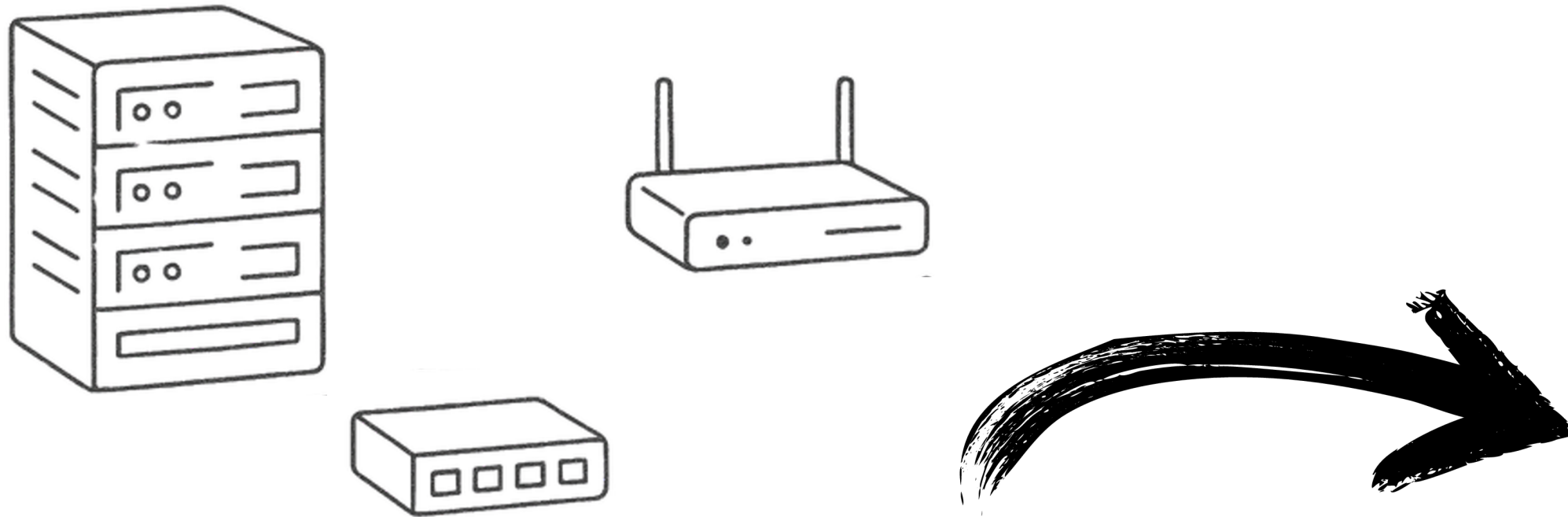


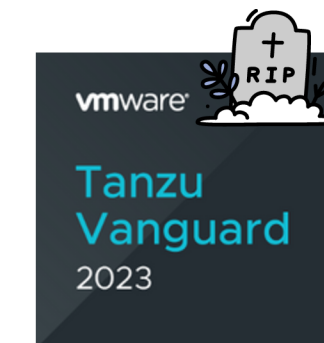
GitOps

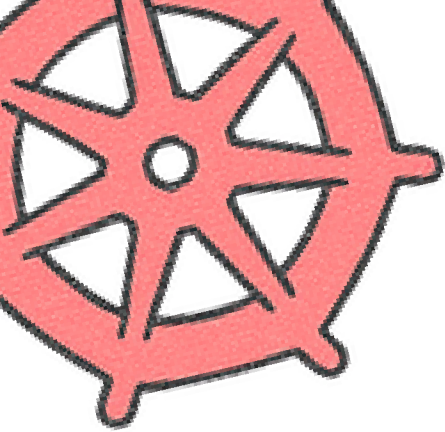
**The CNCF Movement That Changed
How We Operate Clusters**



Maximilian Marschall

- Unit Lead @ evoila
- CNCF Kubestronaut
- Development Lead
- Charity Work
- Homelab Warrior
- Racing Engineer

