



Building a Kubernetes Platform That Scales from SaaS to Self-Managed

Lessons from GitLab's Internal Platform

About me

- Name: Florian Forster
- Joined GitLab in 2024
- Technical lead for the "Runway" platform team
- Passionate about free/open source software



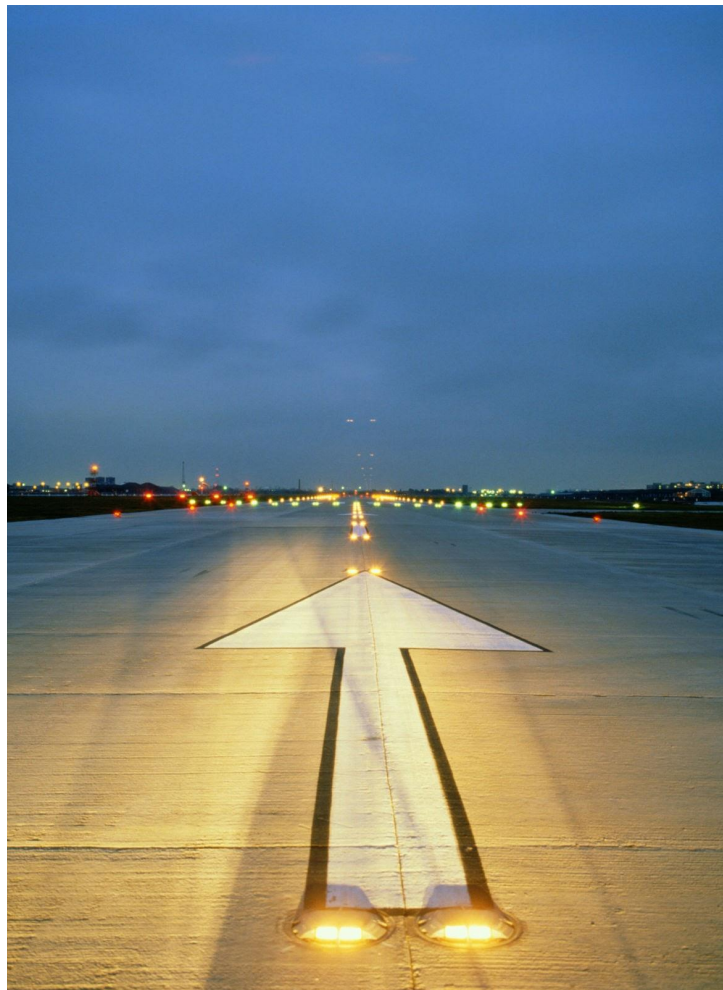
GitLab at a Glance

- GitLab is built on a monolithic Ruby on Rails application
- The monolith serves as the backbone for both SaaS and self-managed offerings
- It has over 2,500 employees and more than 5,000 open source contributors
- Scalability friction led to extracting features into separate services



What is Runway?

- Runway supports services alongside the monolith
- It is GitLab's internal platform for operating services
- Runway currently powers around a dozen services
- It helps engineering teams extract features into services



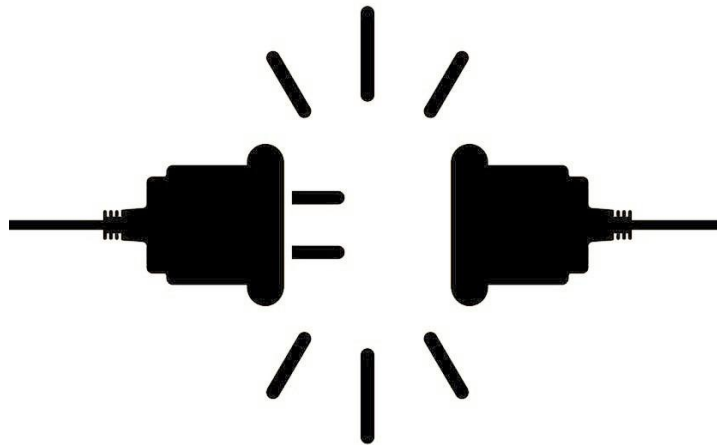
The v1 Architecture

- Cloud Run was the initial choice for the Runway platform
- Time to market was the highest priority when building v1 architecture
- Cloud Run provides a scalable, observable, production-grade HTTP service
- It offers low operational overhead with no cluster management needed



Self-Managed Problem

- Self-managed customers often have specific, non-negotiable requirements
- Data residency and security postures may prohibit external cloud dependencies
- Cloud Run is not a deployment option for these customers
- Choosing a proprietary runtime narrowed our addressable market
- Vendor-lock-in decisions are made for self-managed customers



Why Kubernetes?

