

# Deploying Flutter apps on Linux Desktop

Carlos Nihelton <[carlos.nihelton@canonical.com](mailto:carlos.nihelton@canonical.com)>

Heather Ellsworth <[heather.ellsworth@canonical.com](mailto:heather.ellsworth@canonical.com)>

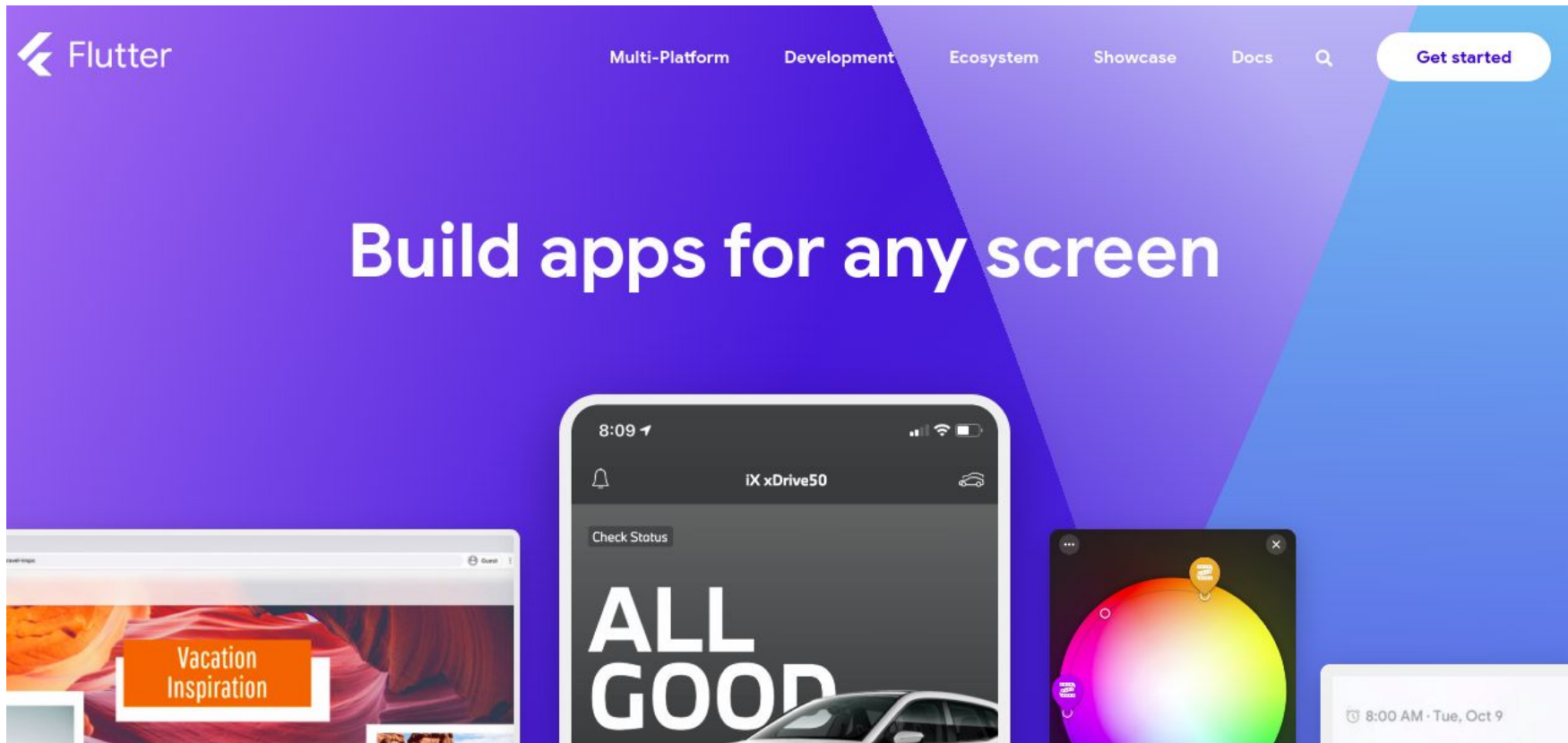
## What you will learn:

