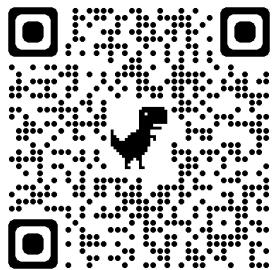


## Orb : A Lightweight Urbit File Serving Engine

See an example of Orb serving at <http://142.93.135.202:9000/>  
Scan QR code to access go-live from your phone browser



Orb is an Urbit frontend (UI, Configuration, and Process Management) on top of a tool I wrote called go-live. Go-live is a dead-simple extremely high performance file-serving CLI tool that has:

- \* Extremely high reliability and performance — can serve a vanilla Unix file directory publicly over the internet to clients.
- \* The files hosted support nesting, and from the browser's perspective, mimic a local directory.
- \* The files can be updated in real time and rely on the browser of the client to display it.
- \* The files that can be hosted can be photos (JPG, PNG) data files (.PDF, .DOC) movie files (.MP4, .MKV) and even webpages (.HTML, .JS, .CSS) which can render web pages similar to Github Pages
- \* Compiles into a lightweight binary that runs on all Unix machines because it's written in lightweight Go

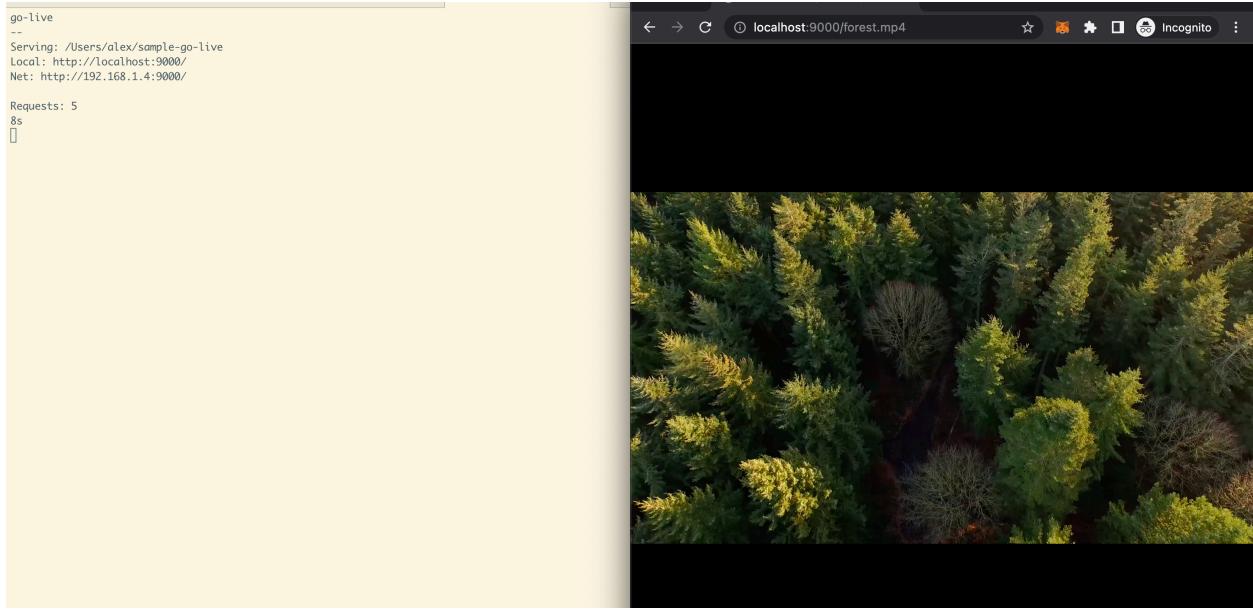
Some of the tools its similar to are Caddy, or Github Pages. Orb supports ARM64 / X64 / X32 on Linux Mac and Windows. The entire binary is 4 MB compiled.

