CloudNative Aalborg presents

Istio in production
by Hans Duedal - Unity

Prometheus 101 - How to get started
by Rasmus Steiniche - neurospace

GitOps - Operations by Pull Request
by Kasper Nissen - Lunar

Hosted by

centrica

neurospace

LUNAR®

An Official
CLOUD NATIVE
COMPUTING FOUNDATION
Meetup Group

CloudNative Aalborg presents

Istio in production
by Hans Duedal - Unity

Prometheus 101 - How to get started
by Rasmus Steiniche - neurospace

GitOps - Operations by Pull Request
by Kasper Nissen - Lunar

Hosted by

centrica

unity

neurospace

LUNAR

An Official CLOUD NATIVE COMPUTING FOUNDATION Meetup Group

CloudNative **Aalborg**

#CloudNativeAalborg
#CloudNativeNordics

Join Slack  @ https://www.cloudnativenordics.com/

An Official
CLOUD NATIVE
COMPUTING FOUNDATION
Meetup Group

Agenda

- 17.00 Welcome
- 17.15 Istio in Production
- 18.00 Food
- 18.30 Prometheus 101
- 19.15 GitOps

An Official
CLOUD NATIVE
COMPUTING FOUNDATION
Meetup Group

A big thank you to this evening's host

And a word from tonight's sponsor

Kristian Majland Abelsen
Head of IT Development

centrica

And a word from tonight's sponsor

Kristian Majland Abelsen
Head of IT Development

centrica

Centrica Energy Trading

Cloud Native #4

11. December 2019

centrica

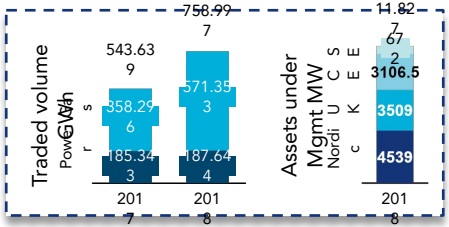
The company facts

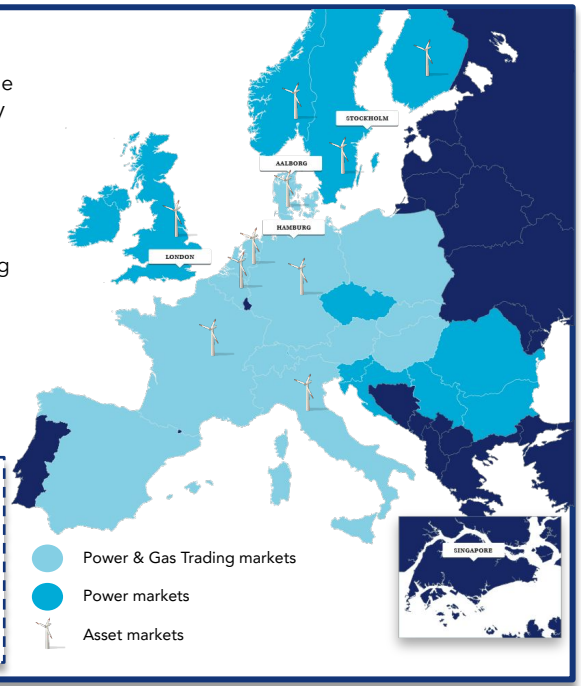
The story

The former Neas Energy was established by four public supply companies in 1998 with the purpose of purchasing electricity in the newly liberalised Danish energy market

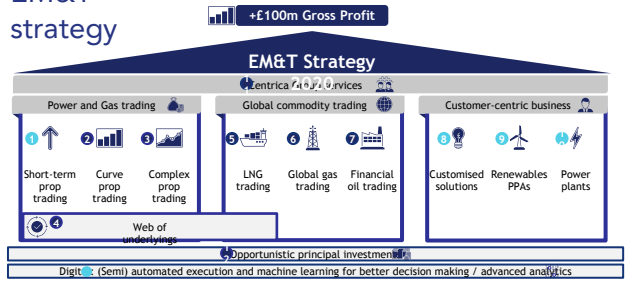
Today, together with Centrica Energy Marketing & Trading, we have transformed into a leading international energy asset management and trading company operating in power, gas and certificate markets across Europe

Our clients are Independent Power Producers; Power Plants, Renewable Energy Asset owners – Wind, Solar & Hydro – and large scale energy users



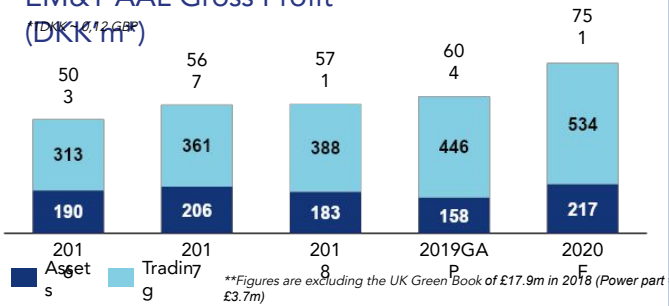


EM&T strategy



EM&T AAL Gross Profit (DKK m)

*DKK 1.12 GBP



**Figures are excluding the UK Green Book of £17.9m in 2018 (Power part = £3.7m)

centrica

Business area overview

Power trading

Customer driven business

Renewable assets
- Power balancing & intraday optimisation
- Hedging structures
- Renewable certificates

WiNRG

Aquila Capital

Power plants
- Power balancing & intraday optimisation
- Ancillary services
- Hedging structures

Welsh Power

VINDENERGI DANMARK

Consumption
- Hedging structures for utilities and corporate off-takers

Hydro

e.on

Google

Trading

Focus
- The short-term physical and financial power markets across Europe (22 markets)

Business Model
- Balancing and optimisation of assets
- Capacity-backed trading across borders
- Proprietary trading

Capabilities
- 24/7 setup (with office in Singapore)
- Intraday trading
- Day Ahead trading
- Cross Border trading
- Trading based on market insights, fundamental models and real-time input from meteorologists

Gas trading

Focus
- The short-term physical and financial gas markets across Europe (13 markets)

Business Model
- Optimisation of assets; storage and transport capacities
- Structured products
- Proprietary trading

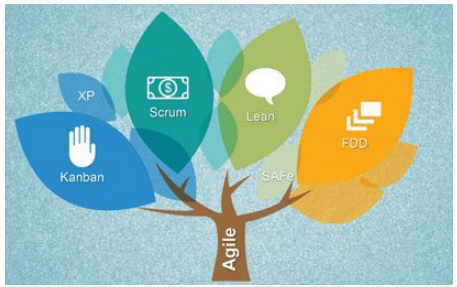
Capabilities
- 24/7 setup (with office in Singapore)
- Within-day trading
- Cross Border trading
- Curve trading
- Trading based on market insights and thorough analysis

Agility and High performance

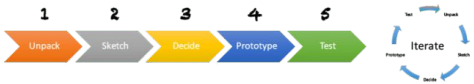
- "think big, start small, scale fast" improves agility, customer value and employee satisfaction

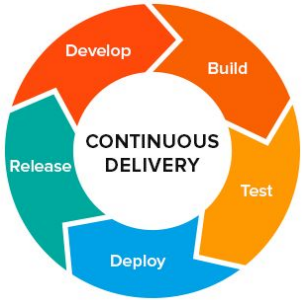
centrica

Working agile



Google Design Sprint





Stable operations

- >5 mio. yearly trades, incl physical delivery
- Very low incident count
- Very high availability 24/7/365 (+99,9%)

- ~250 weekly releases in production
- ~80% automatic test coverage
- +800+ independently releasable components
- Integrated with several OTS pr

ITIL®4

Being agile

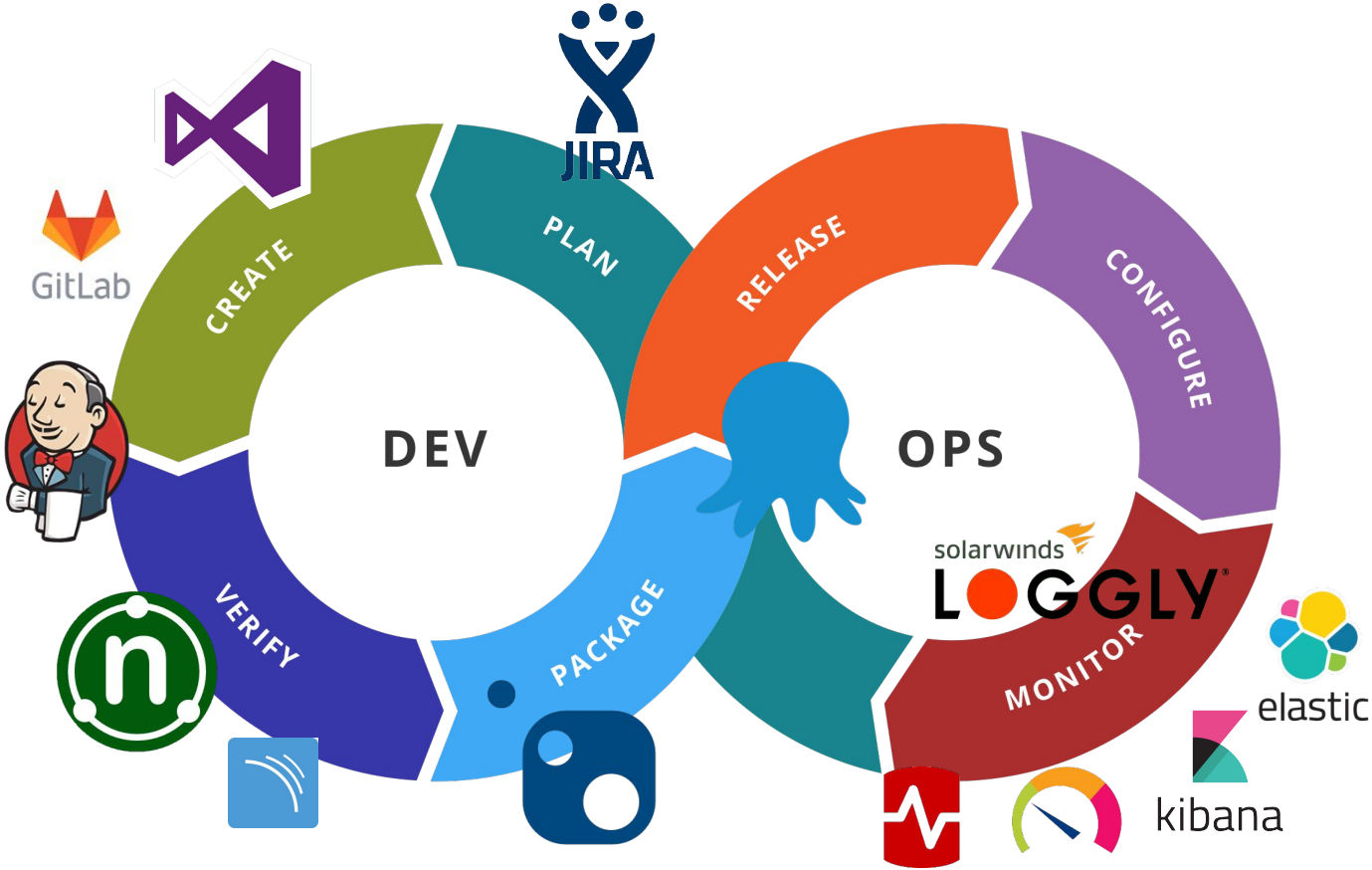
- igns product from Stakeholder inputs and ensures clear prioritization (design for change)
- Team delivers as soon and frequent as possible – get fast feedback through business engagement (MVP + Fast feedback)
- Team is responsible for full stack to support a given value chain (OTS + Bespoke + Data) (Fast time-to-market)
- Team is fully responsible for their software solutions, enforcing empowerment (Dev, Test & QA, Ops)

Continuous delivery

- Minimize the cost and technical risk associated with failure (Fail-fast)
- Release small and often (Microservices & DevOps)
- Department-wide dev. guidelines described in best practice (IT Architecture Manifest)
- Prefer smaller well-integrated services over monoliths (Fast time-to-market and reduce complexity)

DevOps Toolchain

centrica



Hybrid Cloud

centrica

Cloud strategy

Leverage our modern and recilient datacenters for stable application domains

Move to cloud to
- Obtain flexibility
- Utilize managed services
- Adjust ressources on demand
- Support high data growth

Cloud journey so far

Some years of experience on Azure virtual servers, Office365 etc.

Started incubator teams on Cloud services:
- AWS (Provider)
- Kubernetes (Orchestration, Containers)
- Terraform (Infrastructure as code)
- Redis (In-memory DB)
- MongoDB (NoSQL)
- RedShift (DW)
- SageMaker (MachineLearning)
- FireHose, Kafka (Data streaming)

Tweet! Tweet!

#CloudNativeAalborg
#CloudNativeNordics

Make some noise about our awesome community… @CloudNativeFdn, #kubernetes, etc.

CLOUD NATIVE
COMPUTING FOUNDATION

Cloud Native Nordics Community

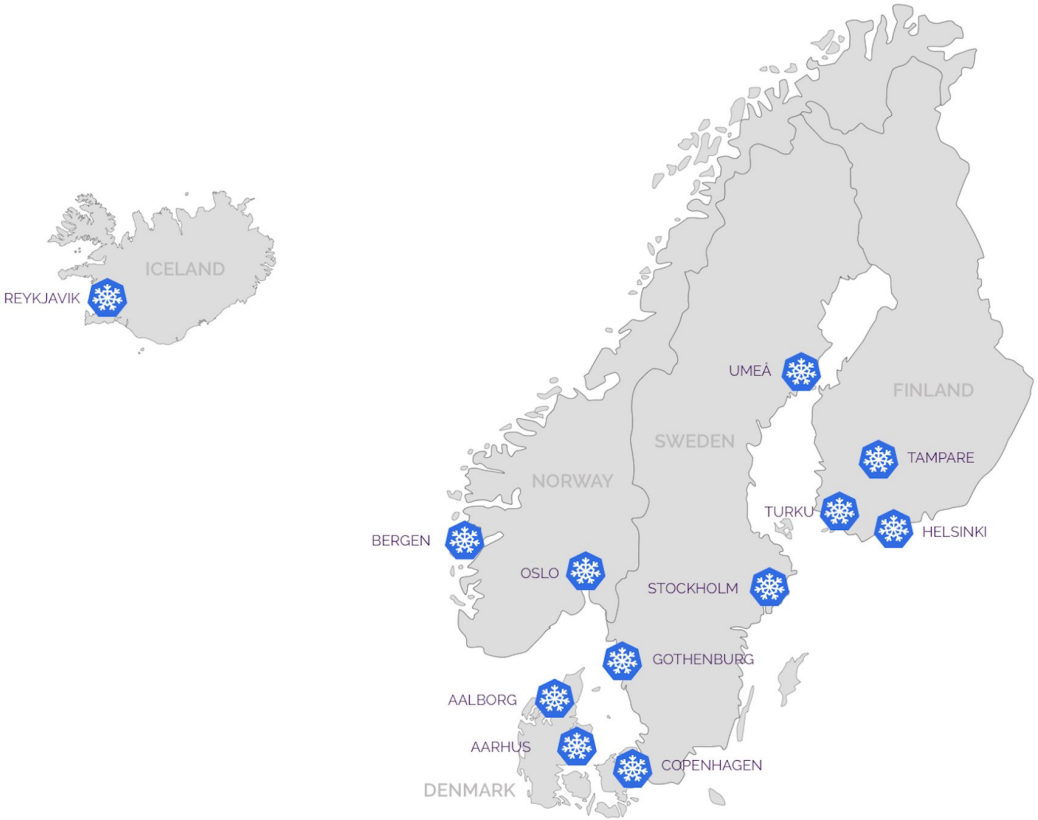
CLOUD NATIVE
NORDICS

CLOUD NATIVE
NORDICS

Continue the discussions and meet Cloud Natives from Denmark, Sweden, Norway, Iceland and Finland

www.cloudnativenordics.com

Cloud Native Nordics Community



CNCF Cloud Native Definition v1.0

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

Meet Niels - the new co-organizer

Niels Wittrup
Senior Software
Engineer and Architect
TrackUnit

Do you want to host a Meetup?

Arne Mejlholm
IT Developer
Sparnord

Simon Bengtsson
Systems Engineer
Sparnord

Camilla Beck Larsen
Systems Engineer
Sparnord

Niels Wittrup
Senior Software Engineer and Architect
TrackUnit

Allan Højgaard Jensen
Development Platform Specialist
Netic

come talk to us...

Do you want to speak at a Meetup?

Arne Mejlholm
IT Developer
Sparnord

Simon Bengtsson
Systems Engineer
Sparnord

Camilla Beck Larsen
Systems Engineer
Sparnord

Niels Wittrup
Senior Software Engineer and Architect
TrackUnit

Allan Højgaard Jensen
Development Platform Specialist
Netic

come talk to us...

Do you want to organize a Meetup?

Arne Mejlholm
IT Developer
Sparnord

Simon Bengtsson
Systems Engineer
Sparnord

Camilla Beck Larsen
Systems Engineer
Sparnord

Niels Wittrup
Senior Software Engineer and Architect
TrackUnit

Allan Højgaard Jensen
Development Platform Specialist
Netic

come talk to us...

Cloud Native Aalborg

We need your feedback...

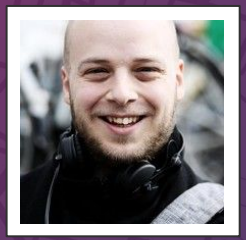
- what can we do better?
- any ideas for future meetups?

- provide feedback on: 

Istio in Production

/by Hans Duedal

SRE - unity



An Official CLOUD NATIVE COMPUTING FOUNDATION Meetup Group

Istio In Production

War-stories from an eventually successful Istio production rollout, for ML workloads in the cloud.

Hans Duedal

Software Engineer

Mobile +45 26 15 99 17
Twitter @hansduedal

hans.duedal @unity3d.com | @gmail.com

unity

Before  I was an SRE at 

This talk & demo relates to work done then.

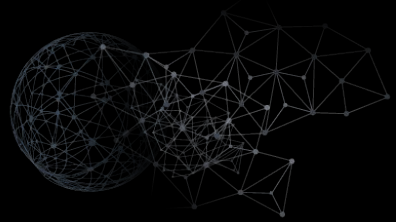
Agenda

- What is a service mesh
- What is Istio
- D.I.Y "Istio"
- Managed Istio
- Self-Deployed Istio
- Brief intro to Kiali
- Autoscaling Demo
- Lessons learned
- How to get started

Service Mesh

You get a metric! You get a metric!
Everyone gets a metric!

What's a service mesh?



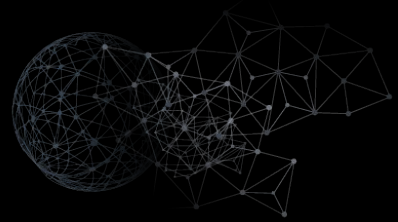
A service mesh describes the network of microservices that make up applications and the interactions between them.

If you, like Monzo bank, have >500 microservices you definitely need service mesh tech in order to understand and manage the complexity.

A service mesh often provides

service discovery	load balancing	failure recovery
metrics	monitoring	A/B testing
canary releases	rate limiting	access control
end-to-end authentication		

Why use a service mesh?



Some of the aforementioned challenges can be solved at the application layer, but support, and the right tool for the job, can differ depending on languages.

You want to use a service mesh to avoid

- Bloated service code
- Duplicated work to make services production-ready
  - load balancing, auto scaling, rate limiting, traffic routing
- Inconsistency across services
  - retry, tls, failover, deadlines, cancellations, etc for each language / framework
- Diffusing responsibility of service management

Service Mesh Technologies

https://l.cncf.io/ lists four, but istio is where the most action is at

kuma

vamp

linkerd

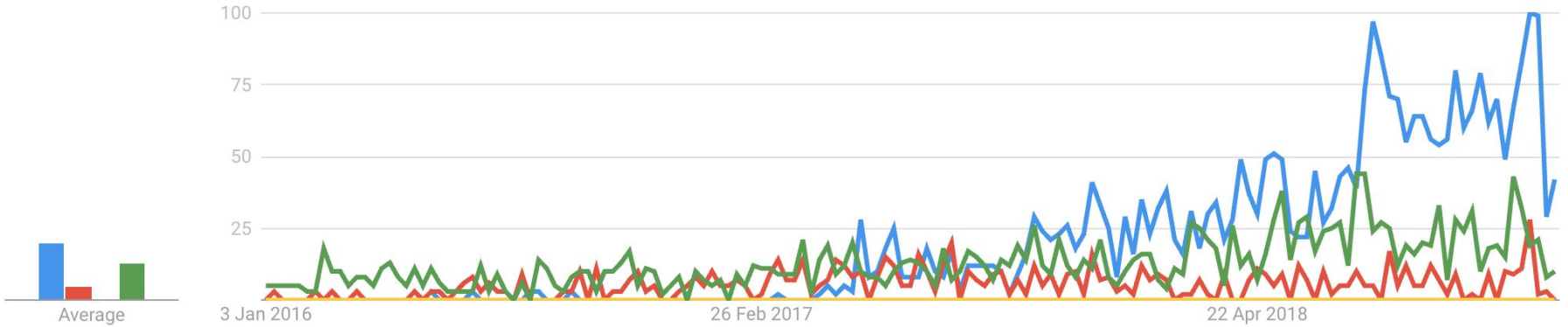
istio

Google Trends | Compare

istio	linkerd	vamp.io	service mesh	+
Search term	Search term	Search term	Search term	

United States | 01/01/2016 - 02/01/2019 | All categories | Web Search

Interest over time



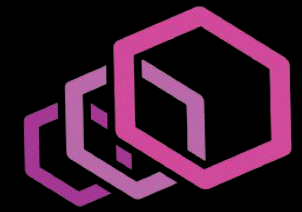
What is Istio?