- What is Flutter
- What are snaps
- How to snap a Flutter app with a few different examples

Section #1

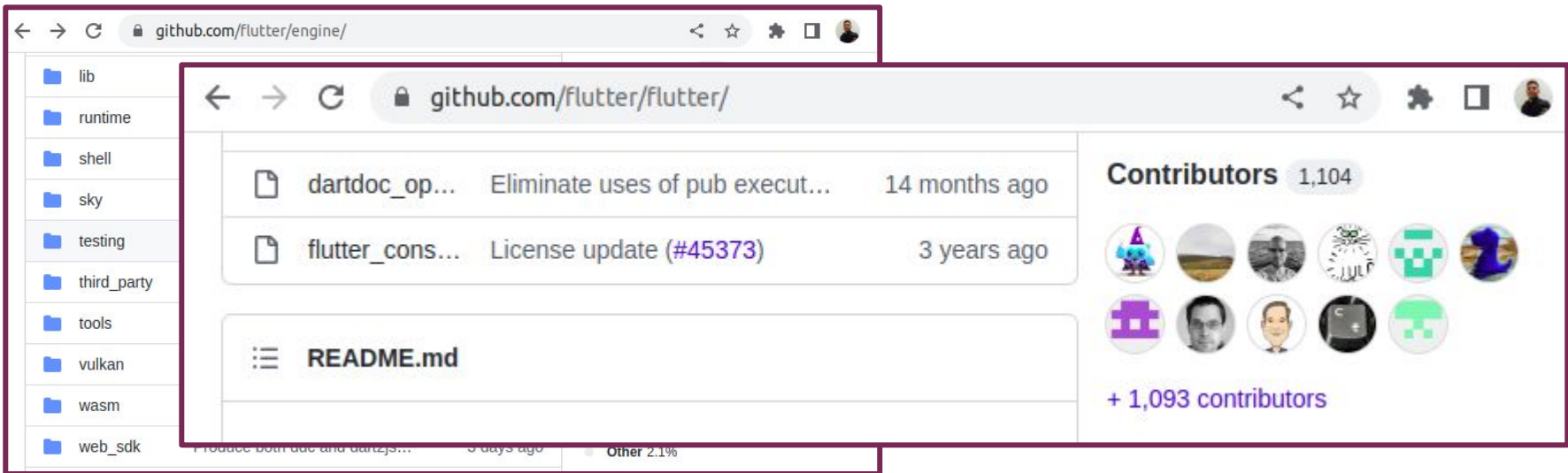
# Foundational Knowledge

# Flutter - What is it?



# Flutter - What is?

It's all about open source



# Flutter - What is?

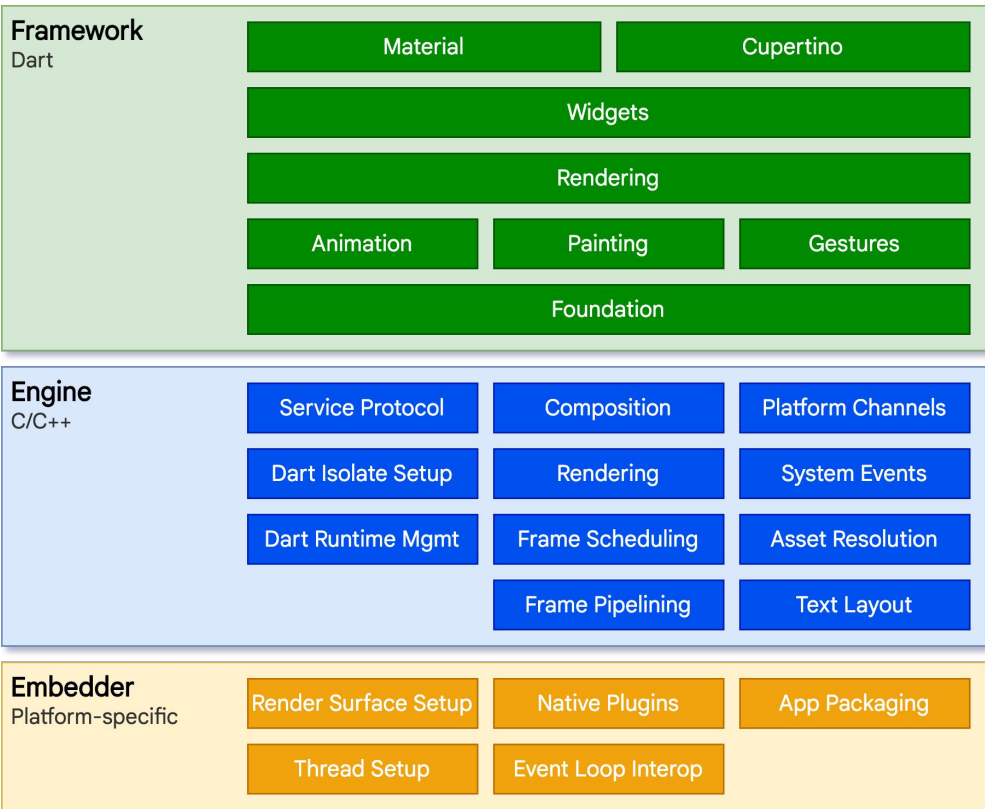
The Ubuntu community has a stake on the Linux support

The image displays two side-by-side screenshots of the GitHub repository for the Flutter engine, specifically the commit history for the Linux platform.

**Left Screenshot:** Shows the commit history for the repository. The top commit is by **dkwingsmt** on Nov 5, 2021, titled "[Linux keyboard] Fix logical keys of up events are not regularized (#...". Below it, a section for "Commits on Oct 29, 2021" features a commit by **robert-ancell** titled "Can't test FView on a server - no display". A user profile card for **robert-ancell** (Robert Ancell) is shown, indicating he is from Napier, New Zealand, and is committed to this repository. A "Follow" button is visible.

**Right Screenshot:** Shows the commit history for the repository, specifically the commit history for the Linux platform. The top commit is by **dkwingsmt** on Jul 21, 2021, titled "Update key mapping to the latest logical key values (#25883)". Below it, a section for "Commits on Jun 25, 2021" features a commit by **jpnurmi** (J-P Nurmi) titled "Linux: add support for TextInputType.none (#26951)". A user profile card for **jpnurmi** is shown, indicating he is from Sweden, committed to this repository, and is a member of Flutter and 8 more. An "Unfollow" button is visible.

# Flutter - What is it?



*Write once in a high level language (Dart)*

