Flatpak Sandboxed Linux apps

START HERE

`flatpak search <text>`
Find an app

`flatpak install <remote> <app-id>`
Install an app

`flatpak run <app-id>`
Start an app

`flatpak list`
What is installed

`flatpak update`
Update everything

`flatpak uninstall <app-id>`
Remove an app

`flatpak uninstall --unused`
Remove unused runtimes

`flatpak info <app-id>`
Details of an app

Remote e.g. flathub → `install` → **Installation** --user or --system → `run` → **Sandbox** app + runtime → `override` → **Permissions** per app or global

CONCEPTS

Apps run in a sandbox on a shared runtime. Installing one app may pull in a runtime that other apps reuse.

A ref is name/arch/branch, e.g. org.mozilla.firefox/x86_64/stable. The short app ID is usually enough.

Install without --user or --system defaults to system-wide and may ask for admin rights. Use --user for a per-user install.

REMOTES

`flatpak remotes`
List configured remotes

`flatpak remote-add --if-not-exists <name> <url-or-.flatpakrepo>`
Add a remote

`flatpak remote-ls <remote>`
List what a remote offers

`flatpak remote-delete <remote>`
Remove a remote

INSTALL AND UPDATE

`flatpak search <text>`
Search remotes for apps

`flatpak install <remote> <app-id>`
Install from a remote

INSTALL AND UPDATE (cont.)

`flatpak install --user <app-id>`
Install for the current user only

`flatpak install -y <app-id>`
Install without prompts

`flatpak update`
Update apps and runtimes

`flatpak update <app-id>`
Update one app

`flatpak update --commit=<commit> <app-id>`
Move to a specific commit
Find commits with remote-info --log; mask the ref afterwards or the next update moves it again.

`flatpak uninstall <app-id>`
Remove an app

`flatpak uninstall --unused`
Remove unused runtimes

`flatpak uninstall --delete-data <app-id>`
Remove app and its data

`flatpak mask <pattern>`
Block updates for matches

`flatpak pin <runtime>`
Keep a runtime from auto-removal

RUNNING

`flatpak run <app-id>`
Start an app

`flatpak run --command=<cmd> <app-id>`
Run another command in the sandbox

RUNNING (cont.)

`flatpak run --filesystem=<path> <app-id>`
Grant a path for this run

`flatpak ps`
List running instances

`flatpak kill <app-id>`
Stop a running app

INSPECT

`flatpak list`
List installed apps and runtimes

`flatpak list --app`
List only apps

`flatpak list --runtime`
List only runtimes

`flatpak info <app-id>`
Show an installed app

`flatpak info -m <app-id>`
Show its metadata and permissions

`flatpak history`
Show install and update history

PERMISSIONS

`flatpak override --user --filesystem=<path> <app-id>`
Give an app a path

`flatpak override --user --socket=<socket> <app-id>`
Allow a socket, e.g. wayland

`flatpak override --user --nosocket=<socket> <app-id>`
Deny a socket

PERMISSIONS (cont.)

`flatpak override --user --env=<VAR>=<value> <app-id>`
Set an env var

`flatpak override --user --show <app-id>`
Show your overrides

`flatpak override --user --reset <app-id>`
Clear your overrides

MAINTENANCE

`flatpak repair --user`
Check and fix the user install

RECIPES

Set up Flathub and install

```
flatpak remote-add --if-not-exists flathub \
  https://dl.flathub.org/repo/flathub.flatpakrepo
flatpak install flathub org.mozilla.firefox
flatpak run org.mozilla.firefox
```

Tidy up

```
flatpak update
flatpak uninstall --unused
flatpak repair --user
```

Let one app see a folder

```
flatpak override --user --filesystem=$HOME/Projects <app-id>
flatpak override --user --show <app-id>
flatpak override --user --reset <app-id>
```

Hang app? Stop it

```
flatpak ps
flatpak kill <app-id>
```