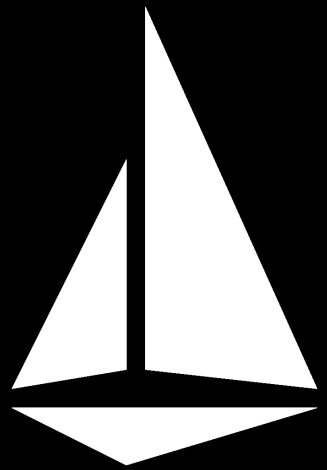
TL;DR; Really smart plumbing

Istio

Istio is *the* service mesh, made by Google and powered by Envoy

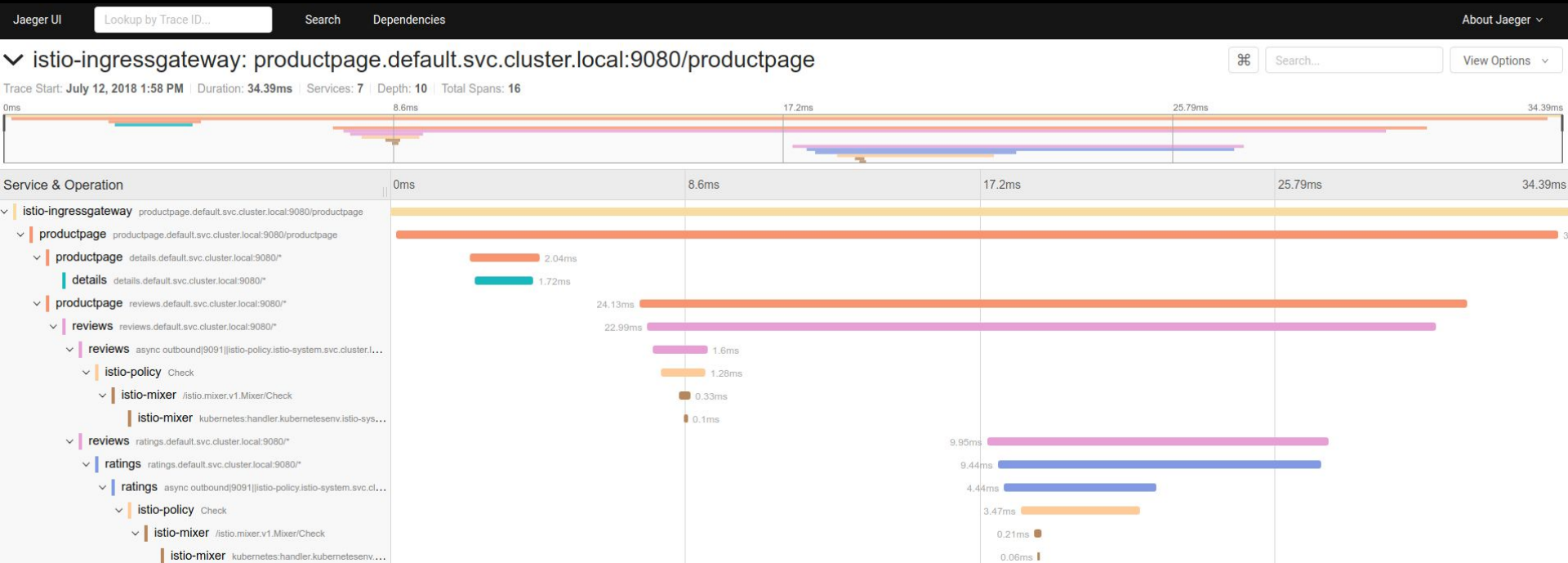
- Observability (what gets people hooked on service metrics)
  - Consistent Metrics without instrumenting apps
  - Trace flow of requests across services
- Resiliency (control over chaos)
  - Systematic fault injection, timeout and retries with timeout budget
- Traffic Control (content-based traffic steering)
  - Traffic splitting / steering, ingress & egress routing
- Security
- Policy Enforcement





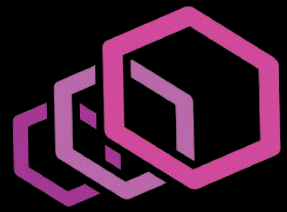
Distributed Tracing & Service Meshes

Service meshes provides distributed tracing, but without modifying your code or buying an expensive APM tool.



Envoy Proxy

- Developed at Lyft – Used at top cloud companies today
- The proxy ("webserver") behind the famous Istio service mesh
  - but we call it the Data Plane
- Configuration is based on microservices
  - You can build a lot of intelligence into the proxy this way
  - Supports gRPC for low overhead and fast RPC calls
- Extremely performant non-blocking C++ codebase

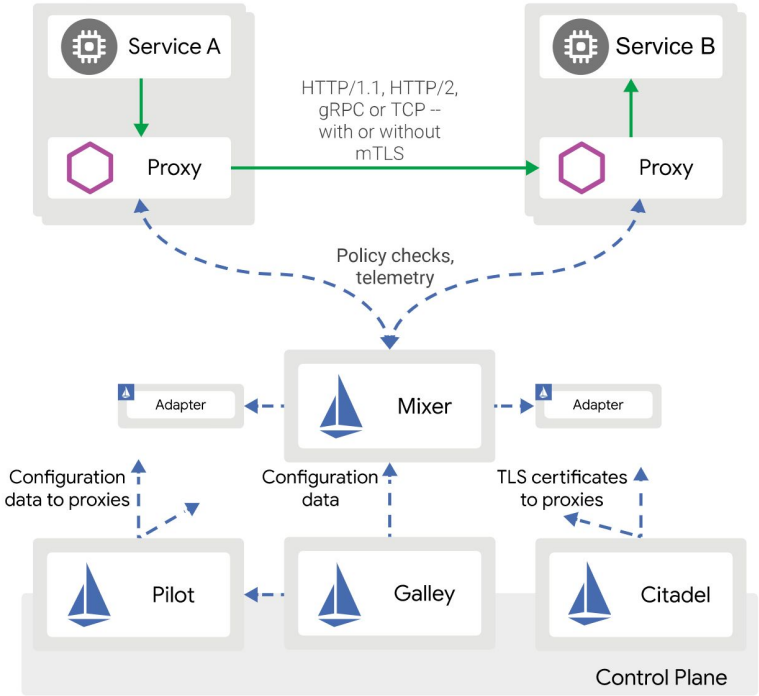


Istio Architecture

Istio is made up of several microservices

Mixer: access control, policies & telemetry
Pilot: service discovery, intelligent routing and resiliency
Citadel: service-to-service and end-user authentication
Galley: platform (ie. kubernetes) configuration layer

You don't have to run all of them, ie. you can run only Pilot if you just want service discovery in the control plane.

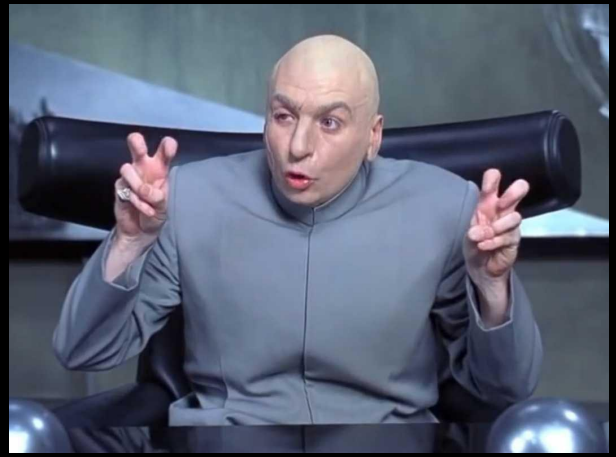


unity

D.I.Y. "Istio"

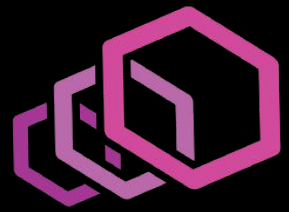
How we started with service meshes

TL;DR; We didn't



Envoy @ Machine Learning

- We used manual envoy sidecars as an intermediate step towards service meshes
- Not quite a service mesh, but we do have envoy both at the edge and as a "sidecar"
  - Adds tracing
  - Rate Limiting
  - Retries
  - Metrics & Tracing
- Only DNS based service discovery, and no IPTables injection meant connecting to localhost:X for service X and localhost:Y for service Y etc.



Pros & Cons of Starting Only Envoy

Pros

- De-risk the project
- Less moving parts, easier debugging
- Gives vital experience with Envoy and it's config, how to debug it etc.

Cons

- Requires you to write more yaml
- Hard to keep track of which localhost port goes where
- Doesn't scale for many µ-services

Classic MVP Tactic

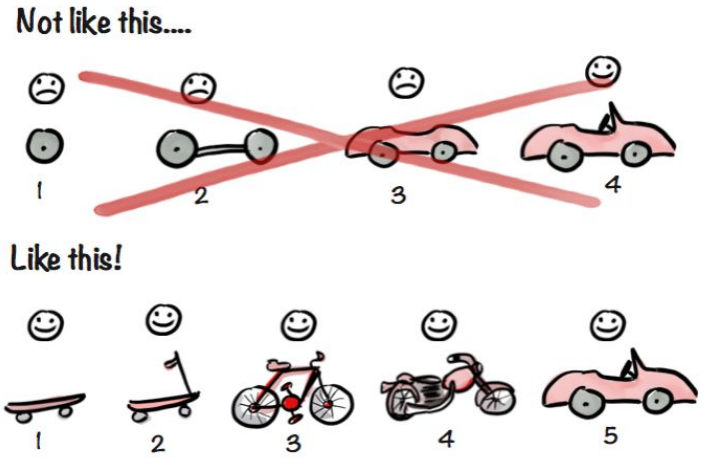
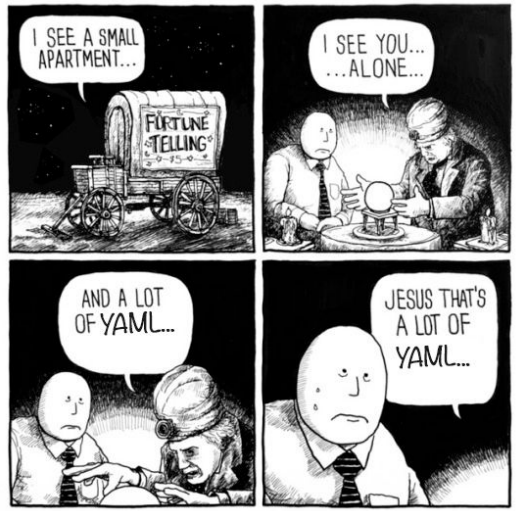


Illustration by Henrik Kniberg

Less Yaml is better

136 manifests/mlservice.yaml

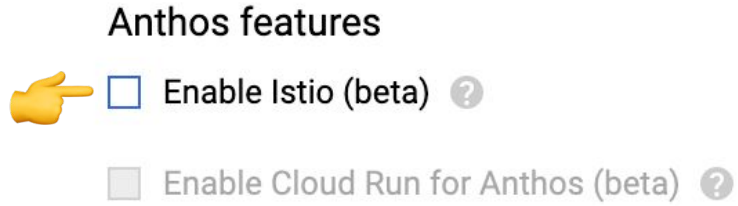
```
@@ -7,72 +7,12 @@ metadata:
      app: mlservice
  spec:
    ports:
-   - port: 9000
+   - name: grpc
+     port: 8989
      targetPort: 50051
    selector:
      app: mlservice
    type: ClusterIP
-  clusterIP: None
-
----
-apiVersion: v1
-data:
-  mlservice.yaml: |
-    static_resources:
-      listeners:
-      - address:
-          socket_address:
-            address: 0.0.0.0
-            port_value: 8989
-        filter_chains:
-        - filters:
-          - name: envoy.http_connection_manager
-            config:
-              codec_type: auto
-              stat_prefix: ingress_http
-              route_config:
-                name: local_route
-                response_headers_to_remove:
-                - "x-envoy-overloaded"
-                virtual_hosts:
-                - name: service
-                  domains:
-                  - "*"
-                  routes:
-                  - match:
-                      prefix: "/"
```



Managed Istio

TL;DR; We tried it so you don't have to.

GKE One-Click Istio



You can enable managed istio with one-click / terraform on GKE

The Good: It works, mostly
Comes with Stackdriver support pre-installed
Batteries included; comes with prometheus & grafana

The Bad: You get vanilla Istio
You have to match Istio and GKE versions
No kiali

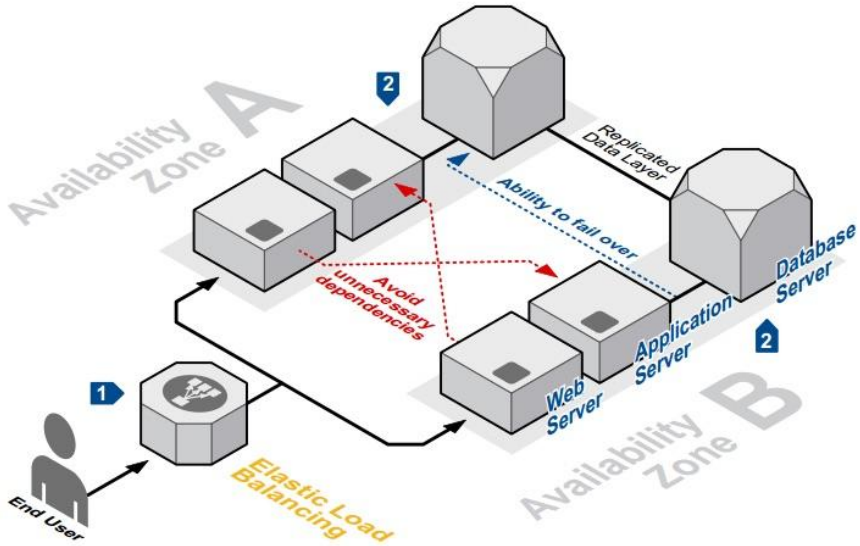
The Ugly: You can't set basic things like anti-affinity for HA deployments
Google's version of GKE & Istio is way behind

Anti Affinity

Certain Istio Services should be run in HA, like "mixer" and "pilot".

You can't change most properties for the managed istio, but you can and should set replicas and adjust HPAs.

Unfortunately there is no anti affinity ensuring that the additional pods don't get scheduled in the same zone / on the same node.



How to design HA applications in the cloud

Version Support

Istio is very much a bleeding edge product, and moving at a rapid pace.

Google only supports specific versions of GKE & Istio together:

cloud.google.com/istio/docs/istio-on-gke/installing

Currently you're stuck with K8S 1.14 & Istio 1.1.16[1] - the latest Istio release is 1.4.2 which means you are missing out on a ton of bugfixes and even breaking changes.

1) If you are on the rapid release channel, you get k8s 1.16 & istio 1.2.7

Self Deployed Istio

TL;DR; Just helm deploy it, it'll be fine

Istio with Helm

I highly recommend taking this route when installing istio, and going for the latest (stable) release.

Good guide on https://istio.io/docs/setup/install/helm/

We didn't actually use tiller, helm is only required to generate the 4785 lines of yaml for kubernetes.

Google has a guide on this as well
cloud.google.com/istio/docs/how-to/installing-oss

Istio Configuration

The defaults are fairly sane for most installations.

Don't edit the helm templates and/or their values, you should maintain a single values.yaml file that overrides the few defaults you want. Ours is a manageable 56 lines, mostly anti-affinity and replica counts.

We use a git submodule to track the upstream templates.

You can find ours at github.com/e-conomic/vml-istio

When you deploy/upgrade istio, all the pods need to redeploy to get (new) sidecars; `kubectl rollout restart` is really good for this.

Kiali

TL;DR; It's like catnip for devops

Kiali

Often featured in talks and demos, Kiali provides service mesh observability

Answers

- What microservices are part of my Istio service mesh?
- How are they connected?
- How are they performing?

Pulls data from Istio Config, Prometheus Metrics and Jaeger Tracing

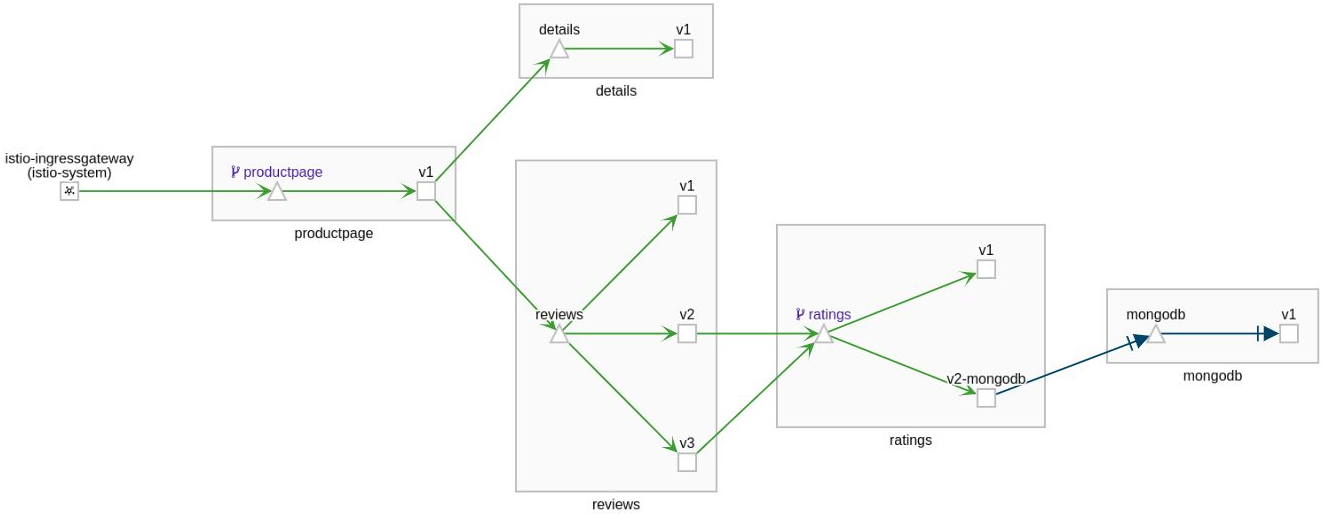
Overview
Graph
Applications
Workloads
Services
Istio Config
Distributed Tracing

Namespace: bookinfo

Graph

May 24, 12:30:28 ... May 24, 12:31:28

Versioned app graph | No edge labels | Display | Find... | Hide... | Last 1m | Every 15s