### Where does Orb fit in:

Orb is an urbit-ified version of go-live with good defaults, and a nice UI for hosting a public file server with whatever you want on it from your Urbit ship. Ideally user spins up Urbit, gets there planet going and then spins up Orb.

1. You open Orb app from the launchpad.
2. You select or create a file directory to hold the files you're going to serve.
3. You can add or remove files as you want to the folder, these can be new files or modified files.
4. End consumers can access your app via the public IP address the Urbit server uses on a port you specify. This can have nginx or cloudflare in front of it.
5. Urbit maintains the go-live process which is a Unix executable file written in Go. This can run as a moon.

**Figure 1: go-live in action go to <https://github.com/antsankov/go-live> to see an actual gif demo and source code.**



**Figure 2: Go-live process running for 718 Days (2+ years), with minimal memory leaks, after serving over 1.3 million web requests (an entire website which was migrated off of Github Pages)**

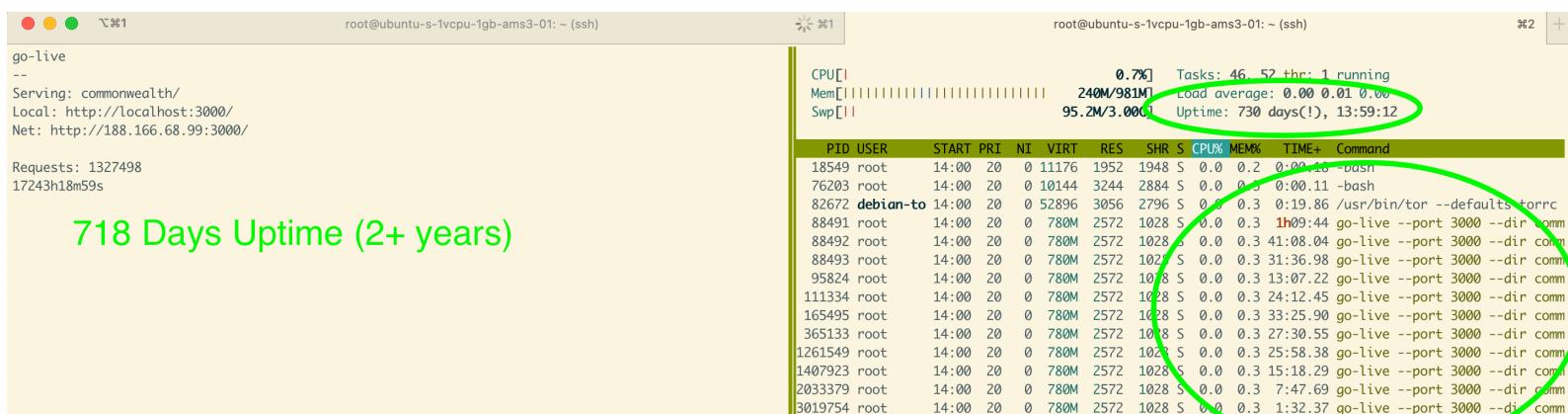


Figure 3: Orb mockup on the home screen ( this would have to be written in Hoon)