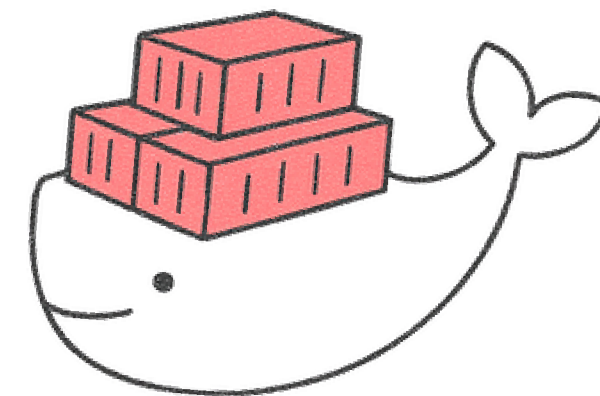
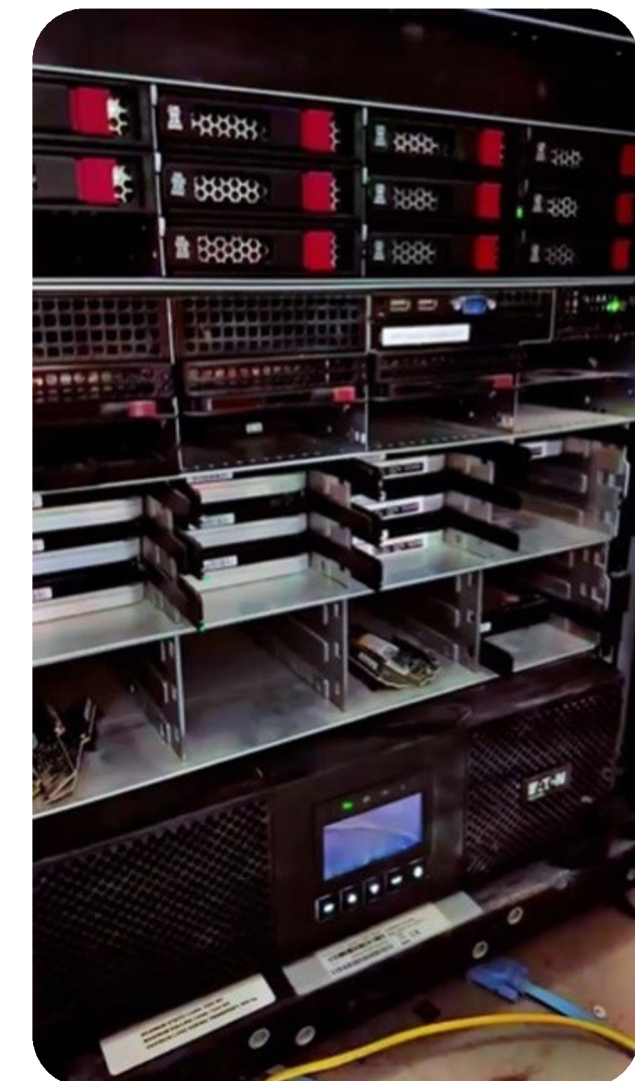




Homelab

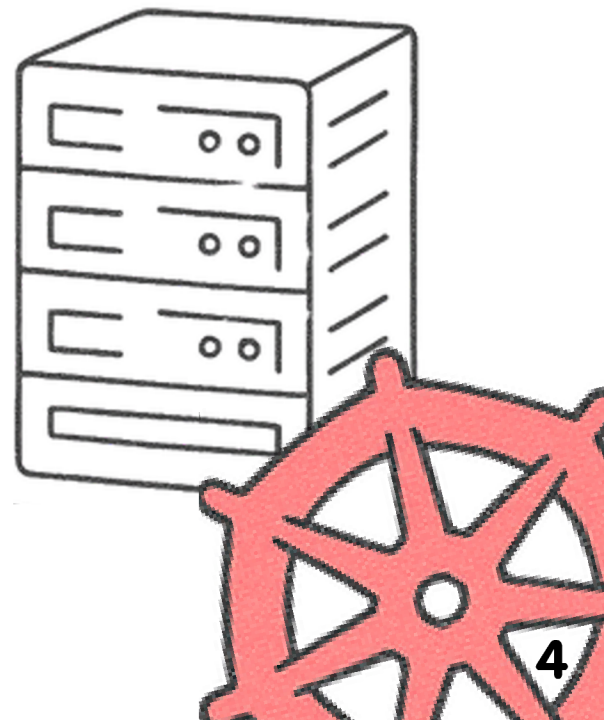
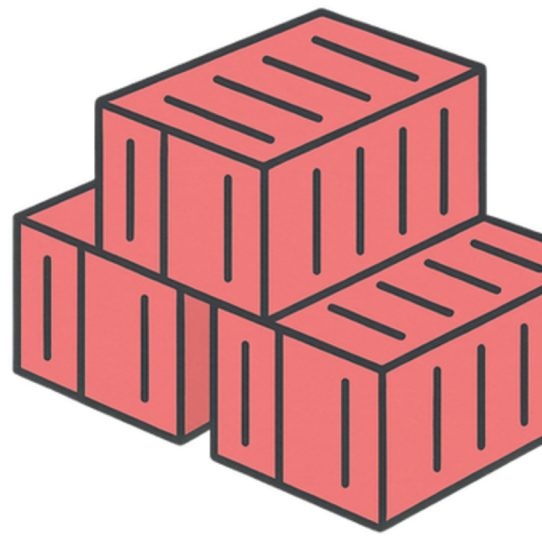


