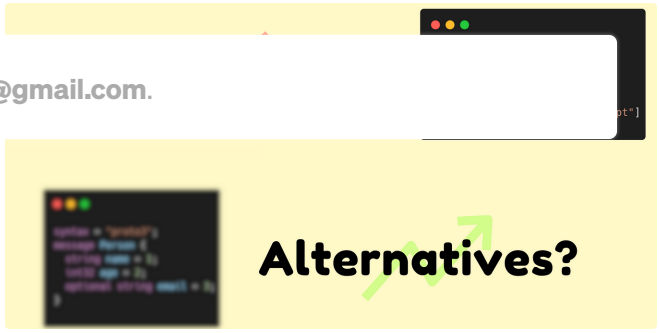
Jarrod Overson

Was Rust Worth It?

From JavaScript to Rust, three years in.

8 min read · Oct 25

1.5K 24



Vaishnav Manoj in DataX Journal

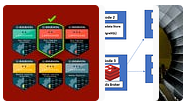
JSON is incredibly slow: Here's What's Faster!

Unlocking the Need for Speed: Optimizing JSON Performance for Lightning-Fast Apps...

16 min read · Sep 28

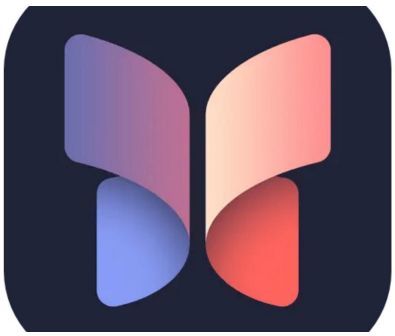
6.9K 87

Lists



New_Reading_List

174 stories · 176 saves



Nikhil Vemu in Mac O'Clock

Tin Plavec

Apple's Journal App Released! — W

Can this replace your existing journal app? Let's find out.

✦ · 6 min read · Oct 29

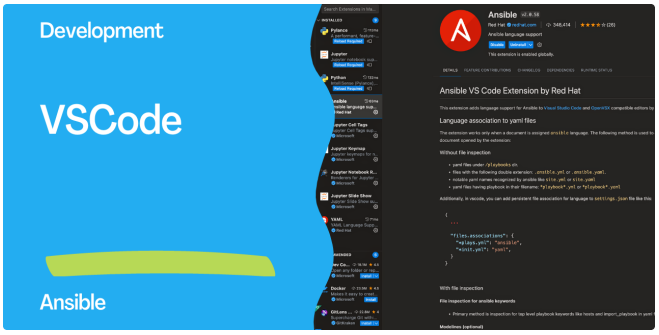
1.4K 25

12 Super Cool Terminal Easter Eggs

Terminal is quite a powerful tool. You will often see people bragging about their knowledge...

✦ · 7 min read · Oct 8

126 5



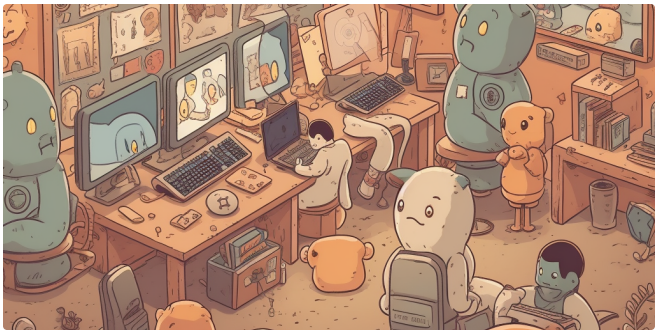
Ansible Pilot in DevOps.dev

Streamline Your Ansible Development with Visual Studio...

Maximize Your Ansible Workflow with the Power of VSCode: Install and Configure the...

✦ · 4 min read · May 10

6



Valerie in Dare To Be Better

4 Alternatives to Postman to Consider After Their Latest Update

The recent changes to Postman made it almost impossible to use. Here are some...

✦ · 3 min read · Oct 4

1.93K 36

See more recommendations