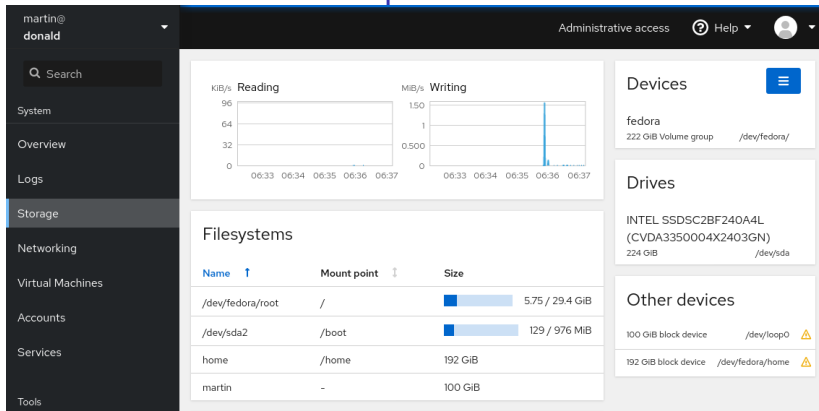# Cockpit Infrastructure

Martin Pitt, Sanne Raymaekers

Cockpit Team

- Interactive Server admin web interface
- Included in all major distros, uses over 100 OS APIs
- 7 team members
- Automated tests, releases, npm/translation updates, VM/container image refreshes

Our Automation Principles

Containerize everything → simple and safe to run locally

No magic infrastructure → reproducible, cloud portability

Automated deployment → scalable, recoverable, ~~bus factor 1~~

Which infrastructure exactly?

- GitHub workflows for all non-KVM tasks
- CentOS CI: Kubernetes ReplicationController
- bos.e2e: systemd-controlled docker
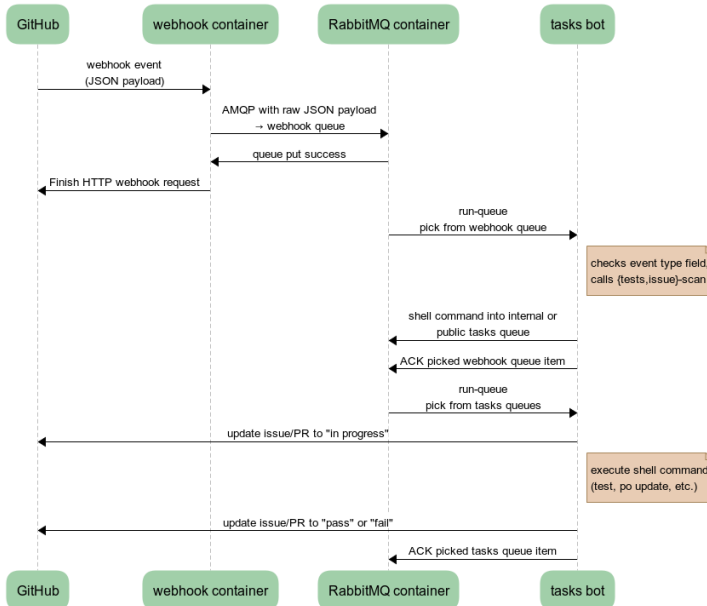- AWS: on-demand c5.metal instance, $$$, systemd podman

Event flow for releases

```
$ git tag -s -m '123

- cool new feature A
- fix heisenberg compensator on Fedora (rhbz#1234)
'
```

→ .github/workflows/release.yml runs release container

→ GitHub release, Fedora dist-git+koji+bodhi, COPR, DockerHub, docs on cockpit home page

Event flow for tests

Strong aspects of our CI

- reproducible, portable
- platform agnostic work queue
- deployment only through Ansible
- fully automated releases
- separate changes in our code from changes in OSes

Weak aspects/challenges of our CI

- arcane test logging and artifacts
- precarious e2e machines
- no monitoring/alerts
- hard to find public infra with /dev/kvm

Links/Documentation

- source.redhat.com/groups/public/cockpit/ cockpit_wiki/cockpit_ci_resources
- github.com/cockpit-project/cockpituous/
- secrets in internal CEE GitLab repo, only accessible to a few team members
- github.com/cockpit-project/bots
- #cockpit on Freenode, cockpit-devel@lists.fedorahosted.org