*Run natively in the target platform.*

# Flutter - What is it?

## Flutter is all about reactive UI

- Widgets are immutable. State is decoupled from the widget.
  - They are translated at runtime into mutable elements and render objects which are handled by the framework transparently\* from the developer's perspective.
- $UI = f(state)$ 
  - *Widget build methods are functional programming*
  - The framework supports multiparadigm programming, though.



# The State of Flutter on Desktop

One common request has been for Windows support.

Today, we're thrilled to announce the availability of support for Windows for Flutter in stable builds.

## Windows and Flutter



Join TechCrunch+

Login

Search Q

TechCrunch+

Startups

Venture

Security

Crypto

Apps

Events

Advertise

More

## Google's Flutter 3 adds support for macOS and Linux desktop apps

Frederic Lardinois @frederic / 9:50 PM GMT+2 • May 11, 2022

Comment



Image Credits: Google

At its I/O developer conference, Google today announced the launch of [Flutter 3](#), the latest version of its open source, multiplatform UI development framework for building natively compiled applications. It's been about four years since the company first launched a beta of Flutter 1.0. At the time, the team's focus was mostly on helping developers build cross-

# Flutter app deployment

Being a native app means it must have some system library dependencies.

- OpenGL, OpenSSL, GNU Lib C, ...

