

Who am I

Ionut Stoica - Full Stack Software Developer

Romanian, living in Louvain-la-Neuve, Belgium GMT+1

stoica.ionut@gmail.com

<https://github.com/iongion>

<https://twitter.com/iongion>

<https://www.linkedin.com/in/iongion/>



Podman Desktop Companion

A familiar desktop graphical interface for the free and open container manager - **Podman!**

The familiar technology is so ubiquitous that exposes developers to tech stack drop risks, have been there, **never again!**

Target

- Those wanting to learn about containers (podman)
 - Containers / Instances / Machines / Volumes / Secrets
 - Kubernetes / Pods and all the overwhelming terminology
- Full stack developers
 - They juggle with different platforms all day
 - They don't need as much CLI interference as power users / devops
 - Switching from OS to OS has its own inconveniences, let this be the last!

Technical goals

1. Look and feel the same on all operating systems

But do integrate with the native UI as much as it is healthy (notification systems / alerts / logging / security / permissions)

2. Reach parity with **The familiar UI**

- a. Provide detailed stats
- b. Support Kubernetes / Pods
- c. Support developer environment sharing

3. Support Windows and MacOS

- a. Windows
 - i. **Problem 1** - communicate with the REST api in WSL from Windows desktop app
 - ii. **Problem 2** - support native volume mounts (Something specific to /mnt/ on WSL)
- b. MacOS
 - i. **Problem 1** - communicate with the REST api in LIMA from MacOS desktop app
 - ii. **Problem 2** - support native volume mounts (LIMA supports it already)

Architecture - trials

Native

Why

- Integrates well with platform UI/UX idioms
- Easy to distribute
- Fast & small ?
- Tech stacks - C / C++ / Swift / Pascal

Why not

- No similar looks - against technical **Goal 1**
- Native apps are hard to develop and maintain, full native means different stacks
- Small, not really, depends on OS
- Ecosystem maturity of the baseline tech stack
- Hard to choose a tech stack - not enough knowledge

Managed - why / easy | why not / big & slow ... but is it really that way ?

- Python - PySide / GTK for Python
- Javascript / Typescript / React / NodeJS
 - React Native
 - Web as desktop app with Electron

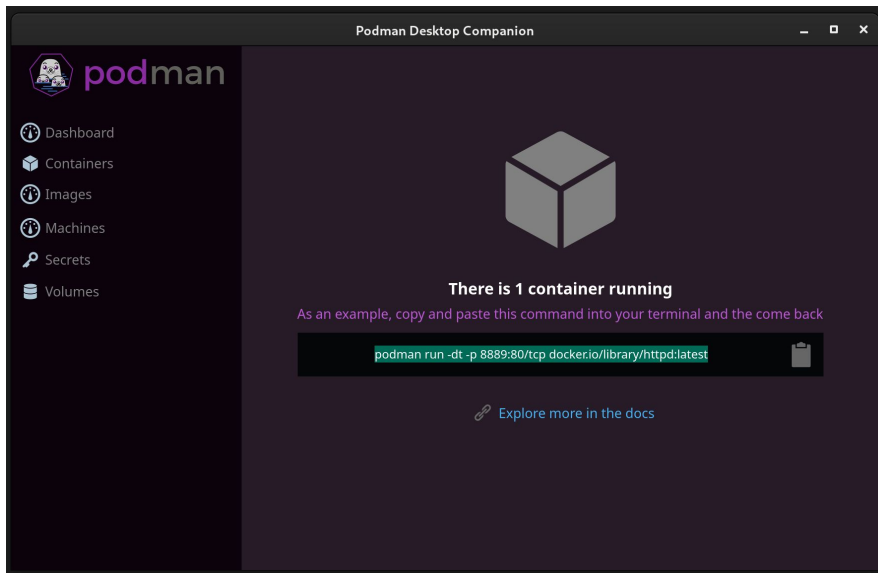
Native Trial using Lazarus

Pros

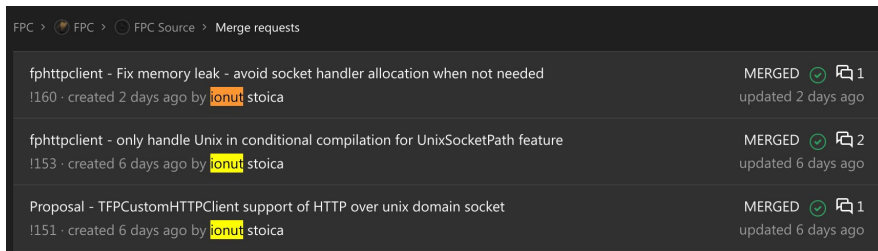
- “Write once, compile anywhere” using Pascal language from childhood
- Similar looks on all platforms - **Pro Goal 1**
- Fast / native speed
- Easy to bundle distribute, small size with all batteries included
- Very friendly community and maintainers