- Kubernetes is an open standard API, not a vendor product
- Build on an abstraction customers can run anywhere
- In principle supports fully air-gapped environments
- Aligns with GKE and EKS, GitLab's existing operational environment



Architecture Shift

- Runway shifted from Cloud Run to a Kubernetes model
- Each service now has its own published Helm chart
- The same chart is used for internal and customer deployments
- This change created a dogfooding loop for installations



Versioning and Decoupling

- Runway services can have their own independent release cadence
- Each service generates its individual Helm chart for deployment flexibility
- Teams can release fixes without waiting for the main GitLab monthly release
- Independent releasability supports developer velocity and faster feedback loops
- Individual charts are being integrated into the umbrella GitLab Helm chart



Complexity Tradeoff

- Kubernetes introduces operational complexity for customers
- Customers handle storage, database, and networking setups
- GET automates infrastructure provisioning for GitLab services
- Optional complex features are packaged as standalone charts



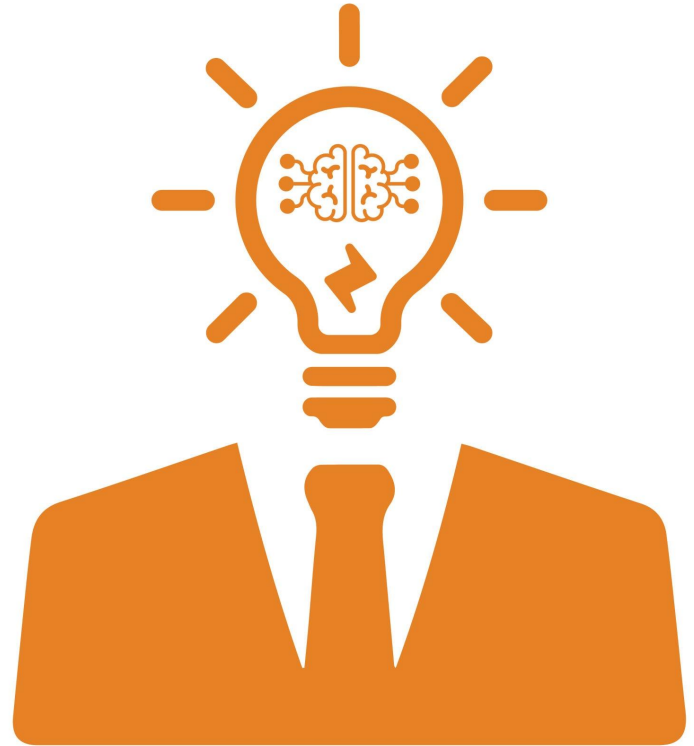
Engineering Experience

- The developer-facing experience remained consistent
- Teams still define services using YAML files in their repository
- Open standards ensure both internal and self-managed experiences share one foundation
- Consistency is a critical design goal for platform success
- Internal experience and self-managed offering share the same tooling



Key Takeaways

- Vended solutions are a genuine tradeoff, not automatically a mistake
- Revisit technology tradeoffs when the business changes
- Open standards help avoid vendor lock-in for customers
- Kubernetes as an API enables a common architectural layer
- A shared architecture streamlines internal and customer tooling





Questions?



Bonus slides

Labkit: Application Framework

- Labkit is an open source Go library developed by GitLab
- It is the expected application framework for Runway services
- Labkit handles cross-cutting concerns like structured logging and tracing
- It integrates with OpenTelemetry for standard telemetry output
- The library helps teams avoid writing boilerplate code for observability



common-ci-tasks – CI as Shared Library

- `common-ci-tasks` is a shared CI configuration library
- It is an open source, MIT licensed GitLab CI configuration library
- Services include it to get standard pipeline behaviors
- It prevents CI duplication across multiple service repositories
- Platform improvements update every service without manual intervention