- 24 HE
- 4 Switches
- 6 Hosts
- 156 cores / 312 threads
- 2752 GB Memory
- 384 TB HDD
- 57 TB Flash

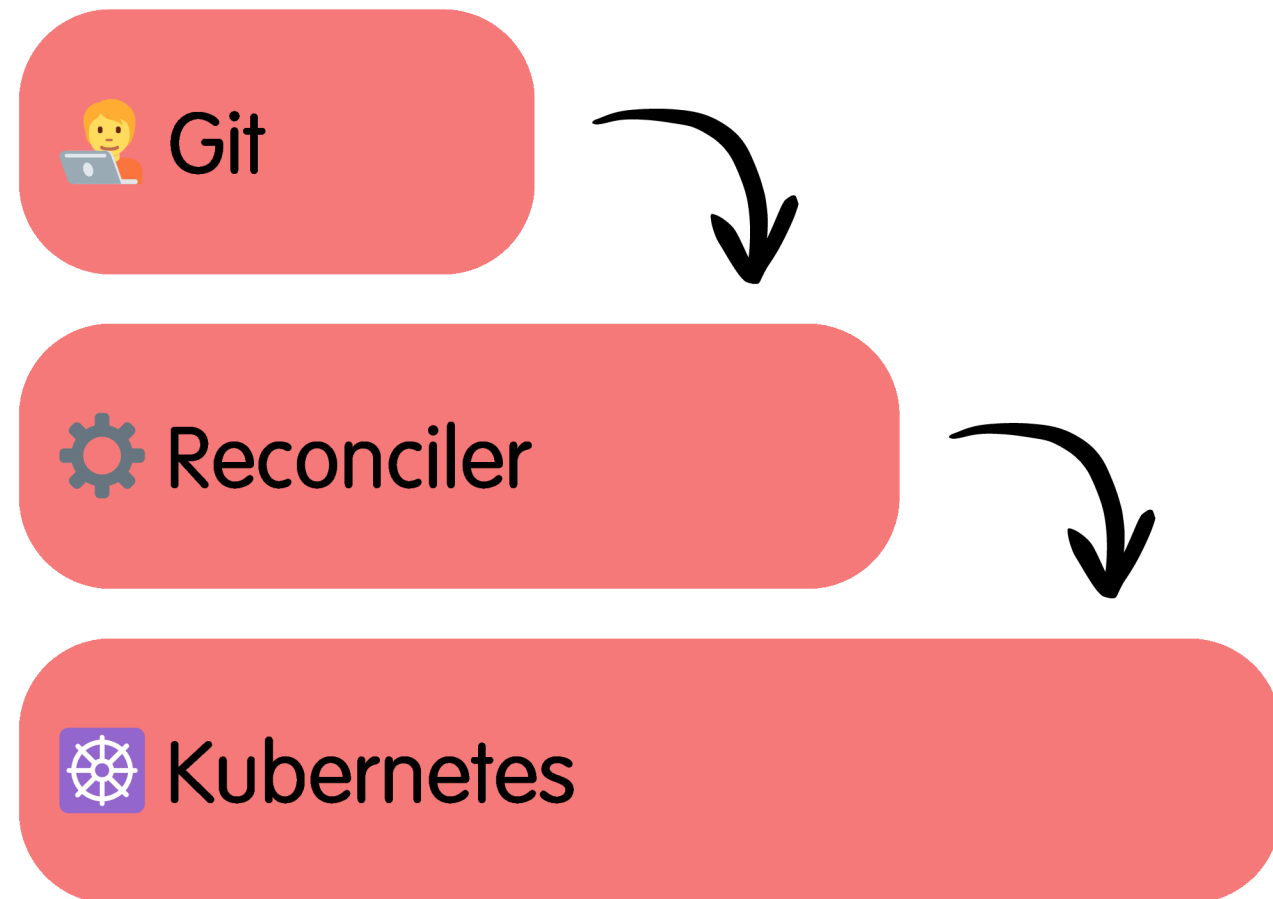
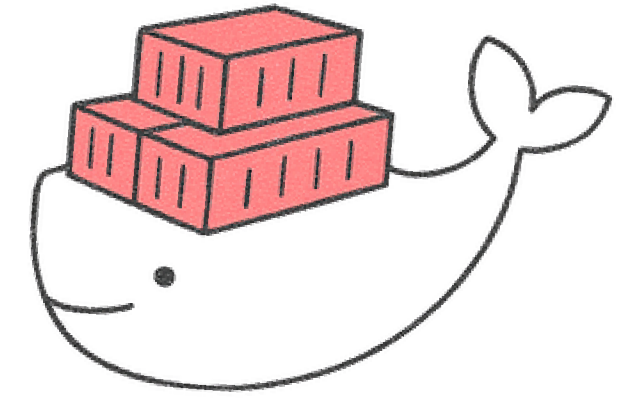


