Obmondo

Building a company on producing Open Source

Klavs Klavsen

Founder of Obmondo

What you’ll hear today

- Something about me
- The idea of Obmondo
  - The view as a consultant
  - The reality of “modern”, open source based operations
- A possible solution
- Building a company
  - Putting a price on open source
  - Financing open source
  - Hiring the right people
  - Remote work?
  - Scaling the company
- Obmondo today

Obmondo

About me – Klavs Klavsen

- **Founder and CEO of Obmondo**
- Been **helping companies build Open Source** based operations for 25+ years
- **Contributed mostly to Open Source** by way of bug reporting, helping online and doing small PRs everywhere I had to fix something
  - Dpkg – small c code fix
  - Kernel SACK bug reporter – officially in kernel as reporter
  - Recompiling apache to go above 256 max children
  - A million other things :)
- Used Linux since 1997, full time desktop user since ~2000
- When I am not working, I try to get time for my wife, my kids, exercise and going to meet and listen to interesting people at meetups

Obmondo

The view as a consultant

- **I Helped build operations envs** for many large orgs, for 20+ years
- **Had to redo every time** as copyright disallows sharing
- **No backports/sharing of features** – reimplement
- **Operations solutions are custom** and diverges over time
- **Everyone is using 80-90% of the same software** to solve the same needs
- **Very expensive** – Not something smaller orgs can afford
- **Everyone is behind** – and I felt frustrated
- **CIS18, regular security scans and recovery tests** – only some large orgs did some
- **Sad to see such an ineffective model** and yet its everywhere
- **I really wanted a more fulfilling day,** and to give others the same

“Modern operations”

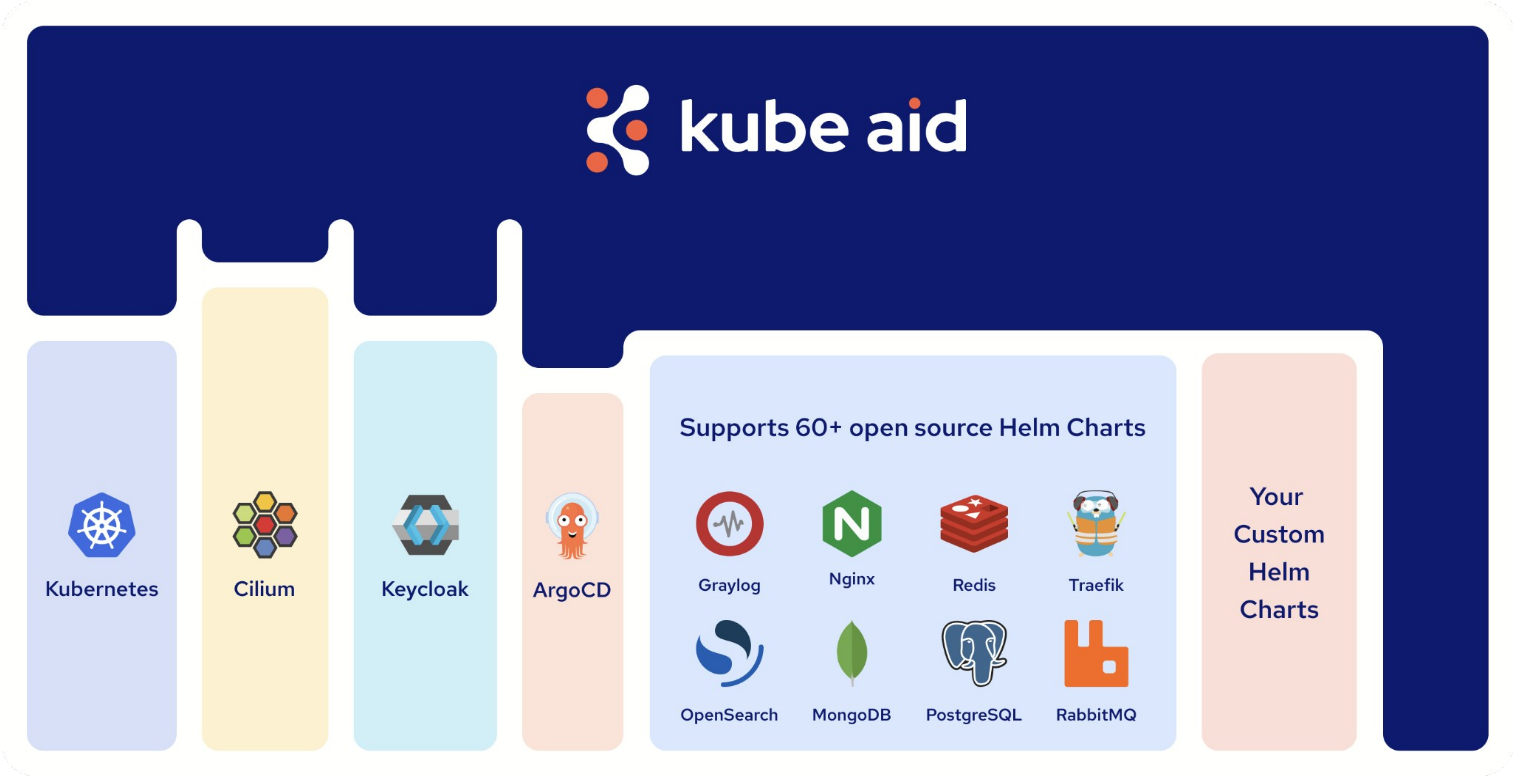
- **Operations is a lot of work**, and req. skills few have
  - Understand security, network and OS/code
  - Understaffed.. - if you are developer doing ops == less time to do ops – you have features to build
- **Problem is managing all these open source components** - ensuring: security updates, secure config, central identity management, supply chain security, backup, recovery tests, network monitoring, network security, storage monitoring etc.
- **Big orgs struggle** to do all this
- **Startups, Individual developers and small orgs** can rarely do all this
- **Collaborating via Open source** does not get done, beyond existing individual projects - everyone builds their own operational setup from zero.

