Using Kubernetes to Change Legacy Systems and Processes in the Public Sector

Audun Fauchald Strand,
Norwegian Labour & Welfare Administration

✉ audun.fauchald.strand@nav.no
audunstrand@gmail.com

audunstrand

Stuck on premise

Too many test environments

Too coarse grained access Control for developers

Network Zones

Overview of dependencies

Monitoring

Cumbersome to create new applications

Low Resource Utilization

Nightly batch jobs

Agenda

NAV

Problems and solutions

NAIS.io

Conclusions

Audun Fauchald Strand

Java developer

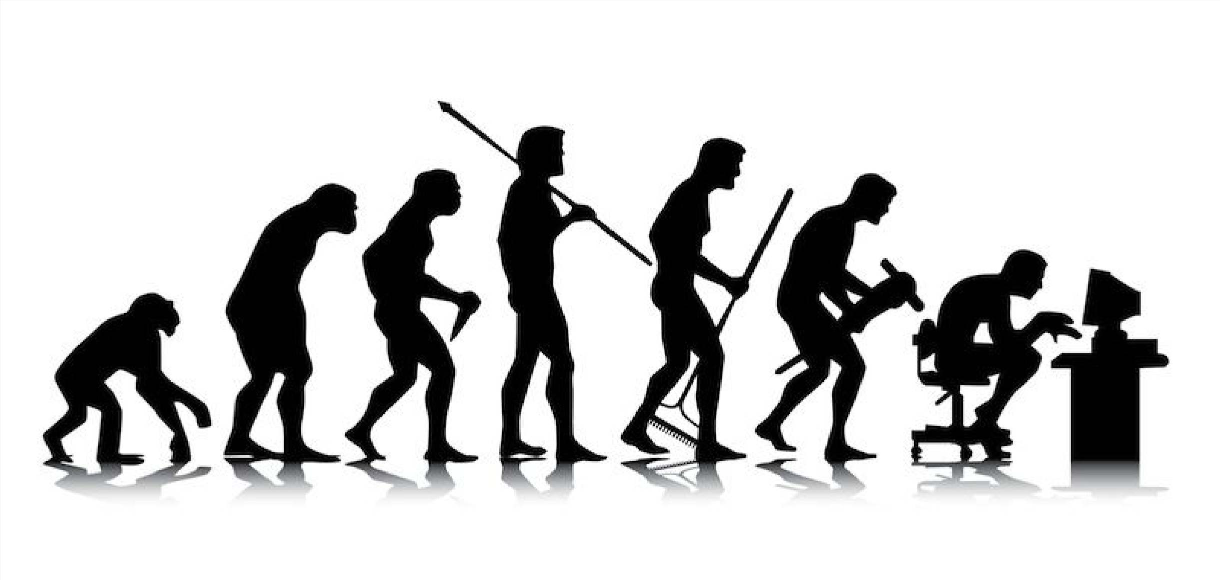
DevOps awakening

Domain Driven Design

Kubernetes

Kafka

Development speed without sacrificing resilience



Kubecon Berlin
April 2017

Continuous delivery while migrating to Kubernetes

@ FINN.no

FINN

Continuous delivery while migrating to Kubernetes

Audun Fauchald Strand

Øyvind Ingebrigtsen Øvergaard

@audunstrand

@oyvindio

NAV - Norwegian Labour & Welfare Administration

https://en.wikipedia.org/wiki/Norwegian_Labour_and_Welfare_Administration

Norwegian Welfare Administration

16000 employees in offices all over Norway

600 in IT

⅓ of the federal budget paid out through NAV

Unemployment benefit

Pensions

Sickness benefit

NAV Technology History

First system in 1967, database still in use

Mainframe

Java in Oracle Database

IBM WebSphere

VMWare

Jboss

Jetty

Kubernetes

NAV pre-2017

Private Cloud Vmware

Self service with web apps

3 month release cycle, 4 weeks testperiode

Separate department doing application operations

Devs have no access to production environment

Culture pre-2017

Developers was mostly external consultants

Operations had all the power

Plan - build - run

Architects

Testers

Release managers

The Big Change

New Boss

New Direction

Hire our own developers

Continuous Delivery

You build it, you run it

NAIS.io

Nais.io

Internal platform

Built to ease migration from old platform to new

Open Source

Problems and Solutions

Problems

Stuck on premise

Too many test environments