Target

- What is **GitOps**?
- Pre **GitOps** era
- How did it start?
- Future of **Gitops**
- Demo 🔥
- Recap



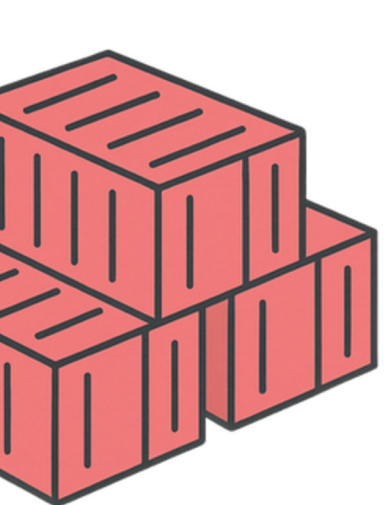
What is GitOps?



- State as Code → declarative
- Git as single source of truth → versioned
- Automation ensures consistency → reconciled
- Drift detection → observable

"If it's not in **Git**, it doesn't exist."

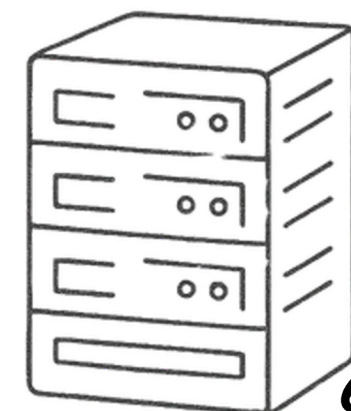




The Pre-GitOps Era

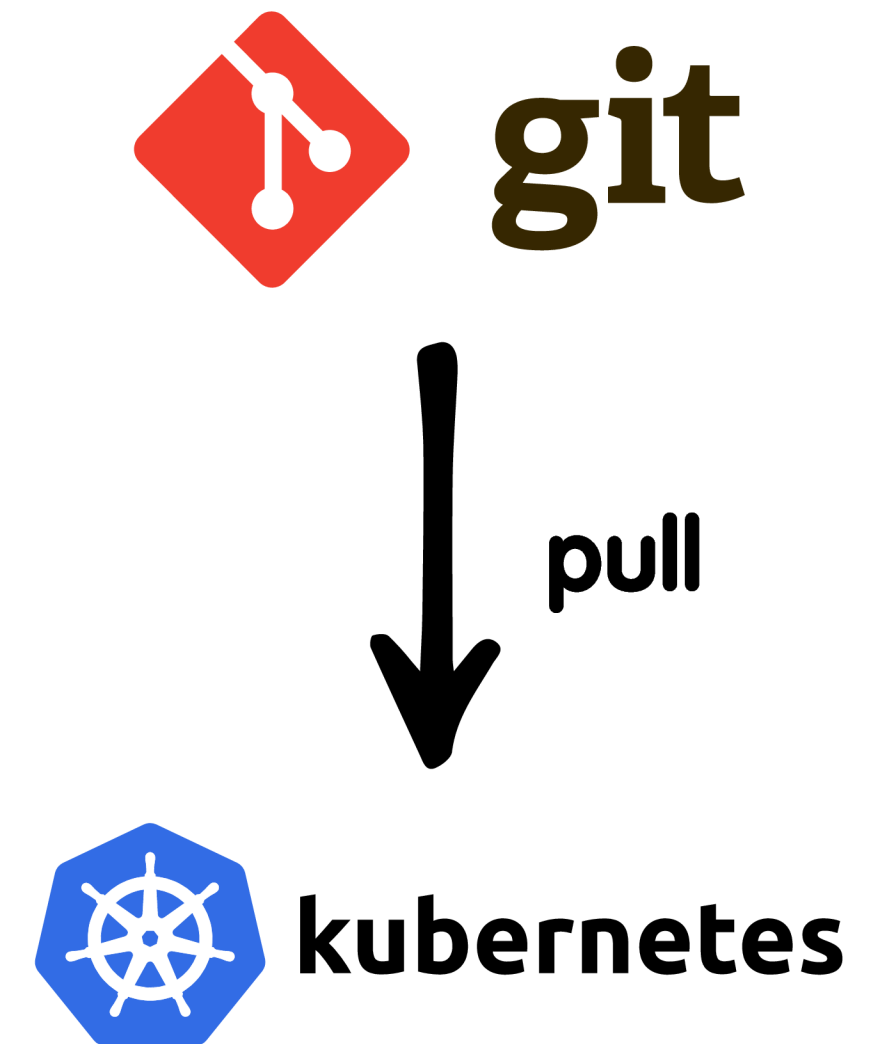
- 2015 Kubernetes GA
- 2016 Helm 2
- 2017 Pipelines everywhere
- 2018 Drift, snowflakes, chaos