Cons

- Using Pascal from Childhood
- Not easy to have collaborators - not so many people know Pascal these days :(
- Not so easy to develop with modern FE philosophy - imperative UI style
- Not so similar looks if not implementing custom UI
- No WebView support - could be useful for dashboard / metrics



Added a patch to fphttpclient to support unix domain socket connections - well received and merged



Other trials

GTK 4 - Looks awesome, with Vala / PyGTK, even more awesome

- Hard to bundle
- Hard to be cross operating system
- Not enough knowledge - don't know how to create custom UI from scratch
- Hard to work with processes and threads - GIO - launching podman as child process is so difficult
- Hard to update

QT - Looks awesome, with PySide, even more awesome

- Hard to bundle
- Not enough knowledge - don't know how to create custom UI from scratch
- Hard to work with processes and threads
- Hard to update

Decision

Electron Web APP - Just deal with it

Why

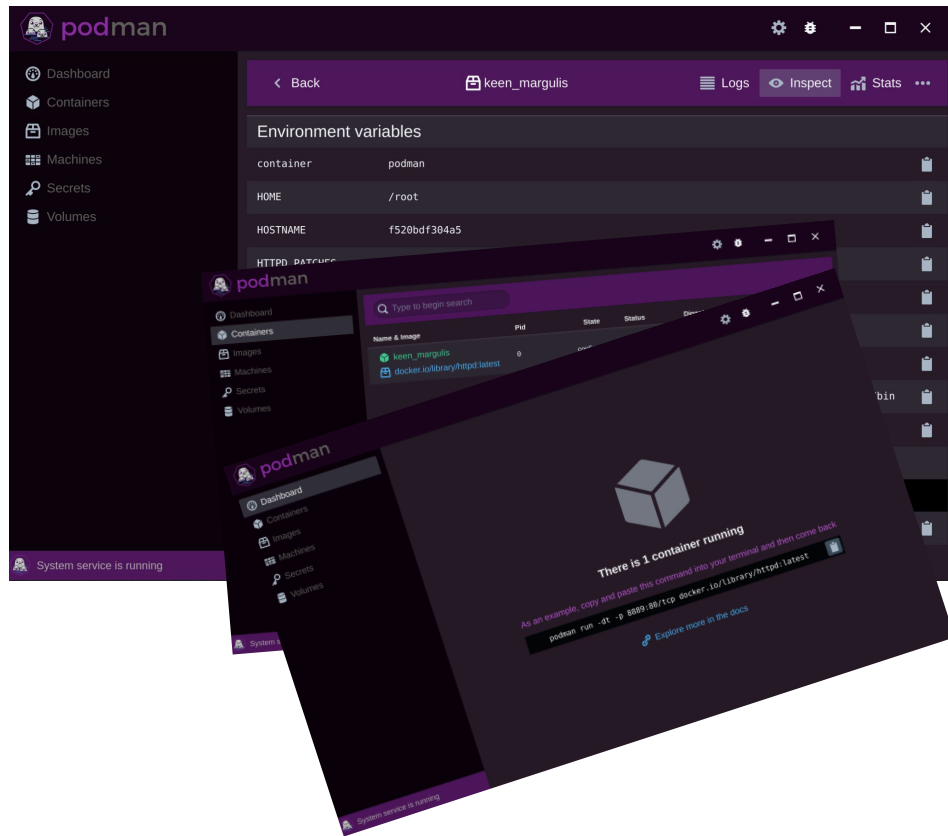
- **Pro Goal 1**
- Easy to develop / share ownership / review
- Easy to bundle for all major OSs
- **Can create any UI component imagined in 0 time**
- React / Typescript / BlueprintJS ecosystem maturity
- Functional way of doing UI
- Community is huge and has answers for everything
- Any modern frontend developer can participate

Why not

- Big bundle size - 100 MB - for a good reason, flatpak / snap of the day is similar in size
- People just hate Electron - for no good reason most of the time

Still exploring

- Electron lightweight alternatives - DeskGap / Tauri
- Explore bundler alternatives - esbuild / swc / vite stacks
- React native GTK / QT / Win32 / Cocoa



Webapp bundled as native app

<https://iongion.github.io/podman-desktop-companion/>

Immediate

Create Windows and Mac binaries ASAP - expose the volume definition UI

- <https://github.com/iongion/podman-desktop-companion/tree/support-windows>

Open github issues

Make more noise - show off / write articles, take the 10 most popular familiar articles and convert them to podman & podman desktop companion

Soon

- **podman-compose** integration - respect and group containers
- Follow podman openapi spec closely, maybe use generated clients instead of implementing from scratch
- Add testing and automation for multiple podman versions
- Pods support
- Remote client support
- Tighter integration with podman team

Thanks

- CRI-O slack channel
- **Anders Björklund** - awesome and invaluable guidance during early prototypes, lots of annoyance towards him :) - <https://github.com/afbjorklund> - PRs / Comments / Requests / Fixes / Clarifications related to the API
- **Matthew Heon** - team introduction, kind words, support
- **Brent Baude and Ashley Cui** - all encouragement

Tom Sweeney - care for all the initiative