A possible solution?

- Complete Operations setup - as open source
  - GitOps – Git has amazing features
  - Used Puppet, to manage Linux since 2008 (GitOps+ Hiera)
  - ArgoCD – K8s GitOps + hack until v2.6
- Verifying the solution
  - We build Puppet based solution (LinuxAid) first
  - Then ArgoCD based solution (KubeAid)
- Solution worked
  - Git tags allows to pin to tags for specific applications,
  - Hiera (Puppet/LinuxAid) & Argocd (k8s/KubeAid)
    – merging configs (global + customer specific)

A possible solution (concept)

So our KubeAid and LinuxAid projects adds all the stuff you build in ops, adjustments and extensions etc. and the documentation needed.



kube aid

Kubernetes

Cilium

Keycloak

ArgoCD

Supports 60+ open source Helm Charts

Graylog

Nginx

Redis

Traefik

OpenSearch

MongoDB

PostgreSQL

RabbitMQ

Your Custom Helm Charts

A possible solution (example)

- Show https://github.com/obmondo/kubeaid
- Helm umbrella chart – show Firewall + docs (graylog f.ex.)
- Kube-prometheus build and mixins
- Cluster api

Obmondo

Putting a price on open source?

Dual-licensing/OpenCore/UI closed/support?

We choose support & services model, with a little bit of UI/api for large enterprises

Price model – subscription/consulting/incentives?

- **low-cost subscription** – including the "normally expected hours of work" – and extra hours billed at discount rates – after experience of person (level 1/2/3)
- **Varying cost consulting rates**: allows for training and lower cost
  - Level 1: go developer with 1-2 year exp, level 2: go developer with 2+ years, level 3: 7+ years exp
- **Sharing costs:** We have copyright of work - so everything we build, everyone gets – and we automatically cost-share across customers – which actually covers upto 90% of the work we do for customers.
  - We share maintenance costs, so that smaller customers pay % matching their node size. We offer an equal share cost division of features all parties want

Financing open source

- **Investors/self-funded/consulting** – I ended up living off savings and doing consultancy, and slowly building the product, and then getting customers who were willing to get in early on a discounted consulting rate
- **Startup helpers** – you can apply for grants I'm told.. and there's "startup incubators" places
- **Copyright** is important. Be sure that contracts state you (also) have copyright over your work on the open source project and thus may release it as open source.

Hiring the right people

Hire people - Once you start to earn money (or have investment capital). Finding the right applicants is really hard.

- Give an initial "test task of 2-4h work" and Interview with screen share, let them walk you through their code and ask them why they did what they did (or didn't do something)
- Have maybe an extra 5-10 min. improvement task to do live coding – help as needed
- Hiring the more experienced – you'll need to actually do some work with them to ascertain

Hiring the right people (continued) …

Contracts & laws – You'll need an employment contract that follows the law in your applicants country and verify you can hire them directly – Denmark doesn't allow it beyond the first employee (ambassador rule)

- Avoid solutions where they don't work directly for you
- Only hire from democracies without draconian laws (GDPR) – ie. the US is not allowed
- If they invoice you, they're subcontractors and must be listed on GDPR agreement with customers

Keep them happy – You need good people. They only stay if you have a high focus on their happiness. Work with your employees to build a handbook on how you work, teach etc. This gets harder as you scale and will need continuous adjustment.

Remote work?

- **Remote & asynchronous is a great –** but you need to structure your company after 100% remote work – or those remote will lose out and not feel a part of the company
- **Chat communication –** must be primary way of communicating
  - We use self-hosted Mattermost (slack alternative) and signal as fallback
  - Create a culture, with braindump and standup - online
- **Meeting in person** - is very important
  - We get everyone to a new place, every 6 months and have a work-week together
  - Typically we go to a city where a colleague lives.

Scaling the company

- Teams – You'll need to divide people into teams
  - No matter the size, make sure you have at least 2-3 who can lead, plan and review code
  - Figure out how they can share planning – we went with extending gitea with a 4-week planning mode, improving its Kanban to help with label automation and adding global milestones
- Listen - Remember everyone else won't be you
  - Have an open dialogue in your daily communication
  - Schedule team and 1 on 1's talks regularly
  - Work to solve the challenges, with them
- Finances is important – you need to be sure you have margins on income vs. expense – the more employees – the more margins.
  - We went with developing our own tools for internal financial overview

About Obmondo

- **Founded in 2018** – now a team of 30+ people
    - With employees in Denmark, Kenya and India
- **Make high quality operations shareable and affordable**
    - Lowering costs, improving security and hopefully the life of every Developer/SRE :)
- **Make a living supporting and building Open Source** by offering
    - Infrastructure maintenance through collaboration
    - Minimizing operational costs through cost-sharing and consultant levels(1/2/3)
    - 24/7 monitoring and incident handling
- **Focus on what we're good at –** Linux and Kubernetes
- **Developers first** – we develop primarily in Golang – which helps us fix and improve anything – automation first :)

Thank You.

kubeaid

Obmondo

linuxaid