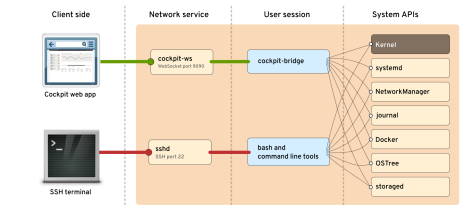
Monty Python's Flying Cockpit

Cockpit intro

- For this talk we assume a basic familiarity with Cockpit
- If you have never seen it, the short-short version.
- Conceptually: Linux session running in a web browser; moral server equivalent of what GNOME is on a desktop
- Tool for experimenting, learning, troubleshooting, and doing infrequent tasks

Monty Python's Flying Cockpit

Architecture



- To understand this talk, you need to know a bit about how cockpit works internally
- Consider what happens with normal SSH session: You want to do stuff on a remote OS which requires running commands, opening files, perhaps talking to a TCP port, and so on. But all that SSH gives you is a text stdin and stdout, i.e. a pipe pair.
- What you need to connect these is something that translates between that pipe and the executables, sockets, D-Bus interfaces of the OS. That is a shell like "bash" for an interactive SSH session.
- Cockpit is a web UI written in JavaScript, but it's the same situation: The browser possibly runs on the other side of the planet, and it can only talk "websocket", which is essentially a pipe. For cockpit, the translator is the bridge. It translates these OS interfaces to a multiplexed JSON stream.

Demo: Fedora Server

- Connect to fedsrv with stable client, explain Client flatpak and cockpit UI
- Show `rpm -qa cockpit` in Terminal, installed by default
- Put myself in the position of the cockpit web UI; connect via SSH to fedsrv and run the bridge
- `cockpit-bridge --interact=---`, with bridge-cli.txt
- We can run a program with arguments, for example ping; we get the chunks of output, and eventually an exit code
- bridge has many channel types, for the file system, sockets, D-Bus, inotify, or metrics (second demo)
- roughly machine readable version of bash

Demo: Connect to a CentOS 9 Stream cloud instance

- Connect to "fresh cloud instance" `c9s`, get "no bridge found", sob

Monty Python's Flying Cockpit

Making the bridge portable

- What can we do here? We surely must have the bridge pre-installed somehow. We wrote it in C to be performant and be able to talk to low-level system interfaces.
- Lis: We could rewrite the bridge in Python! – Whaat? No, that can never work. It's too slow, and the C bridge is thousands of lines, it'll be too hard. And how would we even get that to the remote machine?
- And besides, what has the Python empire ever done for us? ubiquitous, portable, performant with asyncio, bindable with ctypes, much easier/faster to develop

Monty Python's Flying Cockpit

Goals

Goals

- Be like Ansible: SSH + Python
- Cloud instances, production machines, other distributions
- "Inverse" web app

- Lis convinced me; model this after Ansible and reduce assumptions to the minimal: Python with only included batteries, and SSH connection to managed machine
- Get a foot into the door of pretty much any machine out there
- Still need that feat of getting the bridge to the remote machine; normally a server sends a web app to the browser, but here we need to send the backend code to the server machine, sort of an inverse web app
- Lis has some great technology to pull this off
- time check: 7'30 mins

Monty Python's Flying Cockpit

Demo: Portable bridge

- Remember that c9s machine? Let's start our beta version of cockpit's flatpak with the Python bridge
- Connect to c9s, watch jaws drop
- More pages available, like podman; show magic bridge in ps ux

Rollout plan

Rollout plan

- now: Fedora Rawhide, Debian unstable, Flatpak beta
- soon: Fedora 38, C9S/RHEL 9 devel
- never: Debian stable, Ubuntu 22.04 LTS, RHEL 8

- We've been developing this in the main branch of cockpit with a configure option
- Fixed last critical regression last week, but still a few unstable tests
- Launched to Fedora rawhide, Debian unstable three days ago
- Soon enough Fedora 38 and C9s/RHEL 9 devel
- Don't switch long-term support releases, such as Debian stable, Ubuntu LTS, RHEL 8

Contact:

- `#cockpit` on Fedora Matrix
- https://cockpit-project.org

- Home page leads to mailing lists, chat, documentation
- thanks for your attention; Q+A