Legend

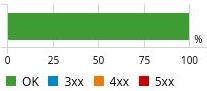
Hide

Namespace: bookinfo
applications, services, workloads

Current Graph:
9 apps
5 services
14 edges

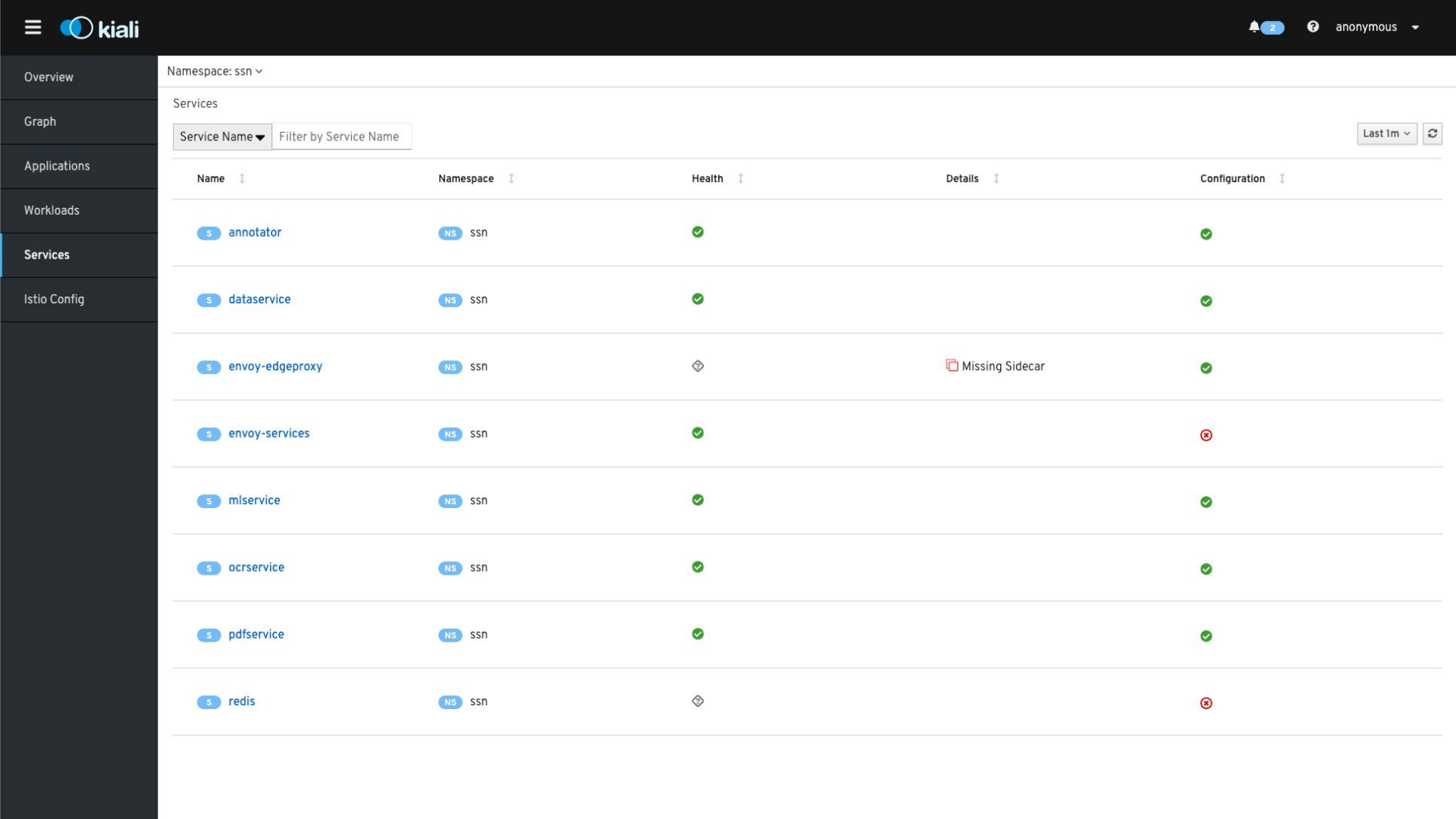
HTTP Traffic (requests per second):

Total	%Success	%Error
3.62	100.00	0.00

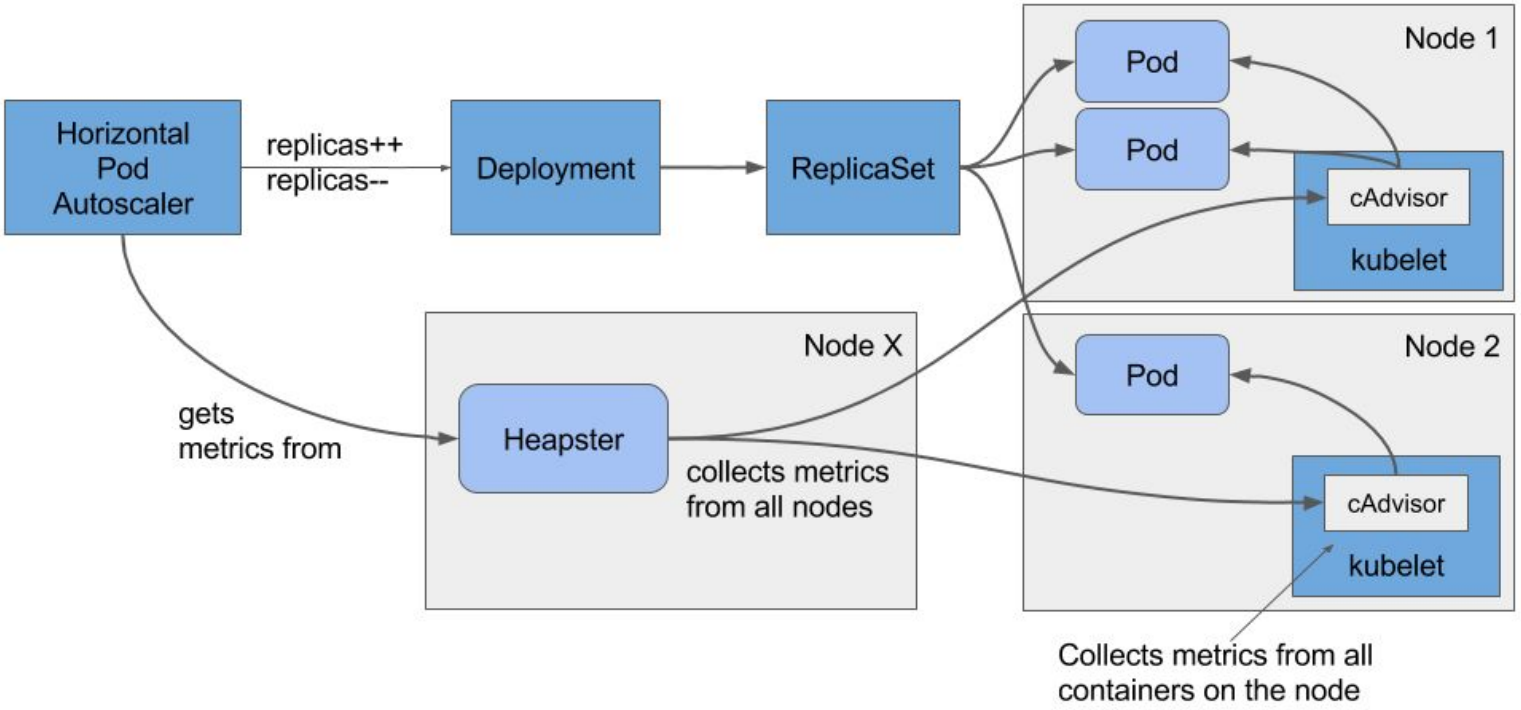


HTTP - Total Request Traffic min / max:
RPS: 3.53 / 3.73 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
Sent: 114.40 / 171.60 B/s
Received: 93.33 / 140.00 B/s


kiali
2
anonymous
Overview
Graph
Applications
Workloads
Services
Istio Config
Namespace: ssn
Services
Service Name
Filter by Service Name
Last 1m
Name
Namespace
Health
Details
Configuration
annotator
ssn
dataservice
ssn
envoy-edgeproxy
ssn
Missing Sidecar
envoy-services
ssn
mlservice
ssn
ocrservice
ssn
pdfservice
ssn
redis
ssn

Kubernetes Refresh

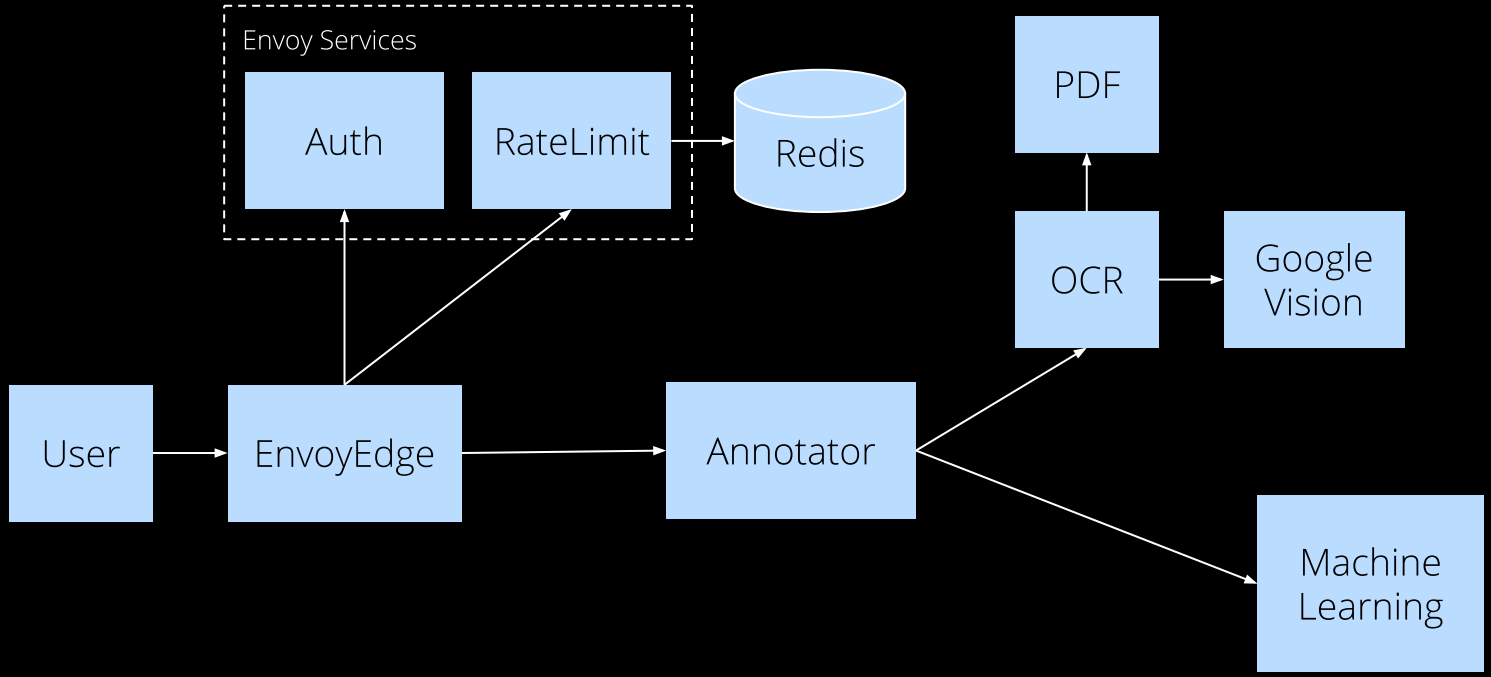


unity

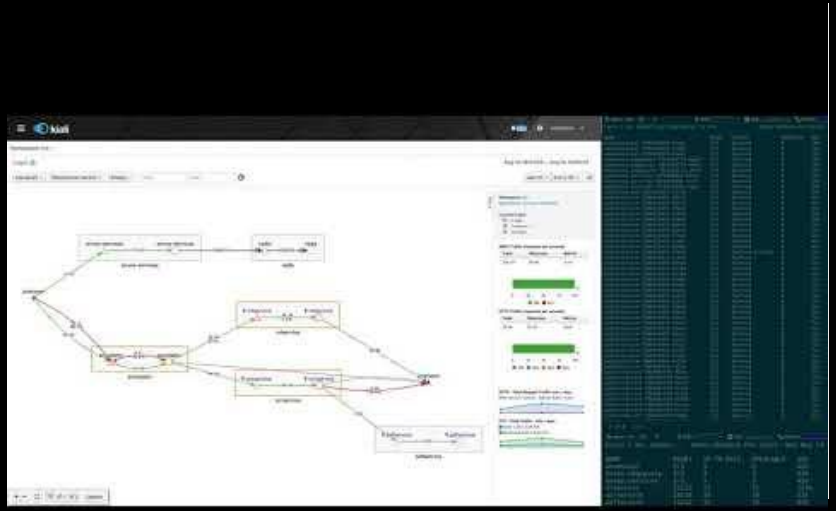
A bit of context

Quick Intro to Visma Machine Learnings infra

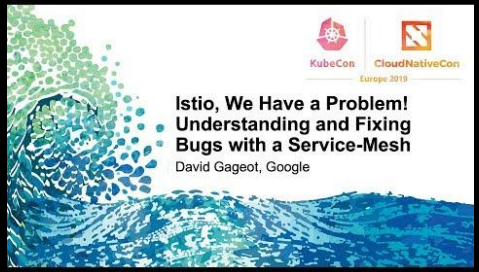
High Level Overview



Auto Scaling Demo



https://youtu.be/_0SvXvyPlOw



Also See
Fast paced service mesh demo

https://www.youtube.com/watch?v=9CQ0PMiOGhg&list=PLj6h78yzYM2PpmMAnvpvsnR4c27wJePh3

Lessons Learned

...and tips for debugging Istio

TL;DR; RTFM

Lesson 1: Name your service ports

Arguably the most important thing is to name your service ports properly

Done right, deploying istio will just work™

All "containerPorts" in kubernetes will be intercepted by Istio Sidecars, so if you have something you don't want intercepted, don't list it as a containerPort and vice-versa.

Wrong

```
spec:
  ports:
  - name: envoy-webserver
    port: 50051
    targetPort: 50051
```

Correct

```
spec:
  ports:
  - name: grpc
    port: 50051
    targetPort: 50051
```

Lesson 2: Istio intercepts egress

By default istio will intercept all egress traffic (ie. calls to the internet). This is potentially a nice feature if you are forced to by the corporate infosec compliance guys, but the rest of the time it's a pain in the butt.

I have seen various weird or unexpected behavior from essentially putting a reverse-proxy in front of everything on the internet, including extremely slow apt-get updates, borked calls to Amazon S3 etc.

Do yourself a favour, include this in your helm values.yaml:

```
global:
  proxy:
    includeIPRanges: "10.0.0.0/8"
```

Lesson 3: Anti-climatic after deploy

After you successfully deploy istio, and your services are now longer throwing errors (see Lesson 1), it will feel very anti-climatic. Since everything a service mesh does is transparent to the application, how do you even know if it's working correctly?

Conclusion: You ~~want~~ need Kiali

Kiali requires jaeger and prometheus, but we found that using the defaults is easy and more than adequate for most needs. We just federate the istio prometheus to our main prometheus for longer retention.

Don't go overboard with backends for jaeger.

Lesson 4: istioctl is nice

Useful commands

`istioctl proxy-status|ps` Have the config been sync'd to the sidecars?

```
istioctl pc <clusters|listeners|routes|endpoints|bootstrap>
<pod-name[.namespace]>
```

Run it with `-o json` to get the current Envoy configuration

Lesson 5: Drop down to envoy layer

Essentially you are expressing high level concepts using the Istio CRDs which boil down to Envoy config. It's very useful to be able to inspect the config directly so you can understand what's going on.

Thus you need to understand Envoy first.

You can port-forward port 15000, and then access the envoy admin interface.

You can toggle envoy debug logs

```
kubectl mypod-1234-abcf -c istio-proxy -- curl -XPOST -s -o
/dev/null \

http://localhost:15000/logging?level=debug
```

Lesson 6: Running your own ingress

Istio comes with an "ingress-gateway" for ingressing traffic into your service mesh.

We started running our own Envoy, and some things did not translate into Istio's CRDs. Istio supports "EnvoyFilter" for injecting custom envoy config, but it was very limited in Istio 1.2 – since Istio 1.3 it's much more capable.

We prefer running our own Ingress Gateway, and use Istio's Pilot service for service discovery so it will ingress traffic to the service mesh.

Lesson 7: Distributed tracing is nice

In order for istio to know the sequence (trace) in which your services is calling each other, you need to propagate tracing headers.

See https://istio.io/faq/distributed-tracing/

Essentially you need to forward these headers on subsequent http calls in your microservice stack: x-request-id, x-b3-traceid, x-b3-spanId, x-b3-parentspanid, x-b3-sampled, x-b3-flags, b3

You should get a nice directed kiali graph if you do.

How to get started

TL;DR; Just start

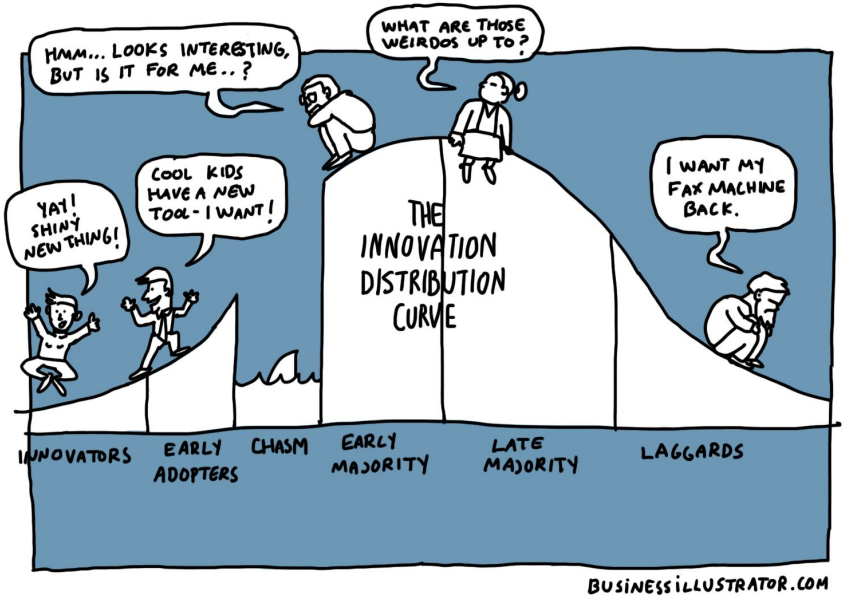
Just Start

Now is a pretty good time to start.

Istio 1.0 & 1.1 was rough, but with 1.2 and 1.3 things have stabilized.

If you tried Istio earlier I encourage you to try again.

Strongly consider getting your feet wet with Envoy first.



unity

Resources

Find out what you can do in this demo;
https://youtu.be/9CQ0PMiOGhg

Katacoda has courses on Istio;
https://www.katacoda.com/search?q=istio

Istio on GKE has good documentation if you're into that kind of thing; https://cloud.google.com/istio/docs/istio-on-gke/overview

Istios documentation is excellent, remember to match your version! https://archive.istio.io/

You can always drill down to Envoy, so knowing their docs helps;
https://www.envoyproxy.io/docs

Thank you

Questions?