It also mean that it may have other runtime dependencies, not managed by the distro:

- Installing those system wide may interfere with other Flutter apps.
- Packaging becomes non-trivial.

# How a Flutter app binary looks like?

```
> tree -L 3
```

```
├── data
│   ├── flutter_assets
│   │   ├── AssetManifest.json
│   │   ├── FontManifest.json
│   │   ├── fonts
│   │   ├── NOTICES.Z
│   │   ├── packages
│   │   ├── shaders
│   │   └── version.json
│   └── icudtl.dat
├── lib
│   ├── libapp.so
│   └── libflutter_linux_gtk.so
└── myapp
```

Your app assets  
(images, fonts, translations, etc)

Your Dart code compiled to machine code.

The Flutter engine.

The platform embedder, aka executable.

6 directories, 8 files

```
~ /Workshop/basic-flutter-snap/build/linux/x64/release/bundle
```

# How a Flutter app binary looks like?

```
> tree -L 3
```

```
├── data
│   ├── flutter_assets
│   │   ├── AssetManifest.json
│   │   ├── assets
│   │   ├── FontManifest.json
│   │   ├── fonts
│   │   ├── kernel_blob.bin
│   │   ├── NOTICES.Z
│   │   ├── packages
│   │   ├── shaders
│   │   └── version.json
│   └── icudtl.dat
├── lib
│   ├── libflutter_linux_gtk.so
│   ├── libhandy_flutter.so
│   ├── libhandy_window_plugin.so
│   ├── libscreen_retriever_plugin.so
│   ├── liburl_launcher_linux_plugin.so
│   ├── libwindow_manager_plugin.so
│   └── libyaru_plugin.so
└── ubuntu_desktop_installer
```

7 directories, 14 files

```
~/Dev/UDI/packages/ubuntu_desktop_installer/build/linux/x64/debug/bundle
```

```
pubspec.yaml
1 name: ubuntu_desktop_installer
2 description: Ubuntu Desktop Installer
3 publish_to: 'none'
4 version: 1.0.0+1
5
6 environment:
7   sdk: '>=2.17.0 <3.0.0'
8   flutter: '>=3.0.0'
9
10 dependencies:
11   async: ^2.8.2
12   collection: ^1.15.0
13   dartx: ^1.0.0
14   dbus: ^0.7.3
15   form_field_validator: ^1.1.0
16   gsettings: ^0.2.5
17   intl: ^0.17.0
18   meta: ^1.7.0
19   nm: ^0.5.0
20   path: ^1.8.0
21   provider: ^6.0.0
22   safe_change_notifier: ^0.2.0
23   scroll_to_index: ^3.0.0
24   yaru: ^0.4.1
25   yaru_icons: ^0.2.1
26   yaru_widgets: ^1.0.2
27
28 dev_dependencies:
```

# Snap - What is it?

- Containerized (LXD) applications
- Safer than traditional debs
- Packaged dependencies along with app
- Faster pipeline to user
- Massive snap store: <https://snapcraft.io>
- Written in yaml
- snapcraft build tool, snapd daemon
- Slots & Plugins
- Extensions
- Interfaces

# snapcraft.yaml

Common pieces:

- metadata (name, description, version, etc.)
- apps - what binaries we are building
- parts - what pieces go into the binaries

```
name: hello-world-gtk
version: '0.1'
summary: Gtk Hello World example
description: A simple Gtk example
base: core22
confinement: strict

apps:
  hello-world-gtk:
    command: src/hello-world-gtk
    plugs:
      - x11
      - wayland

parts:
  hello-world-gtk:
    plugin: dump
    source: .
    stage-packages:
      - libgtk-4-1
```