- Manual `kubectl` and CI pipelines
- No single source of truth
- Config drift and human fatigue

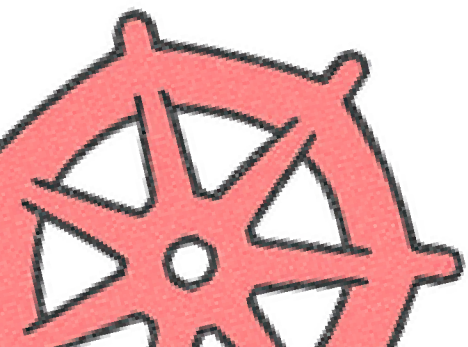


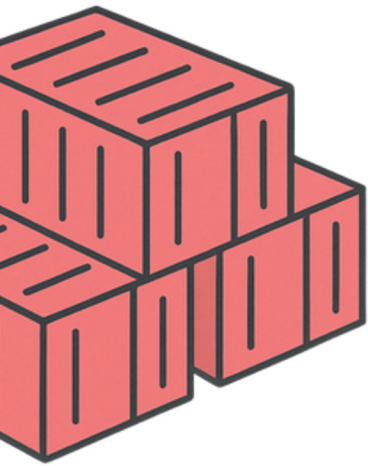
The Birth of GitOps

- Alexis Richardson (Weaveworks) invents "GitOps" 2017
- Flux starts the pull-based revolution
- Desired state stored in Git
- Clusters pull, not people push



"From pipelines to control loops."

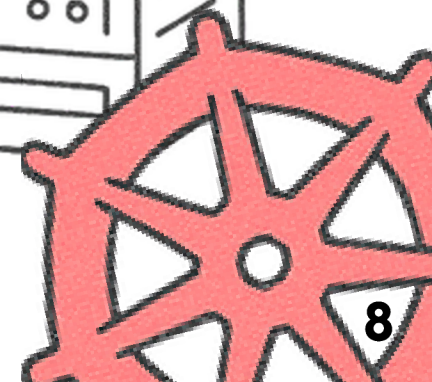




GitOps goes CNCF



- **CNCF** defines principles → Standardization
- **Flux** → **CNCF** graduated project
- **ArgoCD** → **CNCF** graduated project