BTW We’re hiring! careers.unity.com


unity

Food

/sponsored by

centrica

CLOUD NATIVE
COMPUTING FOUNDATION

Prometheus 101

how to get started

/by Rasmus Steiniche

CEO neurospace

neuro space

An Official CLOUD NATIVE COMPUTING FOUNDATION Meetup Group

Prometheus 101

How to get started

$ whoami

Rasmus Steiniche (Linkedin: steiniche)

CEO at neurospace

Co-founder Cloud Native Nordics & Co-organizer Cloud Native Aarhus

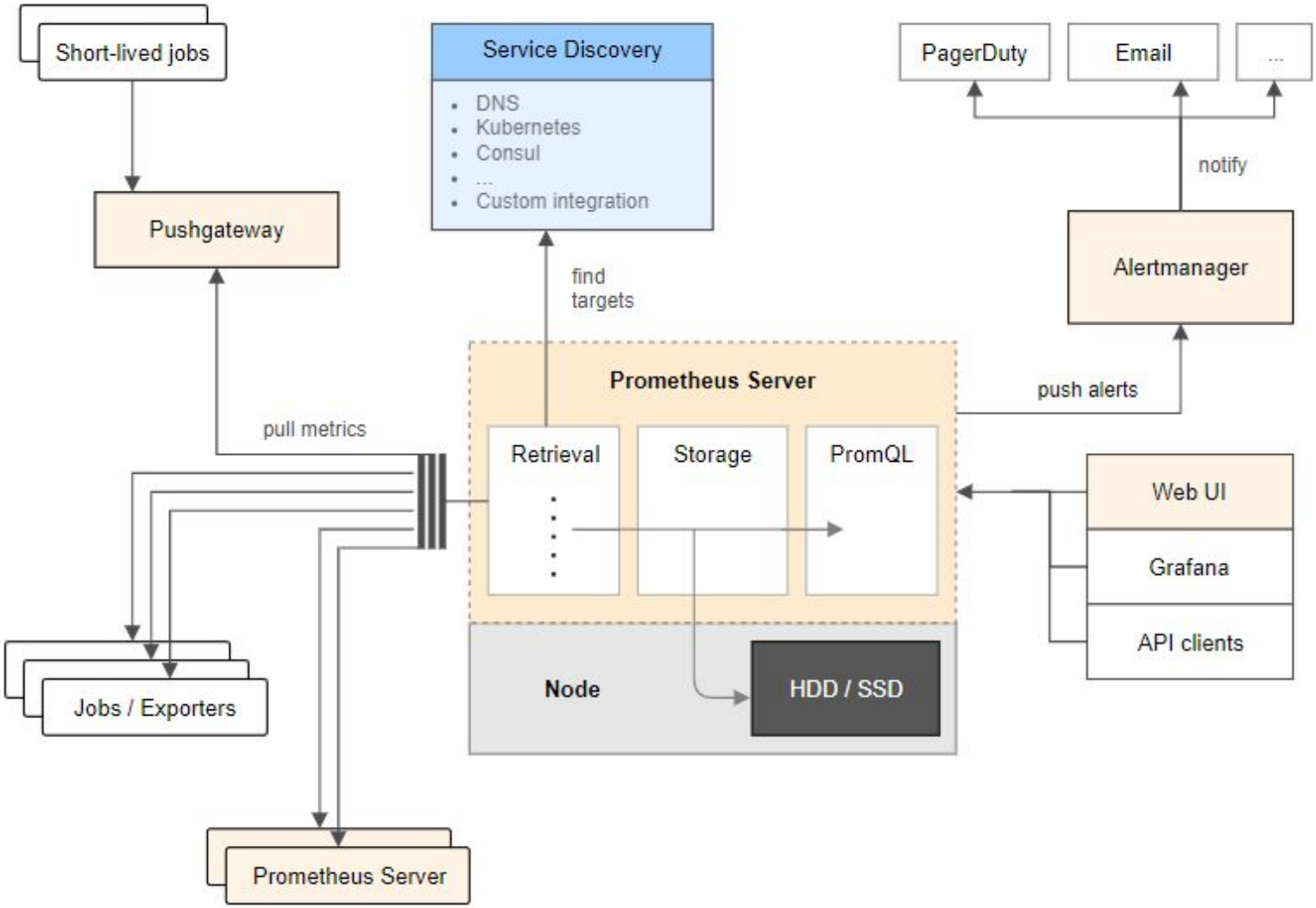
Believes machine learning (AI) will change the world

Started neurospace which help industrial companies utilize data and machine learning

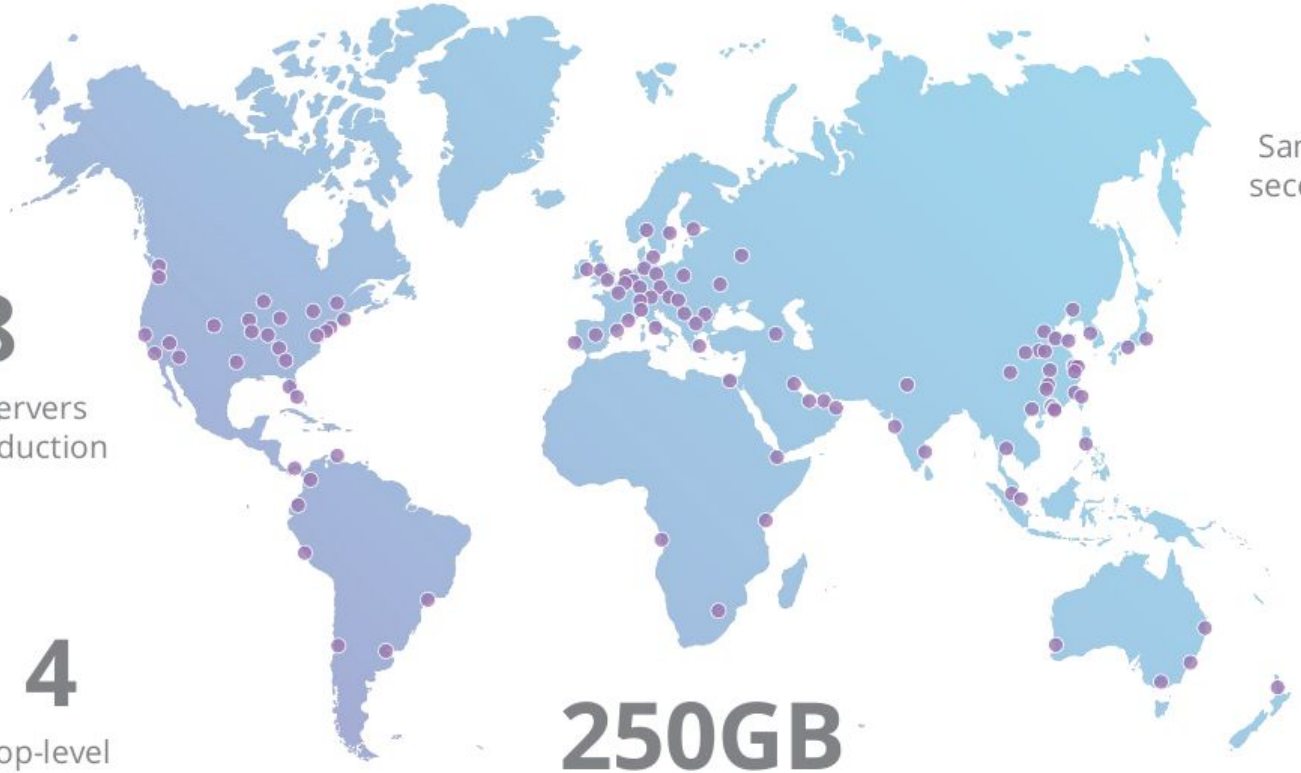
blog: neurospace.io/blog

CLOUD NATIVE
COMPUTING FOUNDATION

Prometheus internals


Short-lived jobs
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
PagerDuty
Email
...
notify
Pushgateway
find
targets
Alertmanager
Prometheus Server
push alerts
pull metrics
Retrieval
Storage
PromQL
Web UI
Grafana
API clients
Node
HDD / SSD
Jobs / Exporters
Prometheus Server

Cloudflare's Prometheus deployment



72k
Samples ingested per second max per server

188
Prometheus servers currently in Production

4.8M
Time-series max per server

4
Top-level Prometheus servers

250GB
Max size of data on disk

https://youtu.be/ypWwvz5t_LE

Time for a demo

https://github.com/Steiniche/prometheus

Thanks to vegasbrianc

Good monitoring - bad monitoring

Ask Rob Ewaschuk

Monitoring Distributed Systems (book) by Betsy Beyer, Rob Ewaschuk

My Philosophy on Alerting

https://docs.google.com/document/d/199PqyG3UsyXlwieHaqbGiWVa8eMWi8zzAn0YfcApr8Q/edit

CLOUD NATIVE COMPUTING FOUNDATION

Thanos

Highly available prometheus with long term storage.

Fixes many of the problems involved with federated prometheus.

https://thanos.io/

Cloud Native Nordics Prometheus and Grafana

The easiest way to get your feet wet!

https://prometheus.cloudnativenordics.com

https://grafana.cloudnativenordics.com

What is your questions?

Gitops

Operations by Pull Request

/by Kasper Nissen

CNCF Ambassador | SRE Lunar

LUNAR®

An Official CLOUD NATIVE COMPUTING FOUNDATION Meetup Group

LUNAR

GitOps

Operations by Pull Request

Kasper Nissen (@phennex)

$ whoami

LUNAR

Kasper Nissen (@phennex)

Cloud Architect / Site Reliability Engineer at Lunar

CNCF Ambassador

Certified Kubernetes Administrator

Cloud Native Aarhus (Cloud Native Copenhagen)

Cloud Native Nordics

Occasional speaker at Meetups, Conferences

Blog: kubecloud.io


kubernetes
Removing failing pod
slave2
slave3
slave4

Agenda

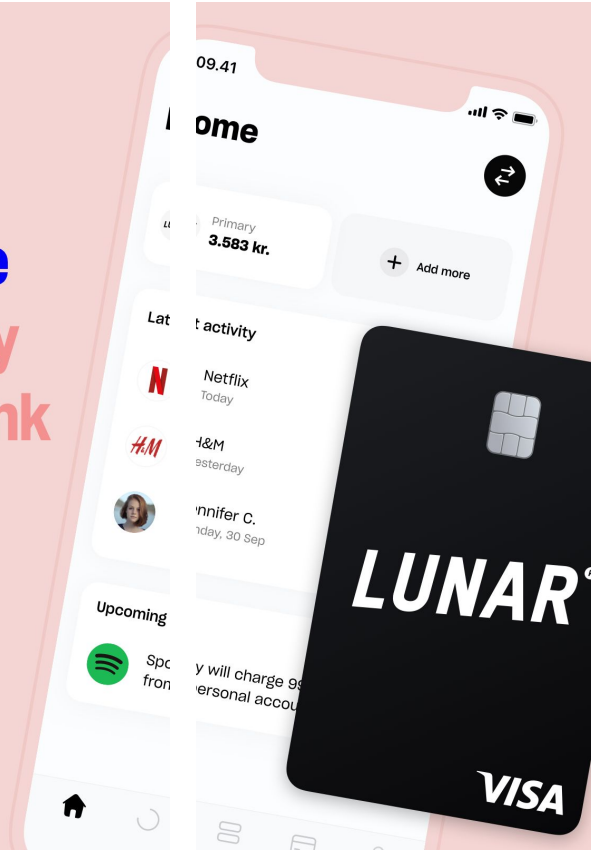
- What is Gitops?
- Flavours of Gitops
- GitOps at LUNAR
- Chaos Engineering
- GitOps at Cloud Native Nordics
- Progressive Delivery

About LUNAR

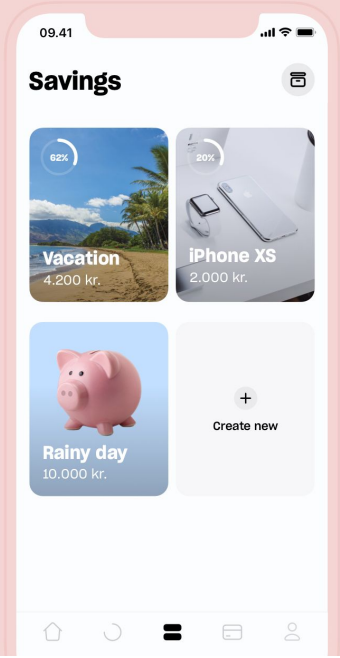
License to build a bank.

LUNAR

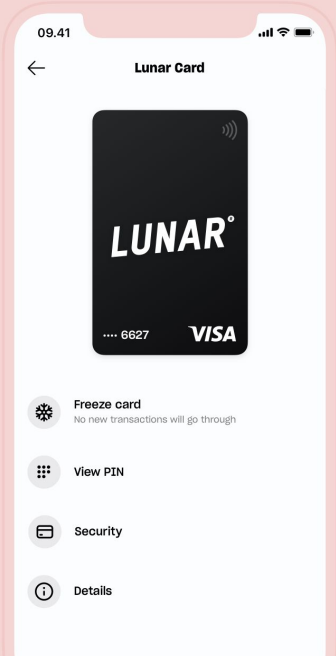
Change
the way you bank



Set goals for your savings



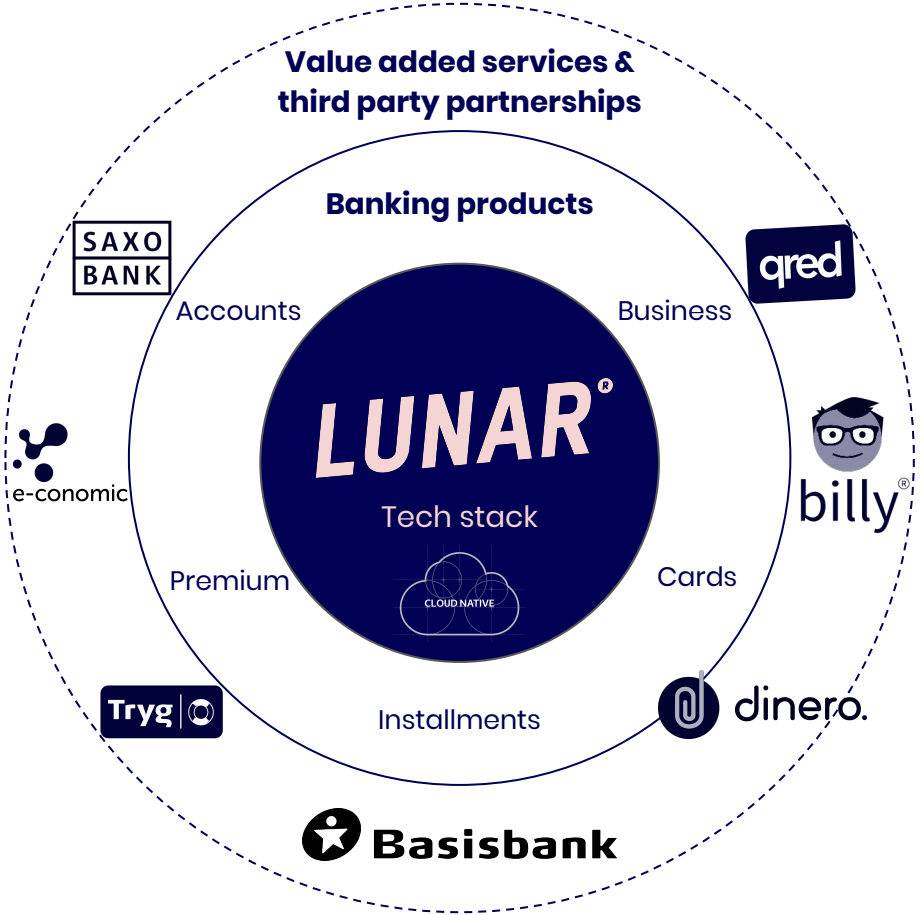
Manage your card easily



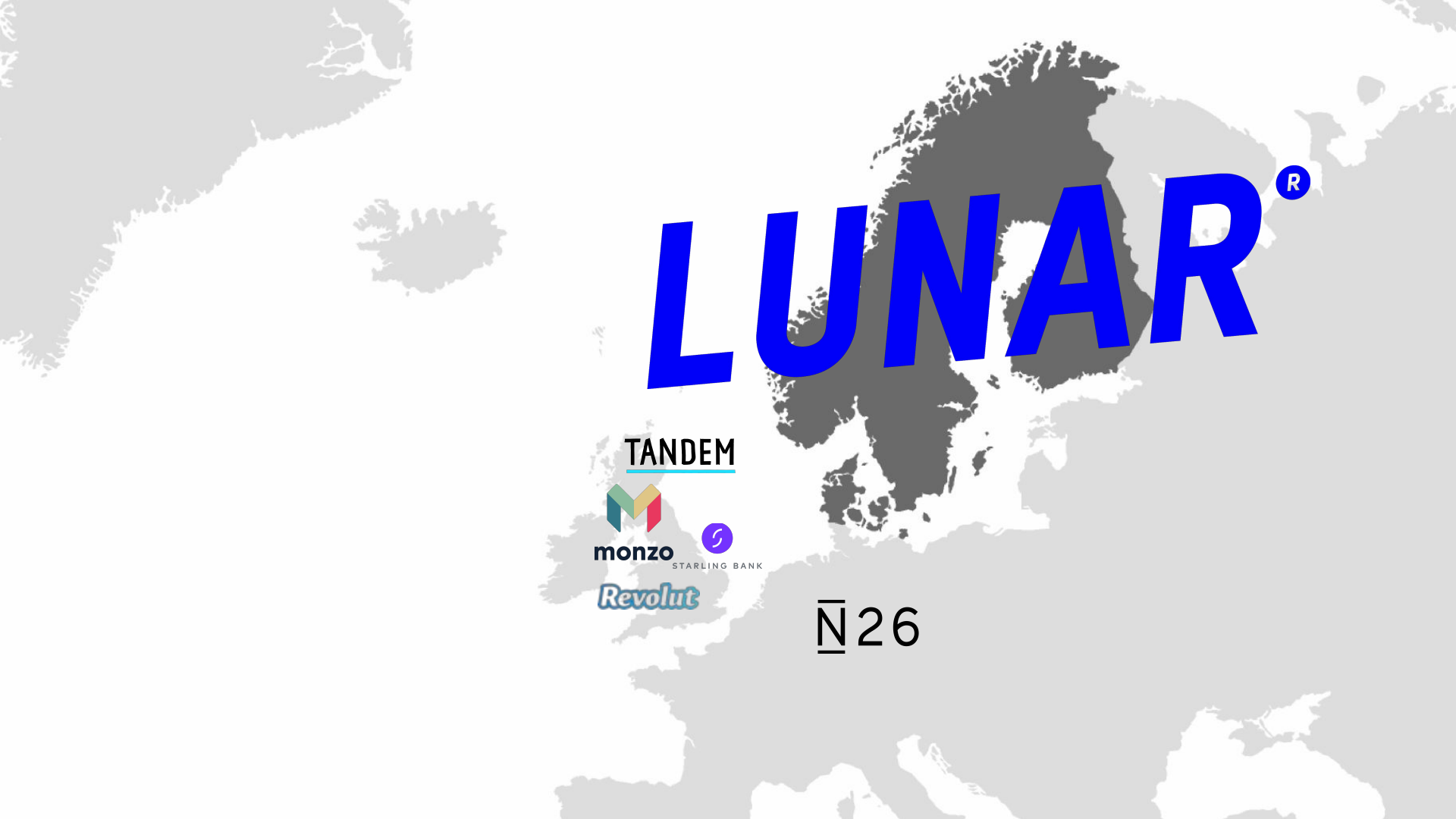
Building a Nordic bank is an enabler to our vision...

... on the back of the bank we are expanding to a Financial Super App

It's basically a single portal to a wide range of products and services from core banking to lifestyle, shopping, hospitality and transportation driven by user experiences



Rethinking the banking experience


LUNAR
TANDEM
monzo
STARLING BANK
Revolut
N26


Faster
Better
Stronger

GIT
OPS

What does reconciliation mean?

LUNAR

reconciliation

noun

UK /ˌrek.əˈnˌsɪl.iˈeɪ.ʃən/ **US** /ˌrek.əˈnˌsɪl.iˈeɪ.ʃən/

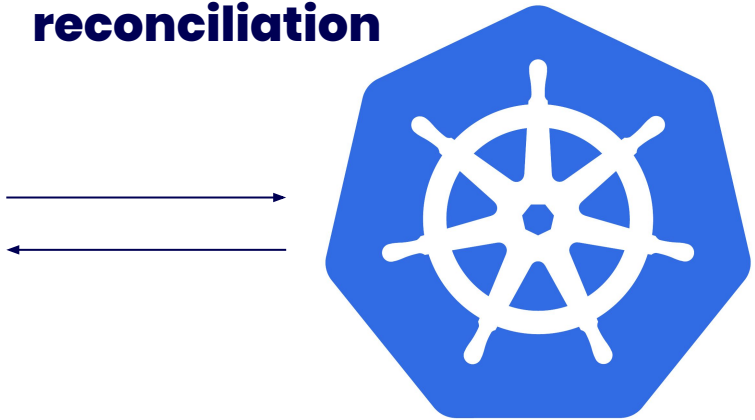
the process of making two people or groups of people friendly again after they have argued seriously or fought and kept apart from each other, or a situation in which this happens

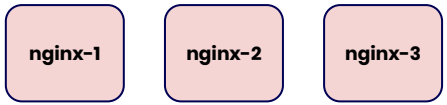
the process of making two opposite beliefs, ideas, or situations agree