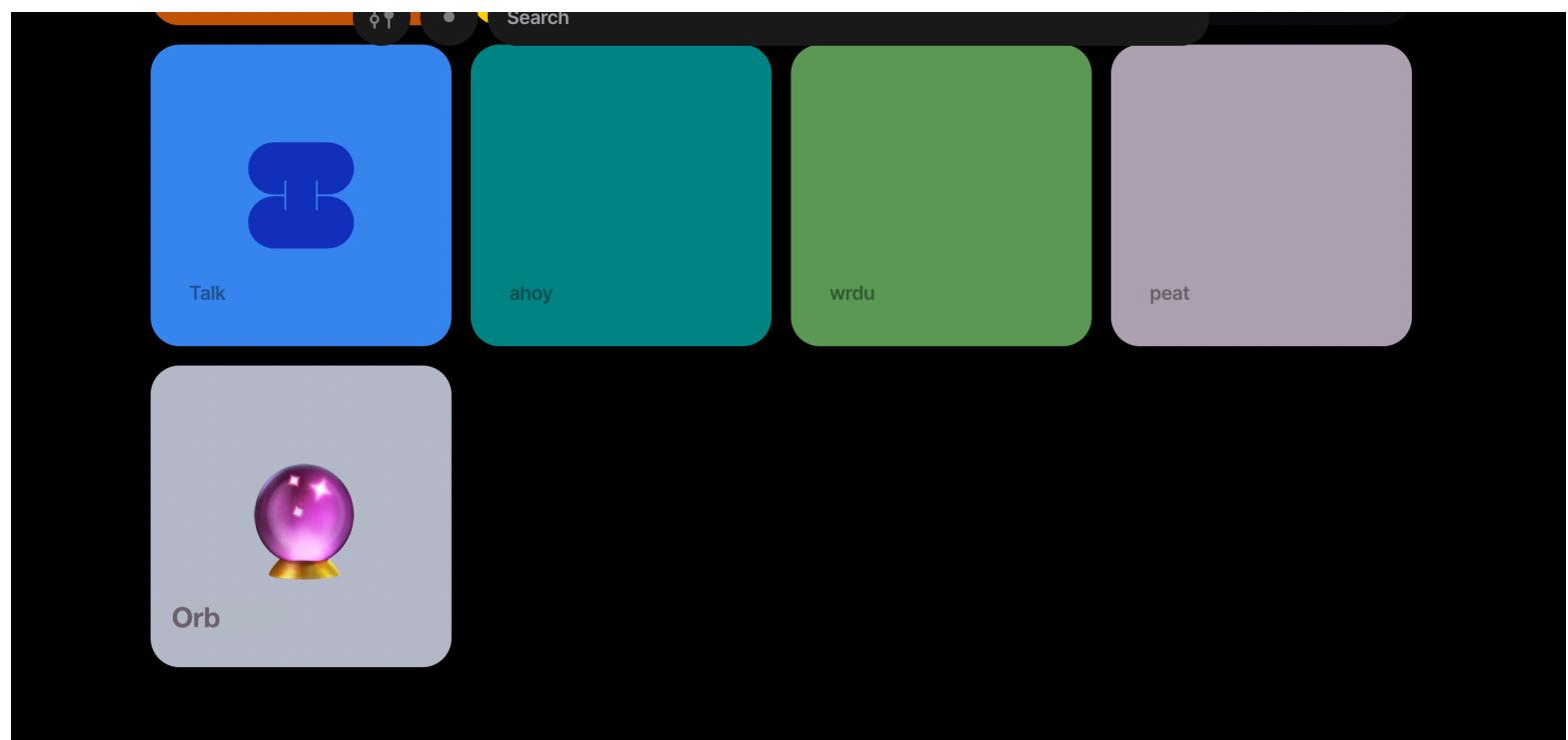


Figure 4 How Orb would appear once the app is open



**Assumptions:**

- Hoon can invoke a unix program (should be an Urbit jet) via the Unix CLI.
- Hoon / Urbit can manipulate the file system (this might be necessary to create the initial Orb directory if it does not exist, but not necessary— could be built into go-live)
- For a user to get real value, it would be best to host Urb on a public server. This light weight utility can support:

**How would users use it:**

- \* Artists or every day people who would like to just add photos or media to a directory
- \* You can host a webpage with full Javascript and CSS in there, exactly the same way as Github Pages with full git manipulation of the directory (Ex: git pull master while the Orb server is running)

**Skills needed:**

Hoon frontend experience, the ability to create a Hoon program that can take in strings via the UI and invoke a Unix application with those strings. The app would be shutdown and restarted every time an option is changed

**EXTRAS:**

How to create an Urbit Jet: <https://developers.urbit.org/reference/runtime/jetting>

Github for go-live: <https://github.com/antsankov/go-live>