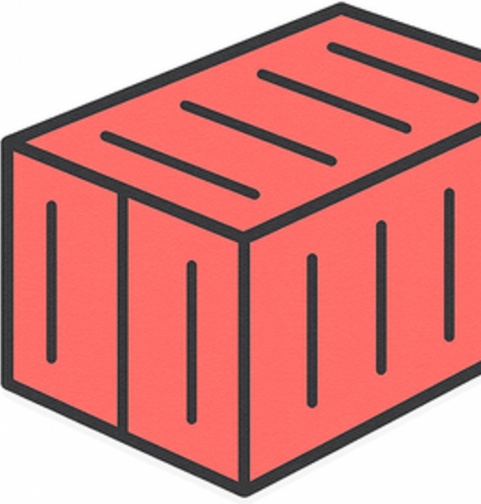
Lessons Learned

What worked

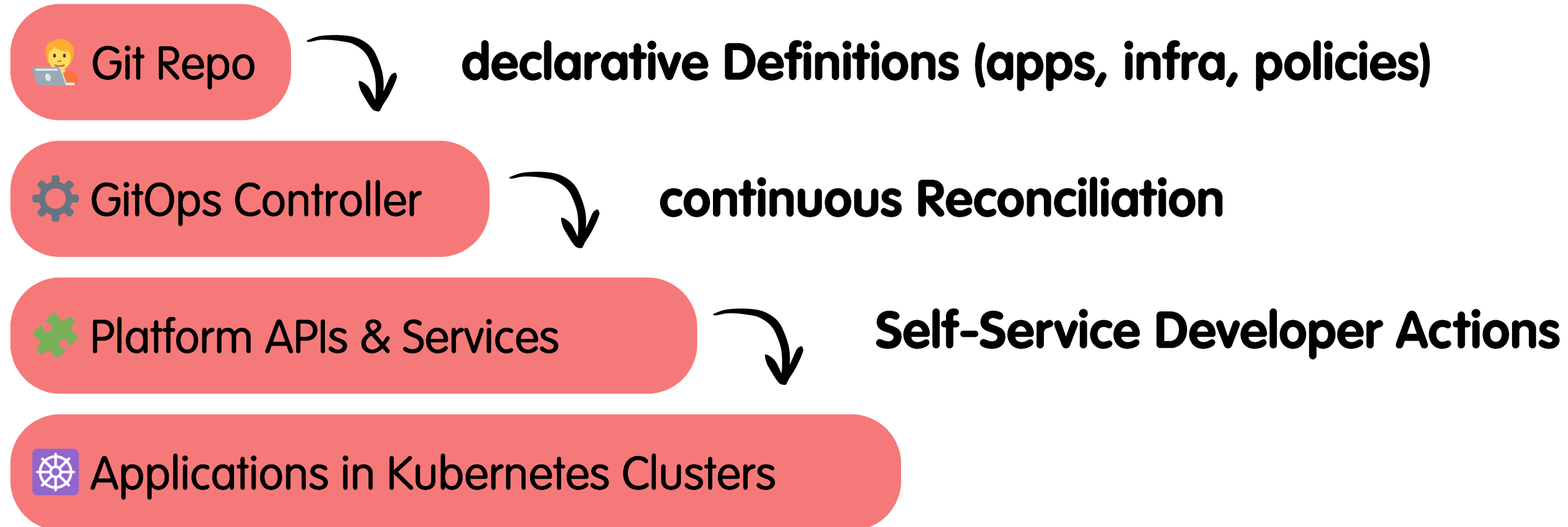
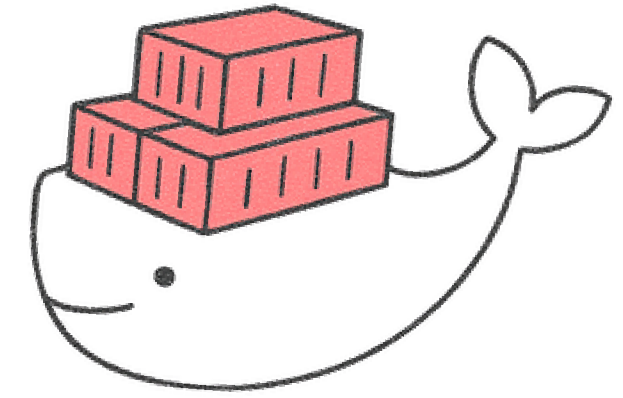
- Source of truth in Git
- Drift detection
- Easier rollbacks
- Auditable change process

What didn't

- Secret management
- Multi-tenancy
- Git bottlenecks
- Human process discipline
- Stage promotion