Source: https://dictionary.cambridge.org/dictionary/english/reconciliation

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.17.5
        ports:
          - containerPort: 80
```

application reconciliation





reconciliation

LUNAR

What is GitOps

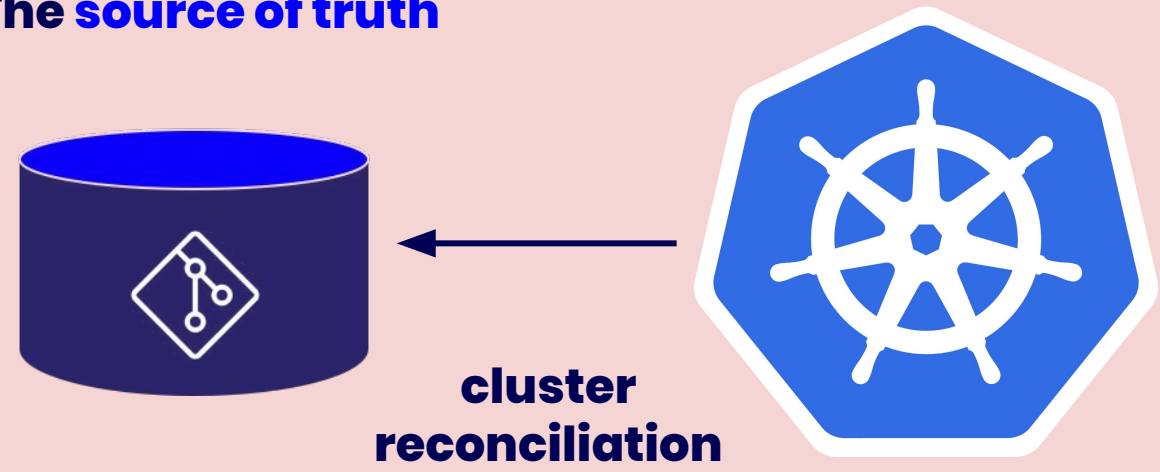
gitops

UK /gɪt/ɒp/ US /gɪt/ɑːp/

GitOps is a way to do Kubernetes cluster management and application delivery. It works by using Git as a single source of truth for declarative infrastructure and applications. With Git at the center of your delivery pipelines, developers can make pull requests to accelerate and simplify application deployments and operations tasks to Kubernetes.

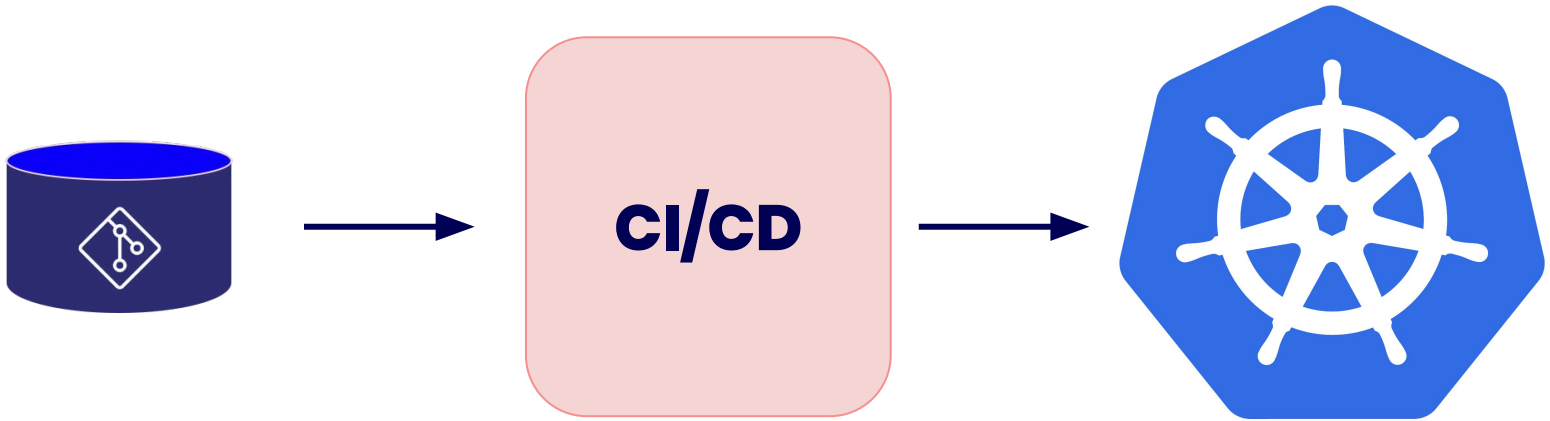
Source : https://www.weave.works/technologies/gitops/

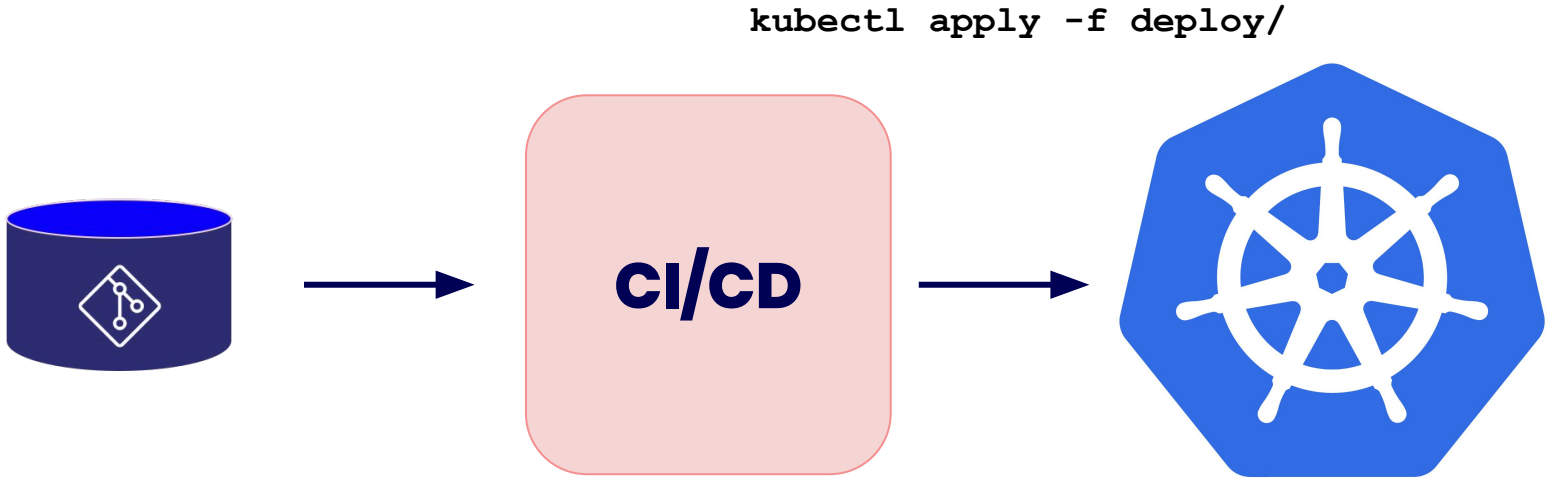
The source of truth



cluster
reconciliation

What's wrong with kubectl apply?


CI/CD


kubectl apply -f deploy/
CI/CD

What's wrong?

LUNAR

- CI/CD needs write access to your clusters
- How to track rollout failures?
- No audit trail of the kubernetes resources
- No single source of truth of the state in the cluster

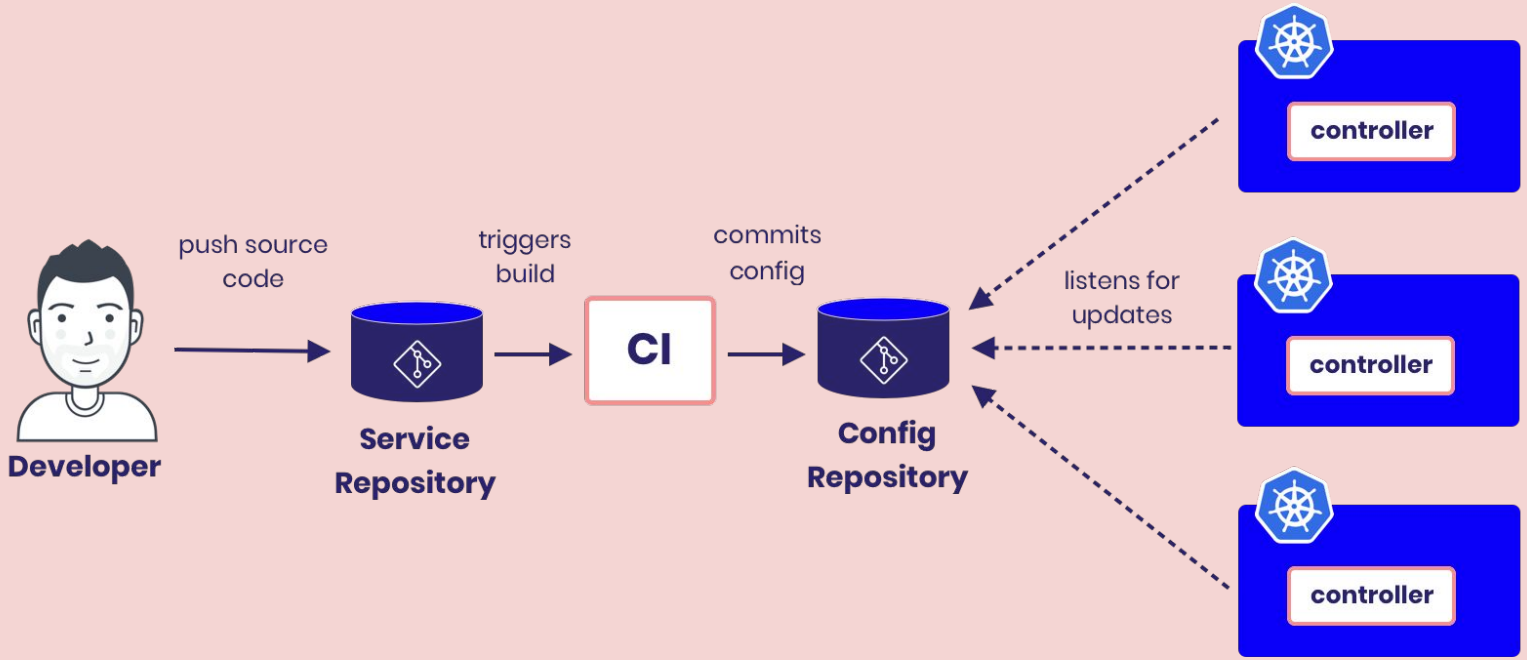
It's imperative. **It should be declarative.**

Flavours of Gitops

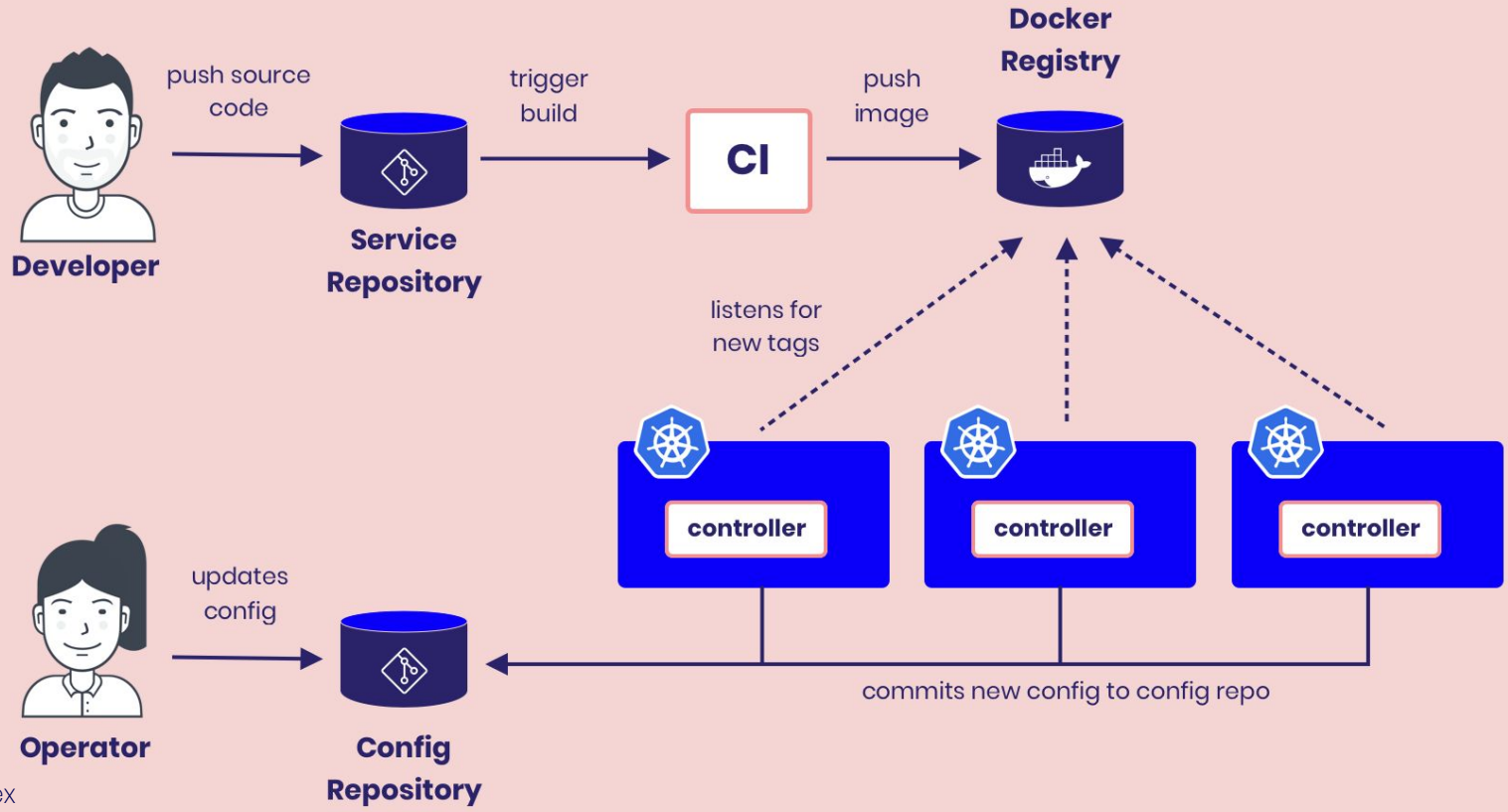
Flavours

- Decentralized One-way flow
- Decentralized Two-way flow
- Centralized flow

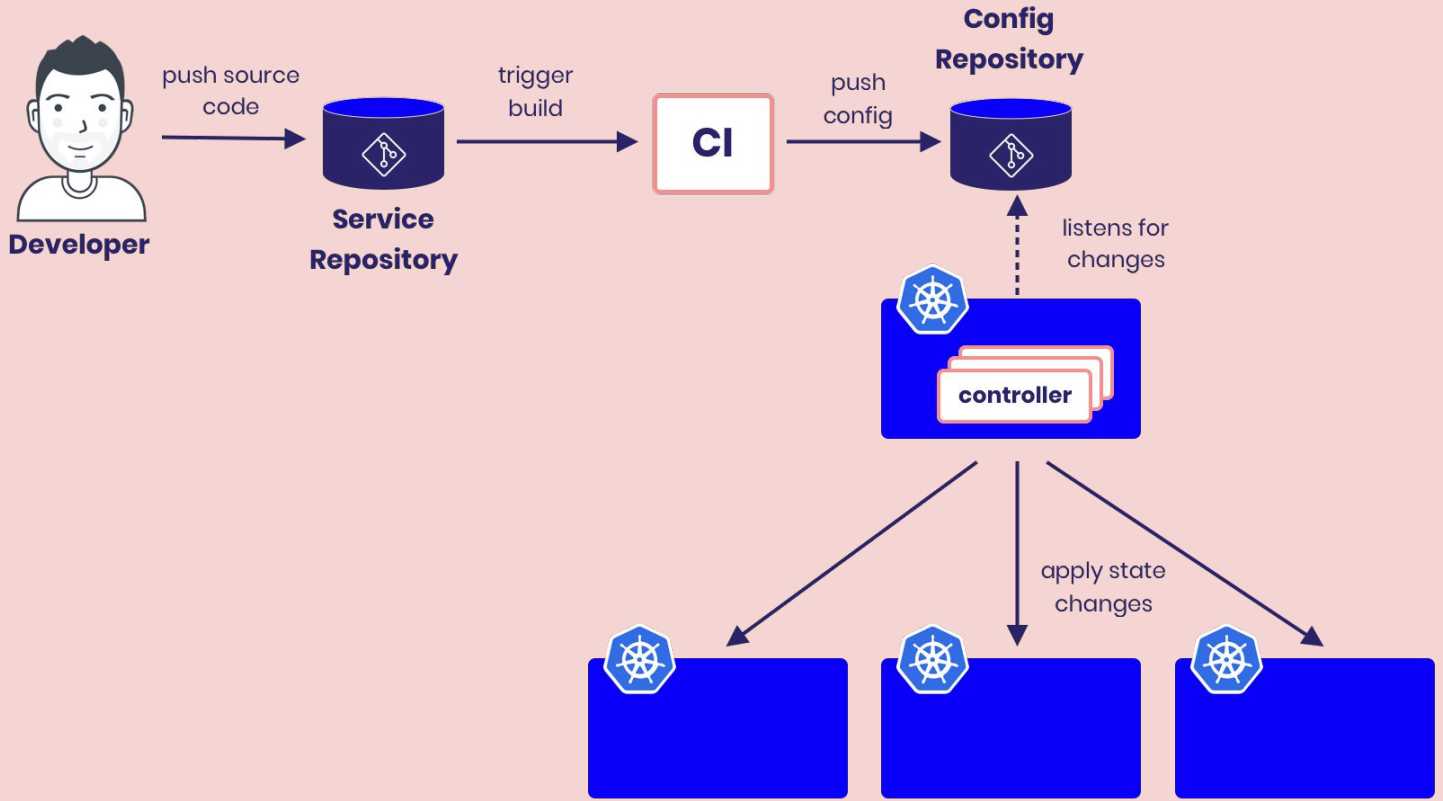
Decentralized one-way flow



Decentralized two-way flow

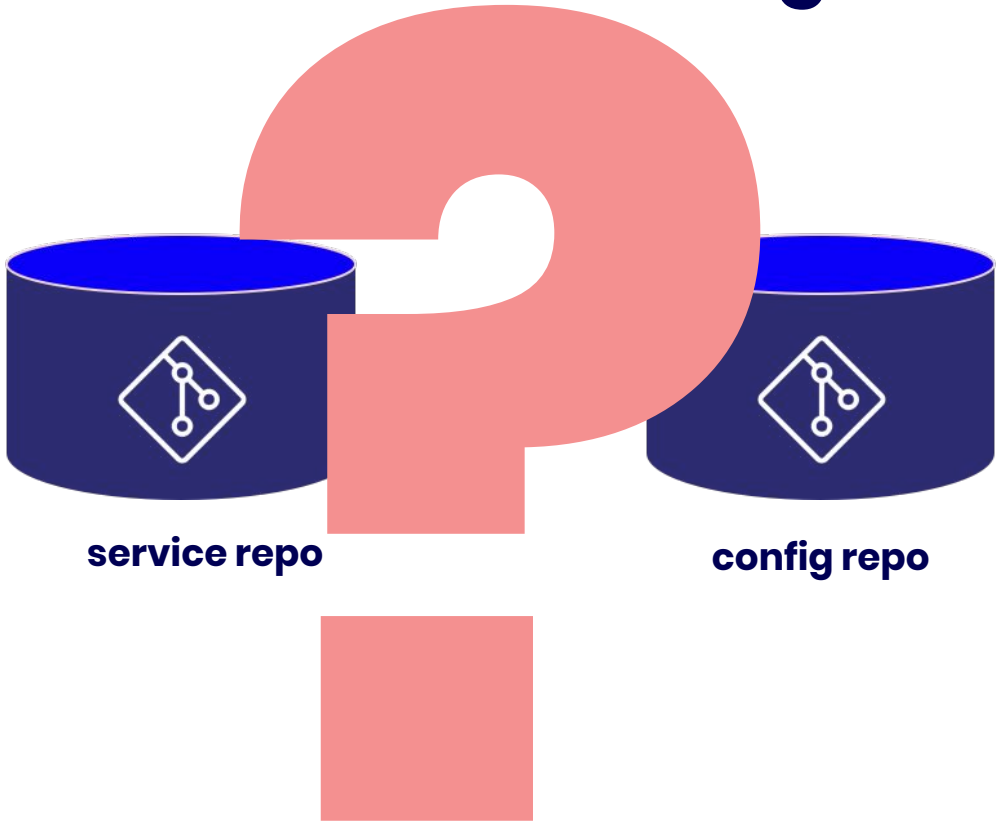


Centralized flow



Where should service config live?

Where should service config live?



Implementations/tooling

originally by weaveworks (now CNCF project)

originally by intuit

Argo CD

a declarative, GitOps continuous delivery tool for Kubernetes

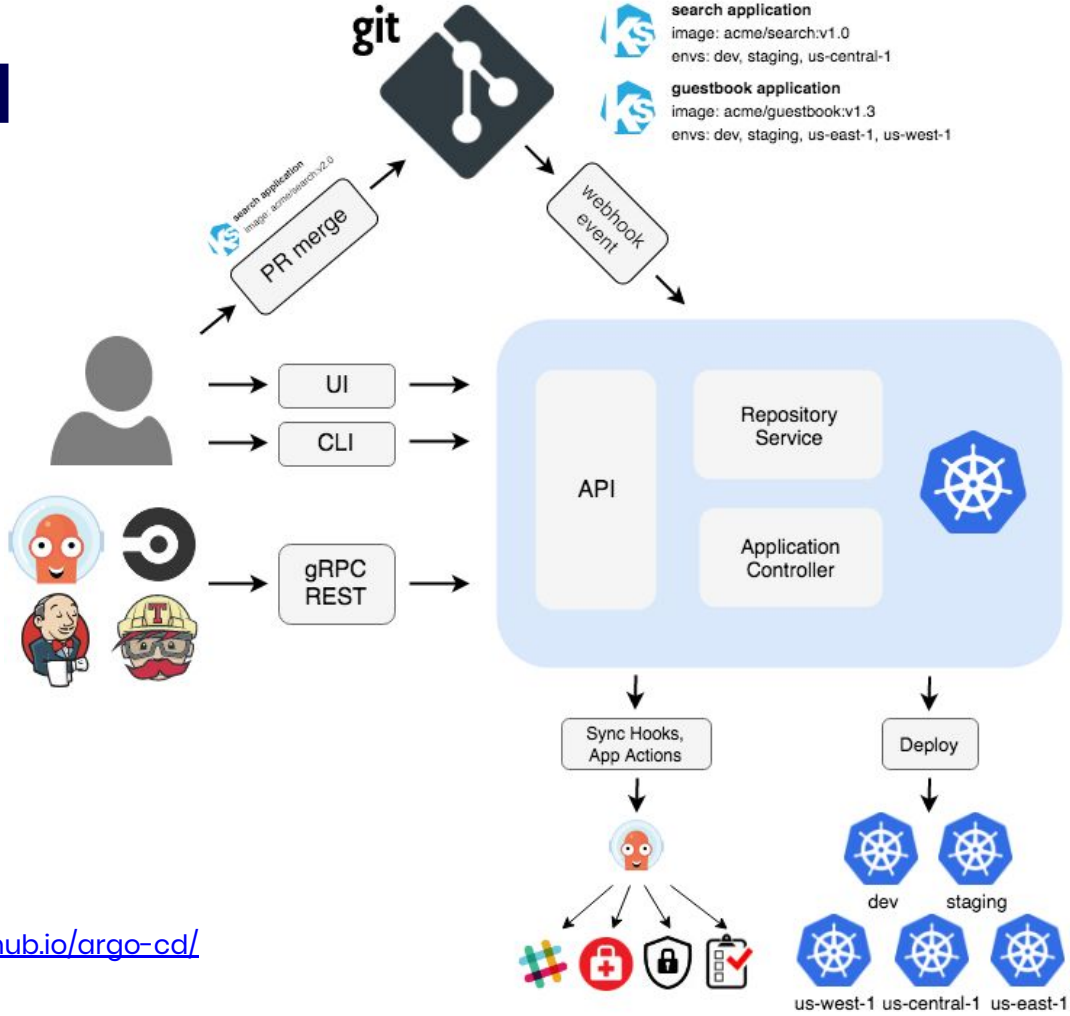
Features

- **Automated deployment** of applications to specified target environments
- Support for multiple config management/templating tools (Kustomize, Helm, Ksonnet, Jsonnet, plain-YAML)
- Ability to **manage** and **deploy** to **multiple clusters**
- SSO Integration (OIDC, OAuth2, LDAP, SAML 2.0, GitHub, GitLab, Microsoft, LinkedIn)
- Multi-tenancy and RBAC policies for authorization
- **Rollback/Roll**-anywhere to any application configuration committed in Git repository
- Health status analysis of application resources
- Automated configuration **drift detection** and **visualization**

Features

- **Automated or manual syncing** of applications to its **desired state**
- **Web UI** which provides real-time view of application activity
- **CLI** for automation and CI integration
- Webhook integration (GitHub, BitBucket, GitLab)
- Access tokens for automation
- PreSync, Sync, PostSync hooks to support complex application rollouts (e.g.blue/green & canary upgrades)
- Audit trails for application events and API calls
- Prometheus metrics
- Parameter overrides for overriding ksonnet/helm parameters in Git

argo-cd



Source: https://argoproj.github.io/argo-cd/

@phennex

Flux CD

The GitOps operator for Kubernetes

What is Flux CD?

"Flux is a tool that automatically **ensures that the state of your Kubernetes cluster matches the configuration you've supplied in Git.** It uses an operator in the cluster to trigger deployments *inside* Kubernetes, which means that you don't need a separate continuous delivery tool."

Source: www.fluxcd.io

Why Flux CD

Declarative

Describe the entire desired state of your system in Git. This includes apps, configuration, dashboards, monitoring, and everything else.

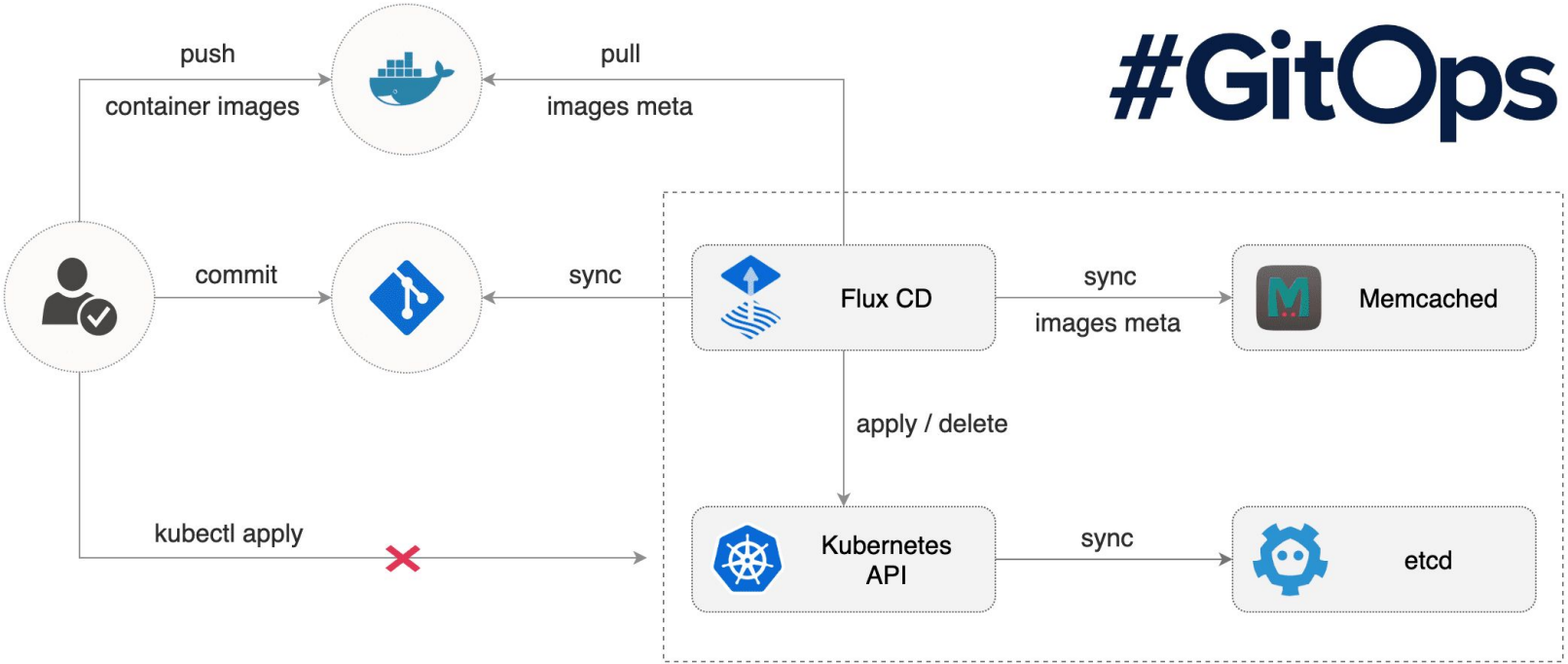
Automated

Use YAML to enforce conformance to the declared system. You don't need to run kubectl because all changes go through Git. Use diffing tools to detect divergence between observed and desired state and receive notifications.

Code, not containers

With Flux, everything is controlled through pull requests, which means no learning curve for new developers. Just use your standard PR process. Your Git history provides a sequence of transactions, allowing you to recover system state from any snapshot. Fix a production issue via pull request rather than making changes to the running system.

The Flux CD workflow



Source: https://fluxcd.io/

Argo Flux

The two biggest GitOps projects joining forces

Argo Flux

- Extract common functionality into **gitops-engine**
  - Access to Git repositories
  - Kubernetes resource cache
  - Manifest Generation
  - Resources reconciliation
  - Sync Planning

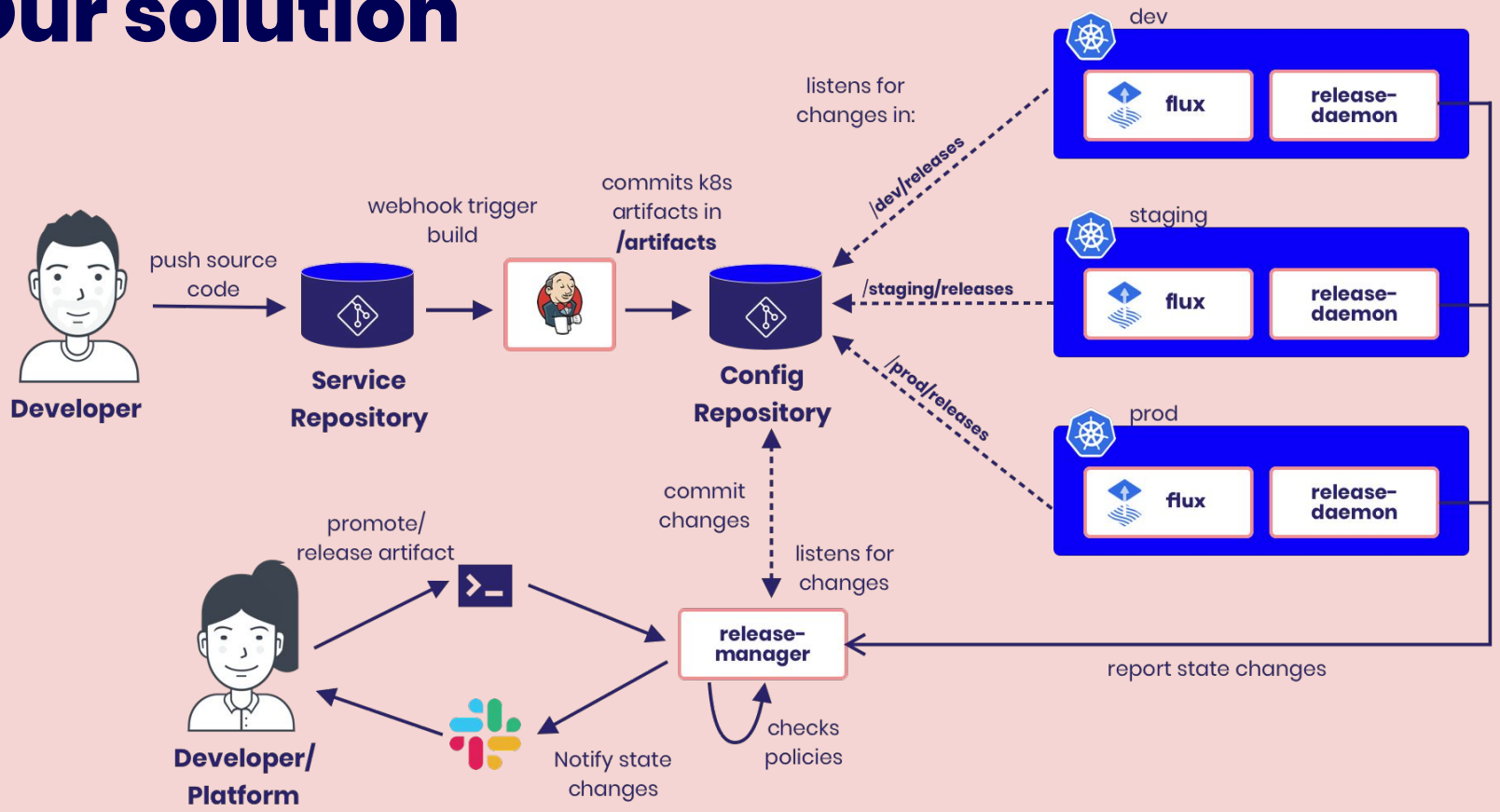
Source: https://github.com/argoproj/gitops-engine

GitOps at Lunar

Why GitOps?

- Audit trail of deployments
- Limit access to clusters
- Make Disaster Recovery an unpainful event

Our solution

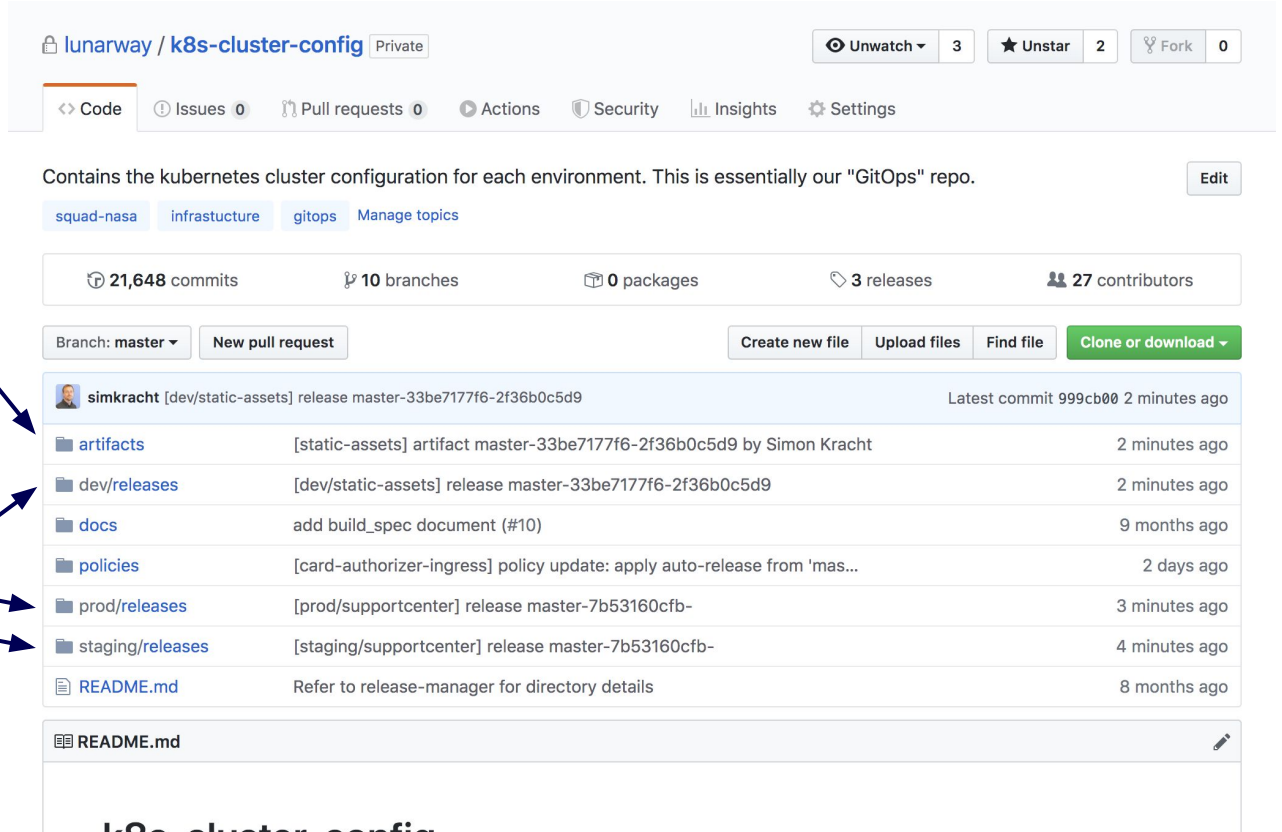


One or more config repos?

Our solution

Artifacts
pushed by Jenkins

Environments
controlled by
release-manager



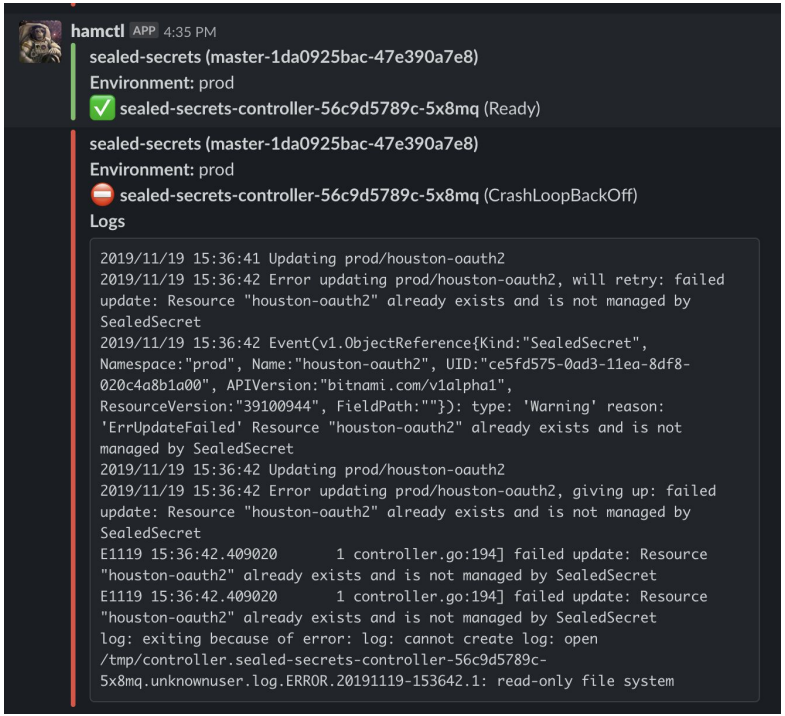
Dealing with multiple environments

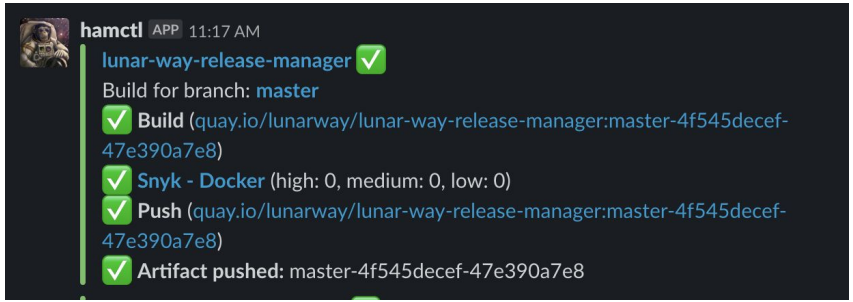
release-manager

- 4 components
  - **release-server**
    - The server itself, this is where the magic happens
  - **release-daemon**
    - Kubernetes controller listens for updates on resources and reports back to release-server
  - **hamctl**
    - CLI for developers to control releases
  - **artifact**
    - Tool for generating metadata object; artifact.json
    - Handles slack communication from CI as well

Source: https://github.com/lunarway/release-manager

release-daemon





Source: https://github.com/lunarway/release-manager

hamctl