Section #2

# Basic Flutter snap

# WIP Core22 Flutter Plugin

We now have a Core 22 Flutter plugin for snapcraft! (*kinda*)



```
sudo snap refresh snapcraft  
--channel=latest/edge/pr-3952
```

or

```
sudo snapcraft install snapcraft  
--channel=latest/edge/pr-3952
```



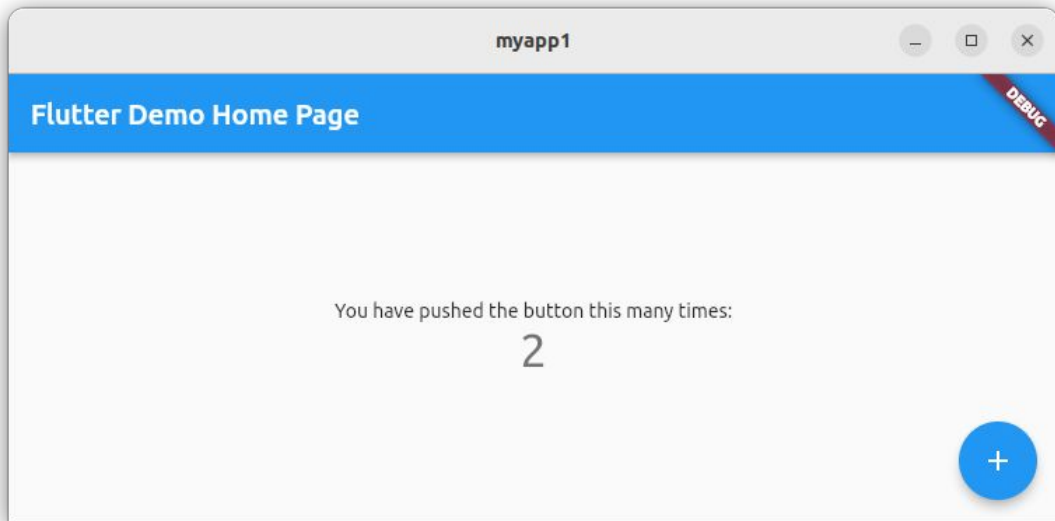
# Create basic flutter app

Simple counter: <https://docs.flutter.dev/get-started/test-drive?tab=terminal>

```
$ flutter create my_app
```

```
$ cd my_app
```

```
$ flutter run
```



# Let's snap it! - metadata section

```
name: flutter-app
grade: stable # must be 'stable' to release into candidate/stable channels
confinement: strict
base: core22
version: '0.1'
summary: 'Barebones Flutter app'
description: 'count things'
```

# Let's snap it! - apps section

apps:

flutter-app:

command: bin/myapp

The executable name must match the app name declared in the app's **pubspec.yaml** file

✱ snapcraft.yaml ✕ | ✱ pubspec.yaml ✕

```
12 command: bin/myapp
13 extensions: [gnome]
14
```

```
1 name: myapp
1 description: A new Flutter project.
2
```

# Let's snap it! - parts section

```
parts:
```

```
  flutter-app:
```

```
    plugin: flutter
```

```
    source: .
```

Section #4

# Snapcraft gotchas

# Common issues you may run into...

Docker - sets ip tables that prevent lxd containers from reaching internet

How to disable:

```
$ iptables -P FORWARD ACCEPT
```

Setup lxd before trying to snap something:

```
$ sudo snap install lxd
```

```
$ lxd init # just hit enter through all of the defaults
```