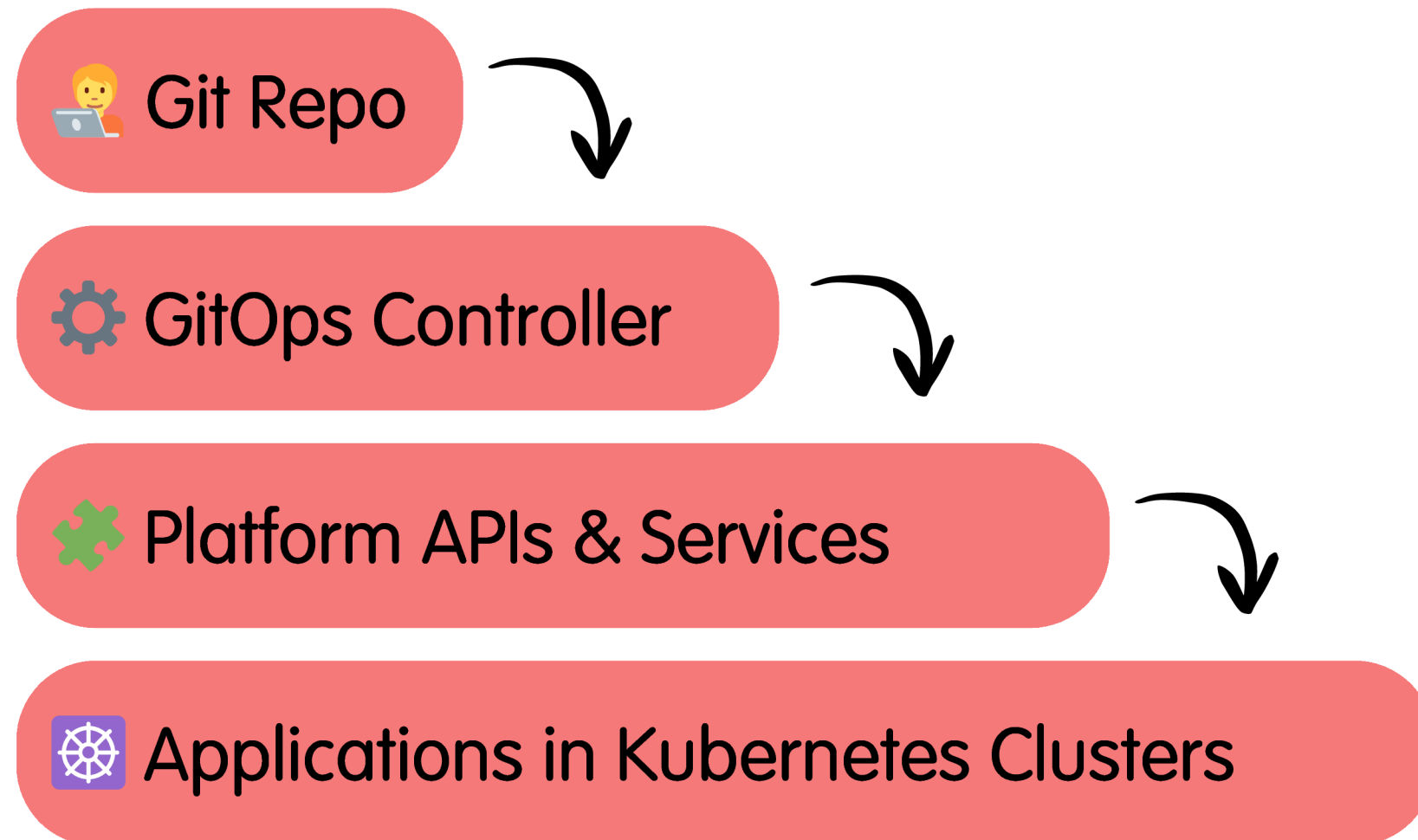
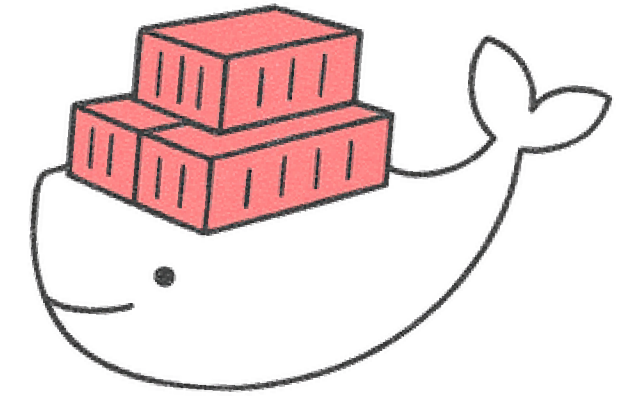
GitOps meets Platform Engineering



"GitOps (and PE) turned operations into a product experience."