```
# implicit order between envs: master -> dev -> staging -> prod
$ hamctl promote --env prod
```

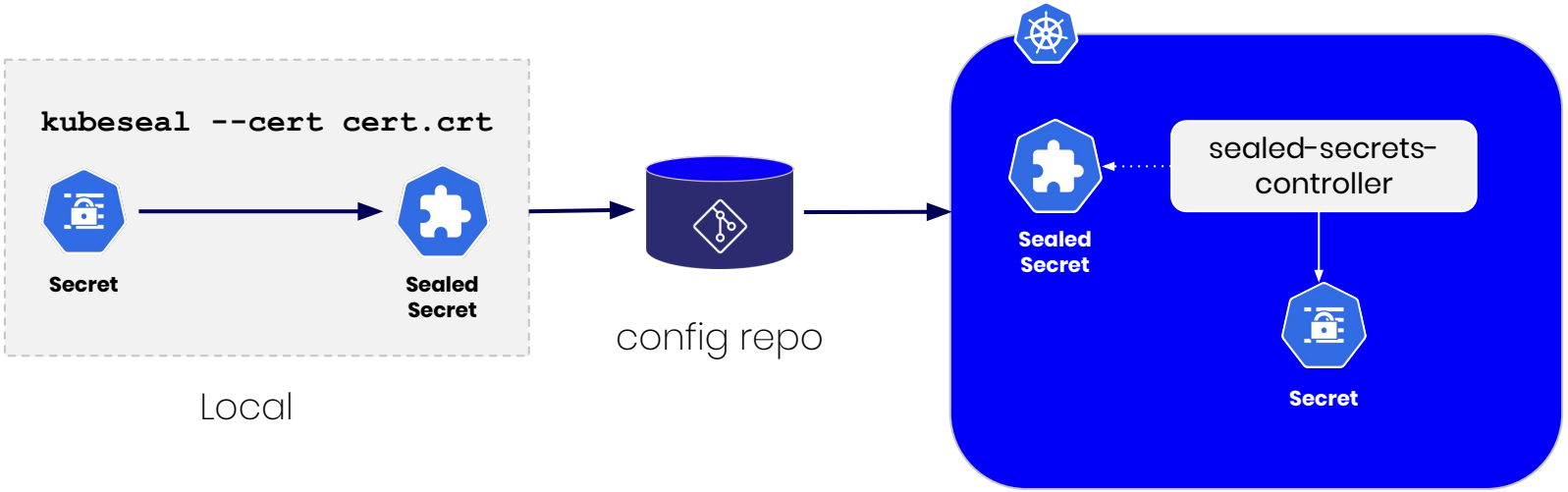
```
# choose which brand to deploy where
$ hamctl release --branch hotfix --env dev
```