Section #3

# Intermediate Flutter snap

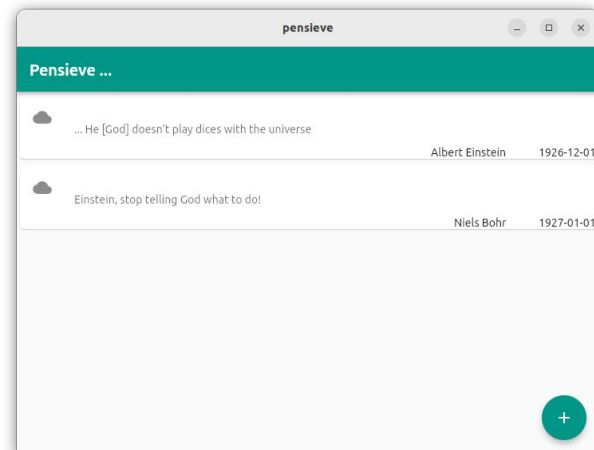
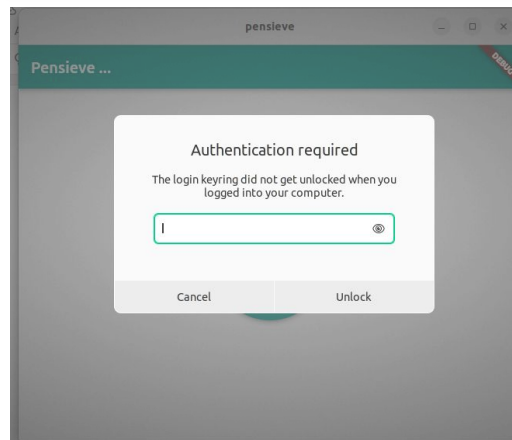
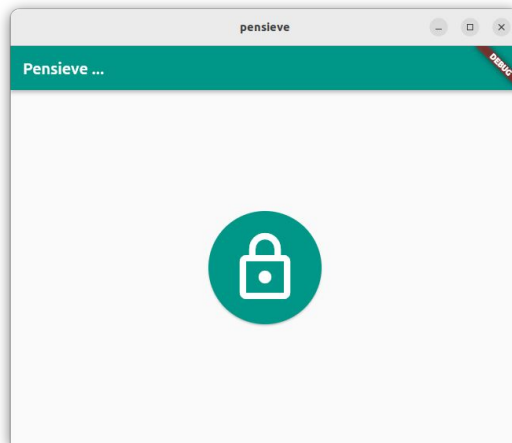
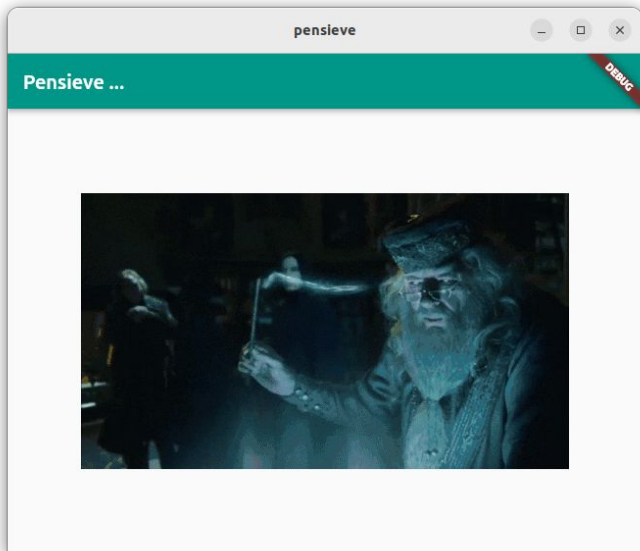
# Intermediate application