GitOps meets Platform Engineering



- Platform engineers define golden paths in Git
- GitOps enforces consistency & compliance by design
- Developers deploy safely through self-service workflows
- Observability closes the loop into Git

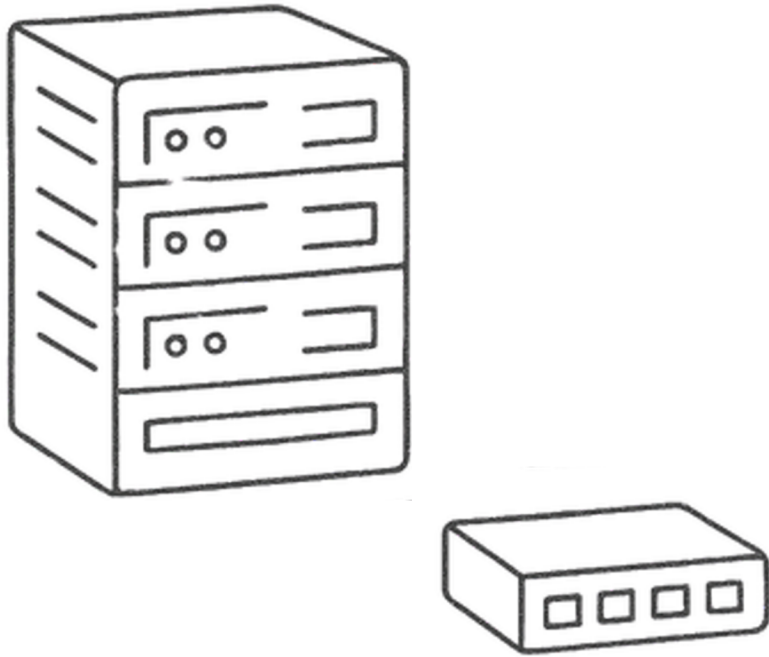
"GitOps (and PE) turned operations into a product experience."



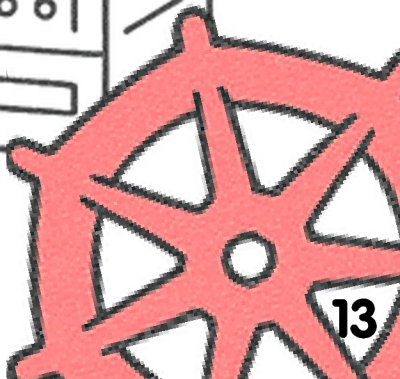
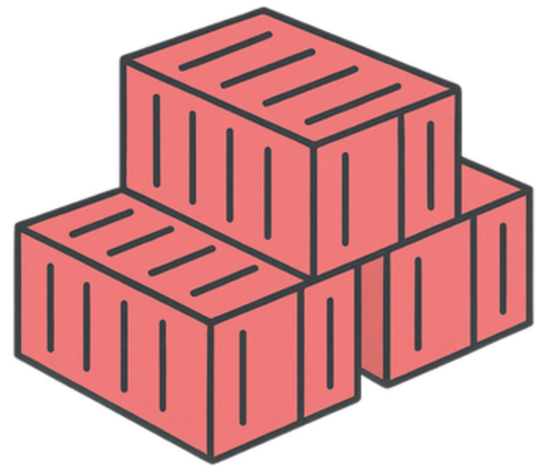
The future of GitOps

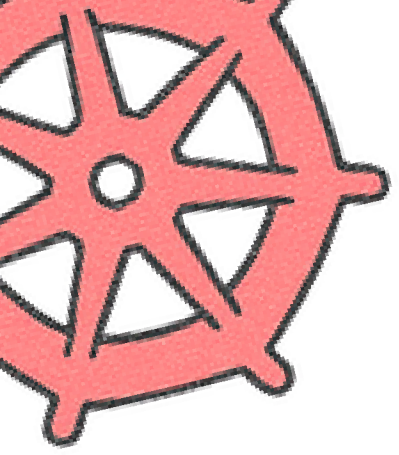
- **GitOps + Policies (Kyverno, OPA,...)**
- **Multi-cluster, multi-cloud orchestration**
- **AI-assisted drift detection**
- **GitOps at the edge and beyond Kubernetes borders**
- **Manifest hydration**





DEMO 🔥





Contact and Q&A

Whats your opinion on
the future of **GitOps**?