```
# setup auto-release policy
$ hamctl policy apply auto-release --branch master
--env dev
```

Source: https://github.com/lunarway/release-manager

What about secrets?

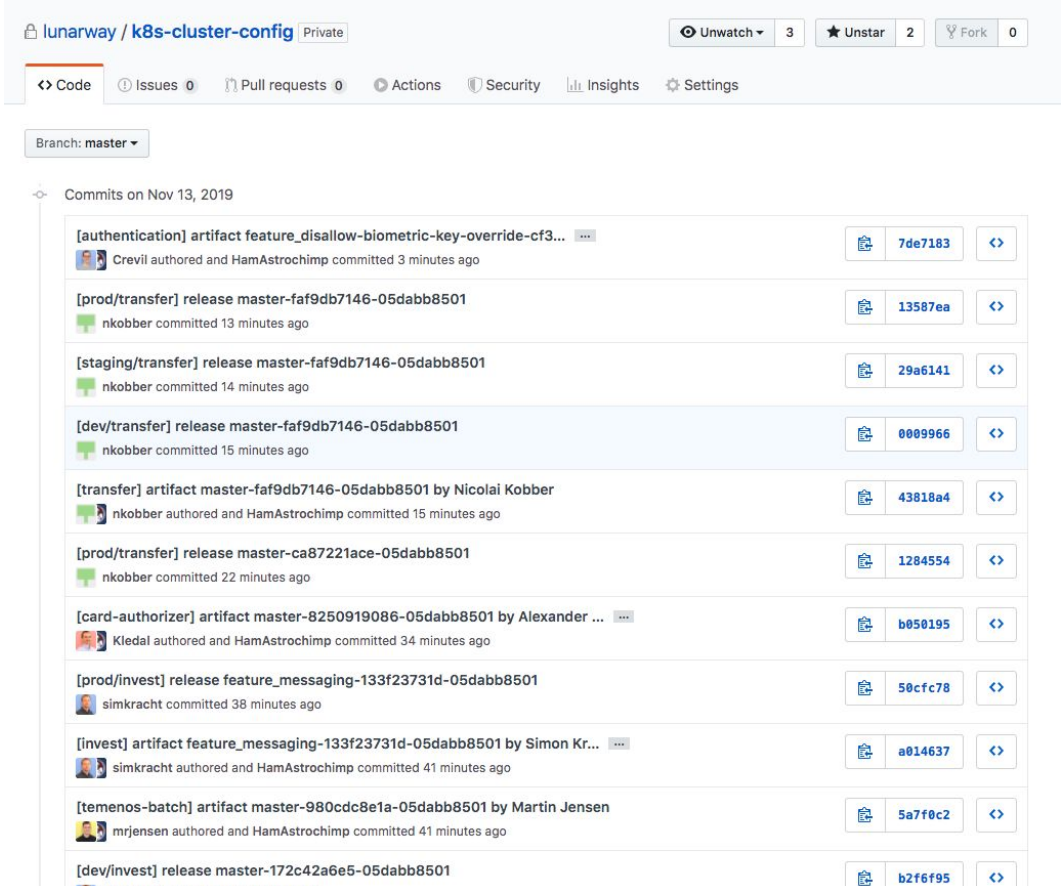
sealed-secrets



Source: https://github.com/bitnami-labs/sealed-secrets

Why is this useful?

Audit


lunarway / k8s-cluster-config Private
Unwatch 3
Unstar 2
Fork 0
Code
Issues 0
Pull requests 0
Actions
Security
Insights
Settings
Branch: master
Commits on Nov 13, 2019
[authentication] artifact feature_disallow-biometric-key-override-cf3...
Crevil authored and HamAstrochimp committed 3 minutes ago
7de7183
[prod/transfer] release master-faf9db7146-05dabb8501
nkobber committed 13 minutes ago
13587ea
[staging/transfer] release master-faf9db7146-05dabb8501
nkobber committed 14 minutes ago
29a6141
[dev/transfer] release master-faf9db7146-05dabb8501
nkobber committed 15 minutes ago
0009966
[transfer] artifact master-faf9db7146-05dabb8501 by Nicolai Kobber
nkobber authored and HamAstrochimp committed 15 minutes ago
43818a4
[prod/transfer] release master-ca87221ace-05dabb8501
nkobber committed 22 minutes ago
1284554
[card-authorizer] artifact master-8250919086-05dabb8501 by Alexander ...
Kledal authored and HamAstrochimp committed 34 minutes ago
b050195
[prod/invest] release feature_messaging-133f23731d-05dabb8501
simkracht committed 38 minutes ago
50cfc78
[invest] artifact feature_messaging-133f23731d-05dabb8501 by Simon Kr...
simkracht authored and HamAstrochimp committed 41 minutes ago
a014637
[temenos-batch] artifact master-980cdc8e1a-05dabb8501 by Martin Jensen
mrjensen authored and HamAstrochimp committed 41 minutes ago
5a7f0c2
[dev/invest] release master-172c42a6e5-05dabb8501
b2f6f95


Forklifting the entire platform
CASCADE
800
45'
JOE IPA
INDIA PALE ALE
10BARREL BREWING CO
HYSTER 50
@phennex


Disaster Recovery
Failover
Gamedays
@phennex

Gameday

How fast can you recover from a cluster failure?

How to restore the environment?

In which order do things need to be restored in?

dns

secret #1..X

volume #1

volume #2

...

volume #X

elb #1

elb #2

...

elb #X

external partner #1

external partner #2

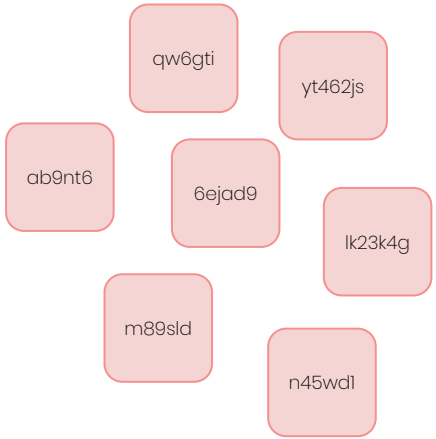
kubernetes configuration

etcd

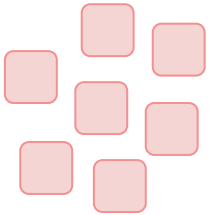

Clusters as
~~cattle~~ herds
instead of pets
@phrenex

Clusters as ~~cattle~~ herds instead of pets

pet instance

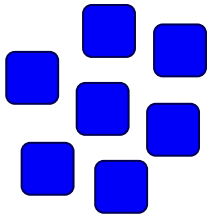


bob



ht998a

m8i7h3



lpq8qr

pet clusters


Chaos Engineering offers a dialogue with your system
@phennex

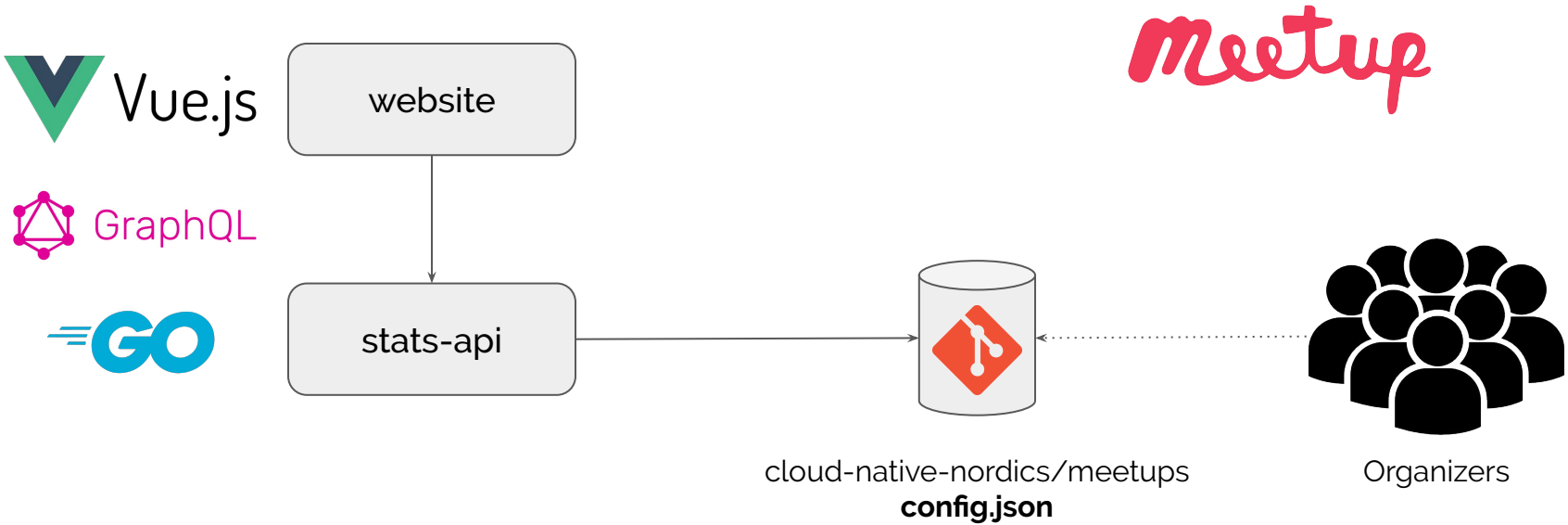
CLOUD NATIVE
NORDICS

GitOps at Cloud Native Nordics

@phennex

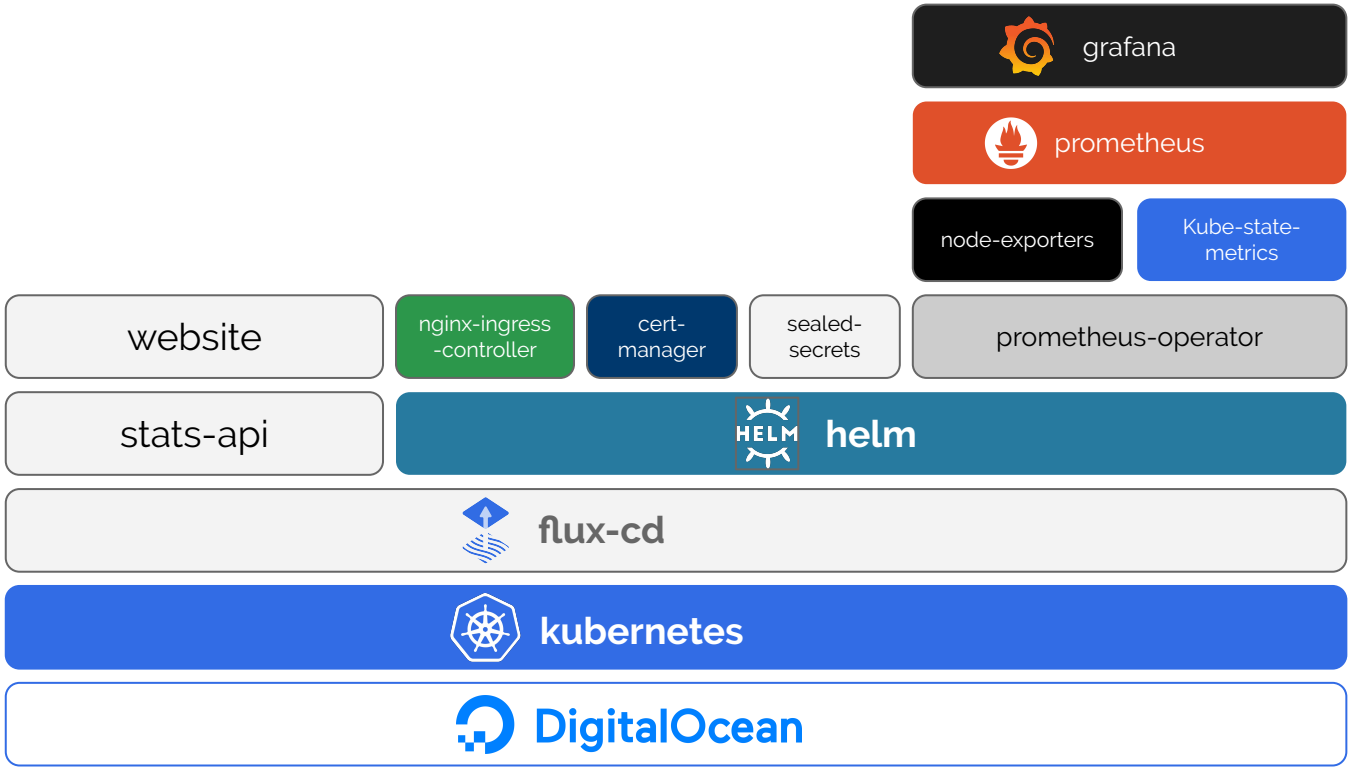
What? Don't they just run a static website?

Decentralized two-way flow

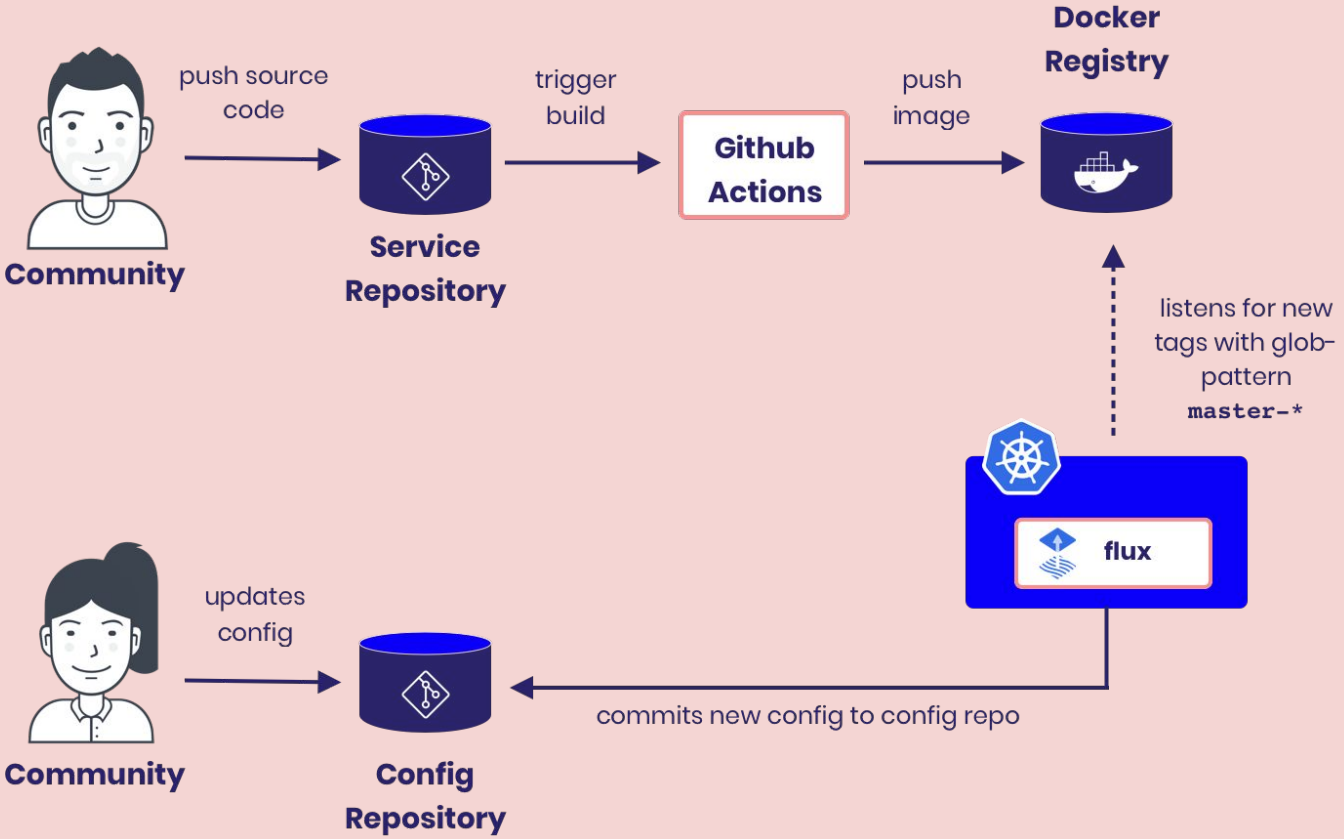


Decentralized two-way flow

https://github.com/cloud-native-nordics/k8s-config-repo

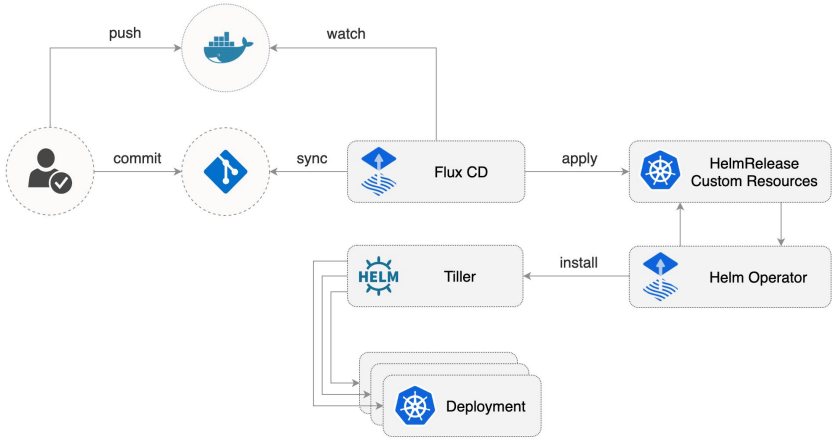


Our solution



HelmRelease

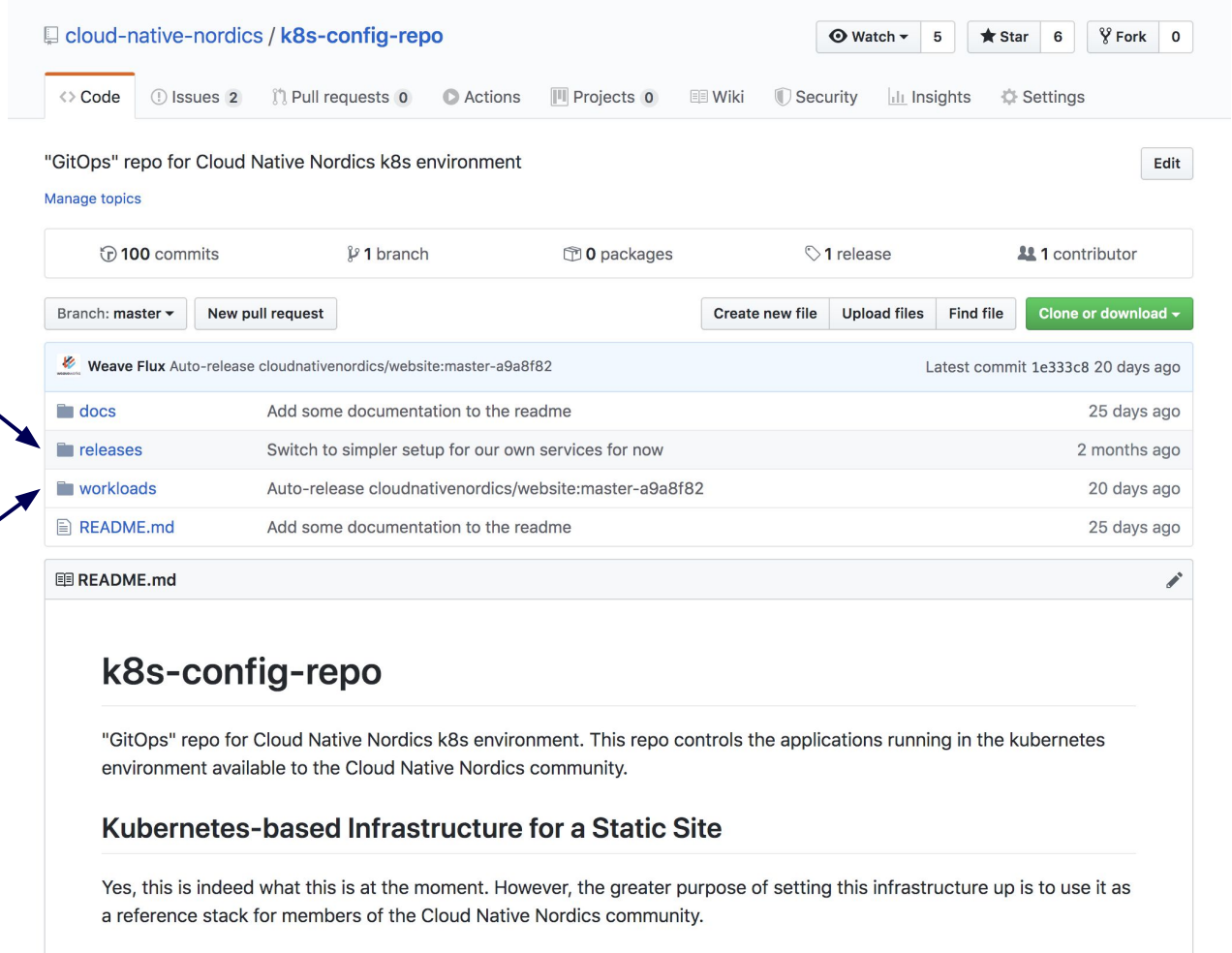
Custom Resource + Operator to support helm