## The pensieve

<https://github.com/CarlosNihelton/pensieve>







# Oops

```
| snapcraft-202211_ x
14 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:07.531 :: [ +4 ms] executing: [build/linux/x64/release/] cmake -G Ninja -DCMAKE_BUILD_TYPE=Release -DFLUTTER_TARGET_P
LATFORM=linux-x64 /root/parts/pensieve/build/linux
13 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:07.898 :: [ +367 ms] -- The CXX compiler identification is Clang 14.0.0
12 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:07.913 :: [ +14 ms] -- Detecting CXX compiler ABI info
11 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.055 :: [ +142 ms] -- Detecting CXX compiler ABI info - done
10 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.063 :: [ +8 ms] -- Check for working CXX compiler: /usr/bin/clang++ - skipped
9 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.068 :: [ +4 ms] -- Detecting CXX compile features
8 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.069 :: [ ] -- Detecting CXX compile features - done
7 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.076 :: [ +7 ms] -- Found PkgConfig: /snap/gnome-42-2204-sdk/current/usr/bin/pkg-config (found version "0.29.2")
6 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.077 :: [ ] -- Checking for module 'gtk+-3.0'
5 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.117 :: [ +39 ms] -- Found gtk+-3.0, version 3.24.34
4 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.240 :: [ +123 ms] -- Checking for module 'glib-2.0'
3 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.254 :: [ +14 ms] -- Found glib-2.0, version 2.74.0
2 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.286 :: [ +31 ms] -- Checking for module 'gio-2.0'
1 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.301 :: [ +14 ms] -- Found gio-2.0, version 2.74.0
751 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.350 :: [ +48 ms] -- Checking for module 'libsecret-1>=0.18.4'
1 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.371 :: [ +21 ms] -- Found libsecret-1, version 0.20.5
2 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.427 :: [ +55 ms] -- Checking for module 'jsoncpp'
3 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.433 :: [ +6 ms] -- No package 'jsoncpp' found
4 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.435 :: [ ] CMake Error at /usr/share/cmake-3.22/Modules/FindPkgConfig.cmake:603 (message):
5 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.435 :: [ +1 ms] A required package was not found
6 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.435 :: [ ] Call Stack (most recent call first):
7 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.435 :: [ ] /usr/share/cmake-3.22/Modules/FindPkgConfig.cmake:825 (_pkg_check_modules_internal)
8 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.436 :: [ ] flutter/ephemeral/.plugin_symlinks/flutter_secure_storage_linux/linux/CMakeLists.txt:14 (pkg_ch
ck_modules)
9 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.436 :: [ ] -- Configuring incomplete, errors occurred!
10 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.436 :: [ ] See also "/root/parts/pensieve/build/build/linux/x64/release/CMakeFiles/CMakeOutput.log".
11 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.445 :: [ +9 ms] Building Linux application... (completed in 918ms)
12 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.446 :: [ ] "flutter linux" took 28,801ms.
13 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.451 :: [ +5 ms] Unable to generate build files
14 2022-11-06 09:38:10.808 :: 2022-11-06 08:38:08.452 :: [ ]
```

# Let's add the required build package

```
> apt search jsoncpp
Sorting... Done
Full Text Search... Done
libjsoncpp-dev/kinetic,now 1.9.5-4 amd64 [installed]
  library for reading and writing JSON for C++ (devel files)

libjsoncpp-doc/kinetic 1.9.5-4 all
  API documentation for libjsoncpp-dev

libjsoncpp25/kinetic,now 1.9.5-4 amd64 [installed,automatic]
  library for reading and writing JSON for C++
```

```
~ /Workshop/pensieve git ↻ snap-p2 !1
```

```
✱ snapcraft.yaml ✕
1 name: pensieve
1 grade: stable
2 confinement: strict
3 base: core22
4 version: '0.1'
5 summary: 'A secure place to dump your secret ideas and citations.'
6 description: 'Pensieve allows you to dump your own thoughts or phrases ot
7   you in some sense but you do not want it readily available for others to
8
9 apps:
10   pensieve:
11     command: bin/pensieve
12     extensions: [gnome]
13
14 parts:
15   pensieve:
16     plugin: flutter
17     source-type: git
18     source: https://github.com/CarlosNihelton/pensieve.git
+ 19     build-packages:
+ 20       - libjsoncpp-dev
21
```

```
> pensieve
/snap/pensieve/x1/bin/pensieve: error while loading shared libraries: libjsoncpp.so.2
5: cannot open shared object file: No such file or directory
```

Section #4

# Advanced Flutter snap

# Advanced application

## GitJournal

<https://github.com/GitJournal/GitJournal>



GitJournal

Mobile first Markdown Notes integrated with Git



 PASSED

License AGPL v3

 REUSE status

Sponsor 

 Donate via Bitcoin

 Donate via Bitcoin

# Let's modernize this snap

There is a core18 based snapcraft.yaml in that repository to build a classic snap for this app.

Let's modernize it:

- core22
- Strict confinement



# Accessing \$HOME

Journal files are written into ~/Documents

Snaps are by design not allowed to access personal files inside \$HOME.

Snaps can have their own \$HOME directory, which is the safe choice for most cases.