```yaml
apiVersion: flux.weave.works/v1beta1
kind: HelmRelease
metadata:
  name: nginx-ingress
  namespace: default
  annotations:
    flux.weave.works/automated: "false"
spec:
  releaseName: nginx-ingress
  chart:
    repository: https://kubernetes-charts...googleapis.com/
    name: nginx-ingress
    version: 1.22.1
  values:
    controller:
      publishService:
        enabled: true
```

HelmRelease
managed using Helm

Regular YAML
kubernetes deployments, etc



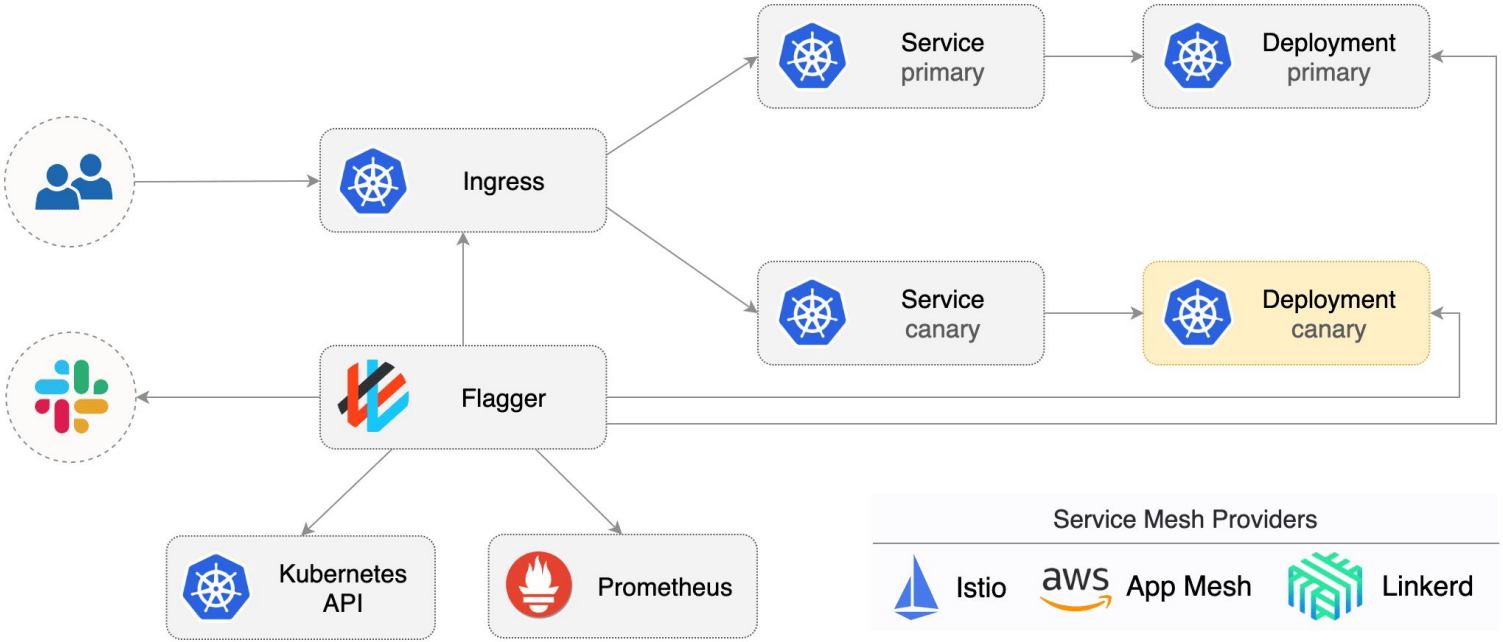
Progressive Delivery

Progressive Delivery

Progressive delivery is the process of pushing changes to a product iteratively, first to a small audience and then to increasingly larger audiences to maintain quality control (QC). The goal of progressive delivery is to improve delivery times for new product features and mitigate risk by controlling who is able to see them.

Source: https://searchitoperations.techtarget.com/definition/progressive-delivery

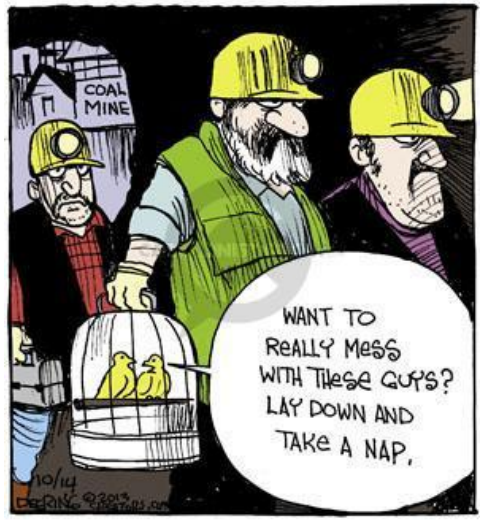
Progressive Delivery



Source: https://github.com/weaveworks/flagger

Canary

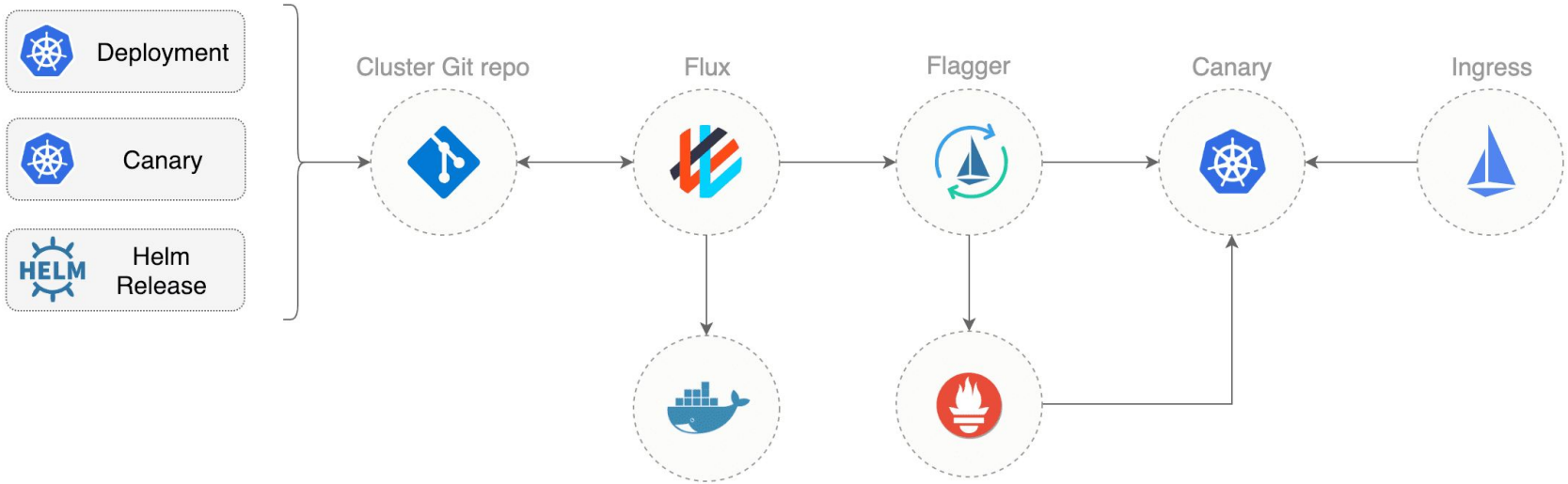
Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.



©Creators Syndicate.

```yaml
apiVersion: flagger.app/v1alpha3
kind: Canary
metadata:
  name: podinfo
spec:
  targetRef:
    kind: Deployment
    name: podinfo
  progressDeadlineSeconds: 60
  service:
    targetPort: 9898
  canaryAnalysis:
    interval: 1m
    threshold: 10
    maxWeight: 50
    stepWeight: 5
    metrics:
    - name: request-success-rate
      threshold: 99
      interval: 1m
    - name: request-duration
      threshold: 500
      interval: 30s
    webhooks:
      - name: load-test
        metadata:
            cmd: "hey -z 1m -q 10 -c 2
  http://podinfo.test:9898/"
```

Progressive Delivery



Source: https://github.com/weaveworks/flagger

LUNAR

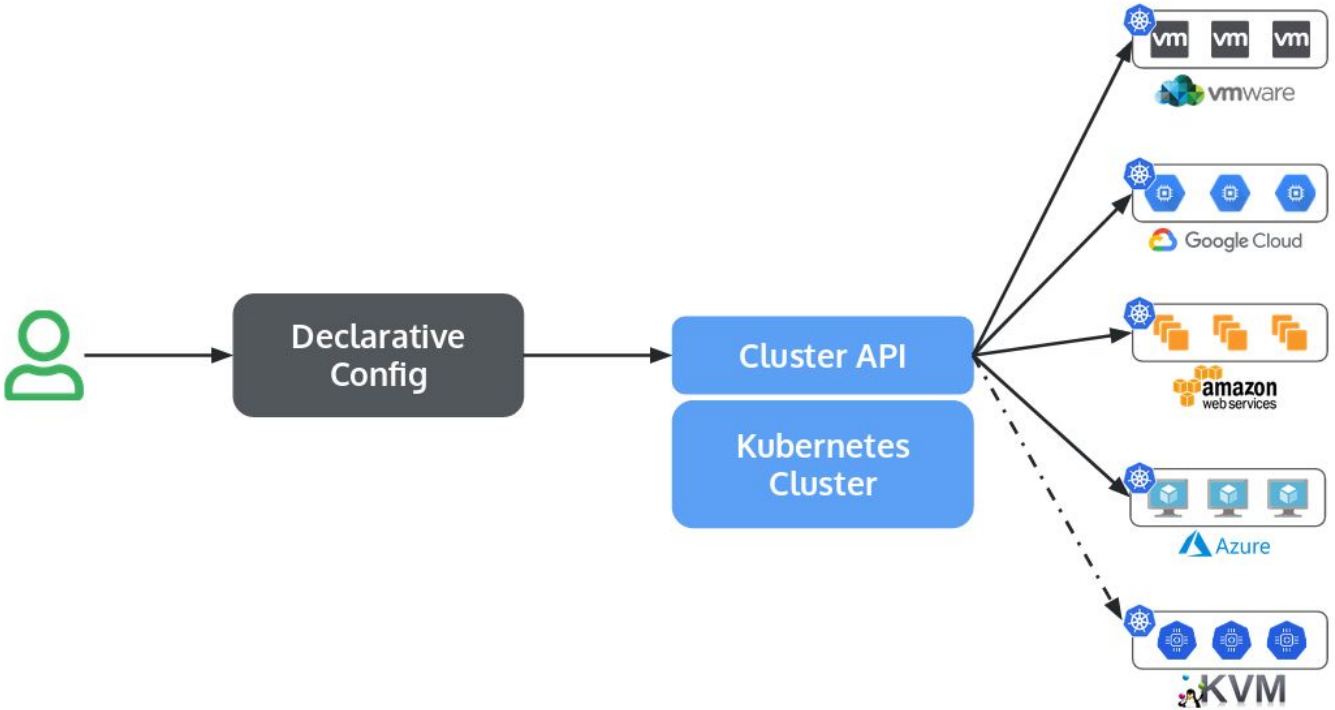
Everything in Git

Everything in Git

- CRDs
  - postgresql-controller
    - manage users, databases, hosts using Kubernetes objects
    - store them in git!
  - AWS Resources
- Infrastructure
  - cluster-api

```yaml
apiVersion: lunar.bank/v1
kind: PostgreSQLUser
metadata:
  name: kni
spec:
  name: kni
  read:
    - host:
        value: some.host.com
      reason: "I am a developer"
  write:
    - host:
        valueFrom:
          configMapKeyRef:
            name: database
            key: db.host
      database:
        value: user
      schema:
        value: user
      reason: "Related to support ticket LW-1234"
      start: 2019-09-16T10:00:00Z
      end: 2019-09-16T14:00:00Z
```

cluster-api



Source: https://itnext.io/kubernetes-cluster-creation-on-baremetal-host-using-cluster-api-1c2373230a17

cluster-api

Cluster

Machine

Machine Set

Machine Deployment

Machine Class

Pod

Replica Set

Deployment

Storage Class

Source: https://itnext.io/kubernetes-cluster-creation-on-baremetal-host-using-cluster-api-1c2373230a17

cluster-api

Use kubernetes to manage kubernetes.

Again....

Using Git!

```yaml
apiVersion: "cluster.k8s.io/v1alpha1"
kind: MachineDeployment
metadata:
  name: nodes
  namespace: kube-system
spec:
  replicas: 5
  selector:
    matchLabels:
      foo: bar
  template:
    metadata:
      labels:
        foo: bar
    spec:
      providerSpec:
        value:
          cloudProvider: "aws"
          cloudProviderSpec:
            region: "eu-central-1"
            availabilityZone: "eu-central-1a"
            vpcId: "vpc-819f62e9"
            subnetId: "subnet-2bff4f43"
            instanceType: "t2.micro"
            instanceProfile: "kubernetes-v1"
            diskSize: 50
            ..
          operatingSystem: "coreos"
          operatingSystemSpec:
            disableAutoUpdate: true
```

Things change...
CI/CD should too

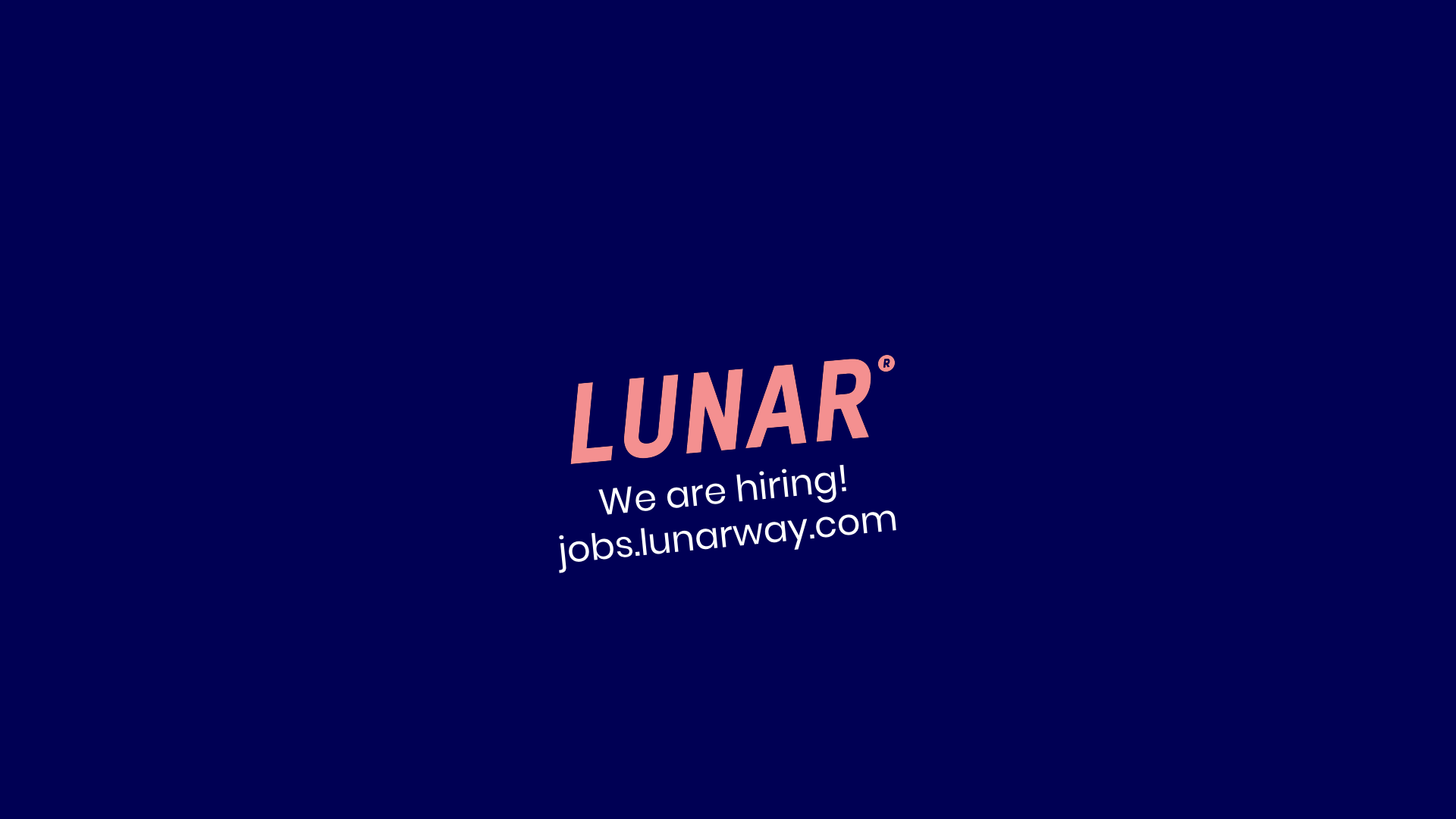
LUNAR

Things change…
Banks should too

Questions?

Contact:

Twitter: @phennex
Github: kaspernissen
E-mail: kni@lunarway.com


LUNAR®
We are hiring!
jobs.lunarway.com

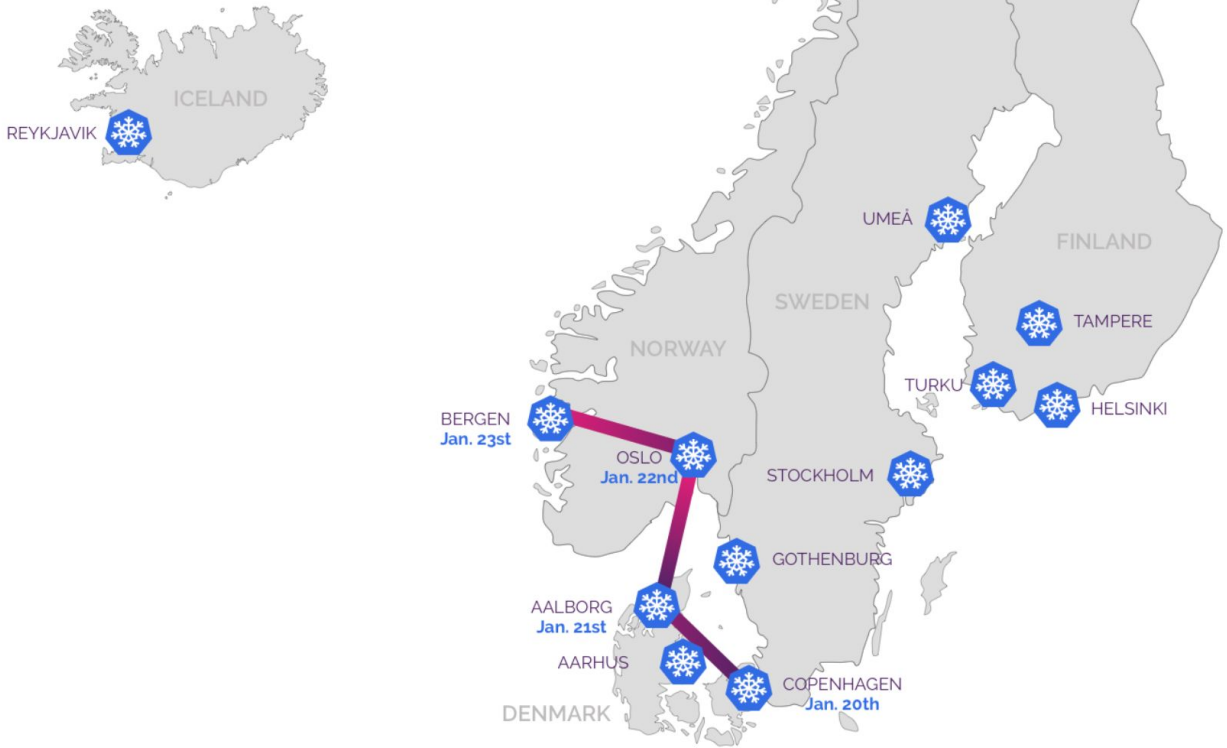
A big thank you to this evening's host

centrica

CLOUD NATIVE NORDICS TOUR

JANUARY 2020 (20.-23.)

Nick Jones - KUDO
Hans Kristian Flaatten, EVRY



CLOUD NATIVE
NORDICS

CloudNative **Aalborg** presents

KUDO - Kubernetes Operators, the easy way

by Nick Jones - D2IQ

D2IQ

Building a Bank from Scratch in the Cloud on Kubernetes

by Hans Kristian Flaatten - EVRY

EVRY

Hosted by

netic

A TRIFORK COMPANY

An Official

CLOUD NATIVE COMPUTING FOUNDATION

Meetup Group

Networking

/by You

Cloudnatives in Aalborg

Happy Holidays

An Official
CLOUD NATIVE
COMPUTING FOUNDATION
Meetup Group

CloudNative **Aalborg**

#CloudNativeAalborg
#CloudNativeNordics

Join Slack  @ https://www.cloudnativenordics.com/

An Official
CLOUD NATIVE
COMPUTING FOUNDATION
Meetup Group