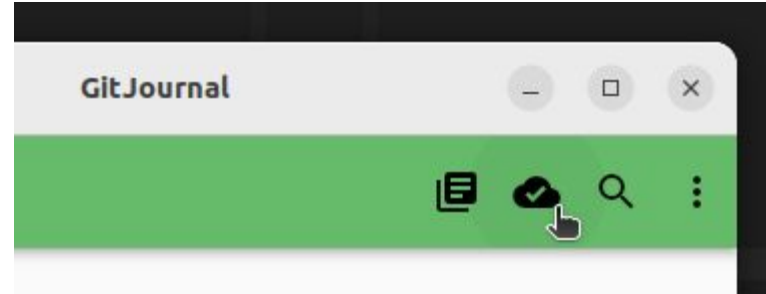
This app looks for ~/Documents by using the Flutter path\_provider package, which, for Linux, relies on xdg-user-dirs.

```
getDocumentsPath() => ~/Documents
```

# Accessing network and inquiring network-manager

The app wants to know if there is network available in order to fetch and pull data from the Git provider.

Inquiring network-manager through DBus is possible with the network-manager interface.





# System file pickers

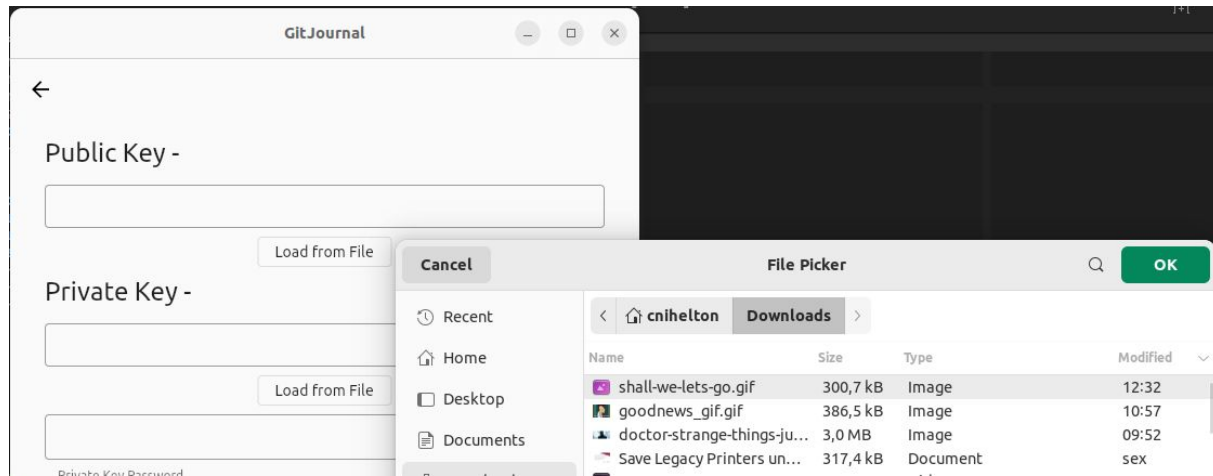
## Zenity

Zenity is not staged as part of gnome extension.

- Reasons for doing so include the fact that snaps are not supposed to load read/write user or system files, by default.

The app allows loading custom SSH keys, which require opening a file picker.

Stage zenity package



# SSH can be quite tricky

The ssh client inside the snap needs to read the user's "`~/.ssh/known_hosts`" file. Failing to read the file is treated the same as if it was empty.

Test (from outside the snap env) if GitHub is in your SSH local hosts. If not, the SSH client will prompt you to verify and accept the fingerprint.

```
> ssh -T git@github.com
git@github.com: Permission denied (publickey).
> ssh -T git@gitlab.com
The authenticity of host 'gitlab.com (172.65.251.78)' can't be established.
ED25519 key fingerprint is SHA256:eUXGGm1YGsMAS7vkcx6JOJd0GHPem5gQp4taiCfCLB8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'gitlab.com' (ED25519) to the list of known hosts.
git@gitlab.com: Permission denied (publickey).
```



Thank you. Questions?