Access Control for developers

Network Zones

Overview of dependencies

Monitoring

Cumbersome to create new applications

Low Resource Utilization

Nightly batch jobs

Stuck on premise

No public cloud data centers in norway

Sensitive data crossing borders is an security issue

Horror stories from governmental organizations in Norway and Sweden

On-premise datacenter operations are inefficient

Long time to migrate

Kubernetes

Get value early from you road to a public cloud

Mirrors most of the offerings in public cloud

- Functions as a Service
- Databases
- Storage

Cloud Native

- Monitoring
- Service Mesh

Too many test environments

More than 20 distinct test environments

Different versions of applications running in different environments

Developer environment, customer testing environment, staging environment

Some environments differ only in data

Namespaces

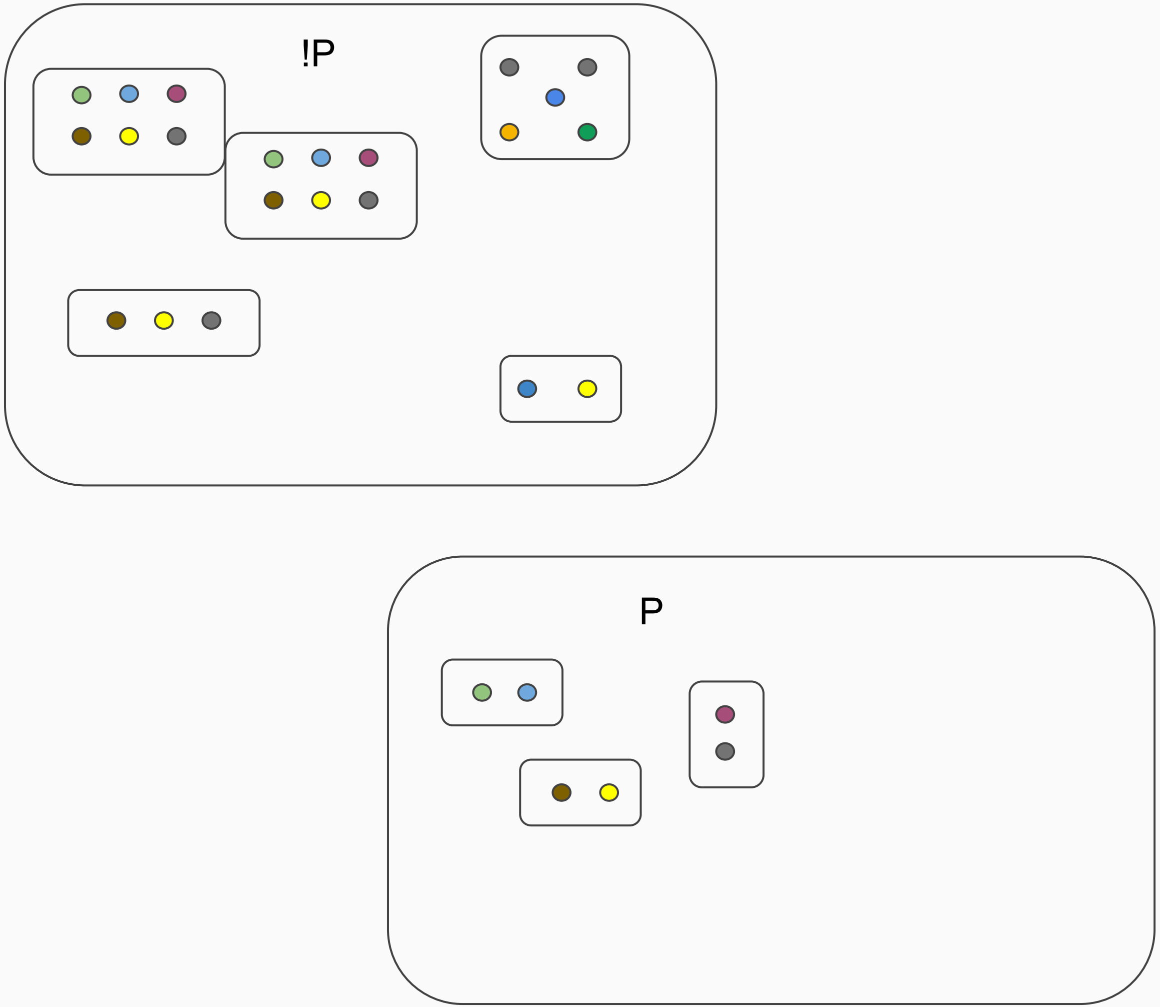
Two kubernetes clusters

- P - Production
- !P - test/staging/development

Automatic provisioning of namespaces in !P

Be difficult! The fewer environments the better

Test environments exists because of 3 month release cycle



Access Control

Before, Ops had access to everything, Devs had no access in production.

No audit logging of what happened, and no personal users

Ops didn't want to give prod access to devs

OpenId Connect
RBAC

Azure AD

OpenId Connect

- Personal users
- Audit logging of what each user does

RBAC

- Namespaces in production matching teams
- Rolebindings giving devs access to only their stuff
- Namespaces in production used for access control

Network Zones

Multitude of network zones

Swiss Cheese firewall between them

Slow and manual routines for opening firewall

Illusion of control

Network Policies

Zero Trust Policy

Network Policies

Overview of dependencies

Multitude of projects trying to create order and give oversight

Huge architecture department drawing in archimate

None of the models are based on runtime, either design-time or development time

Does anyone need this?

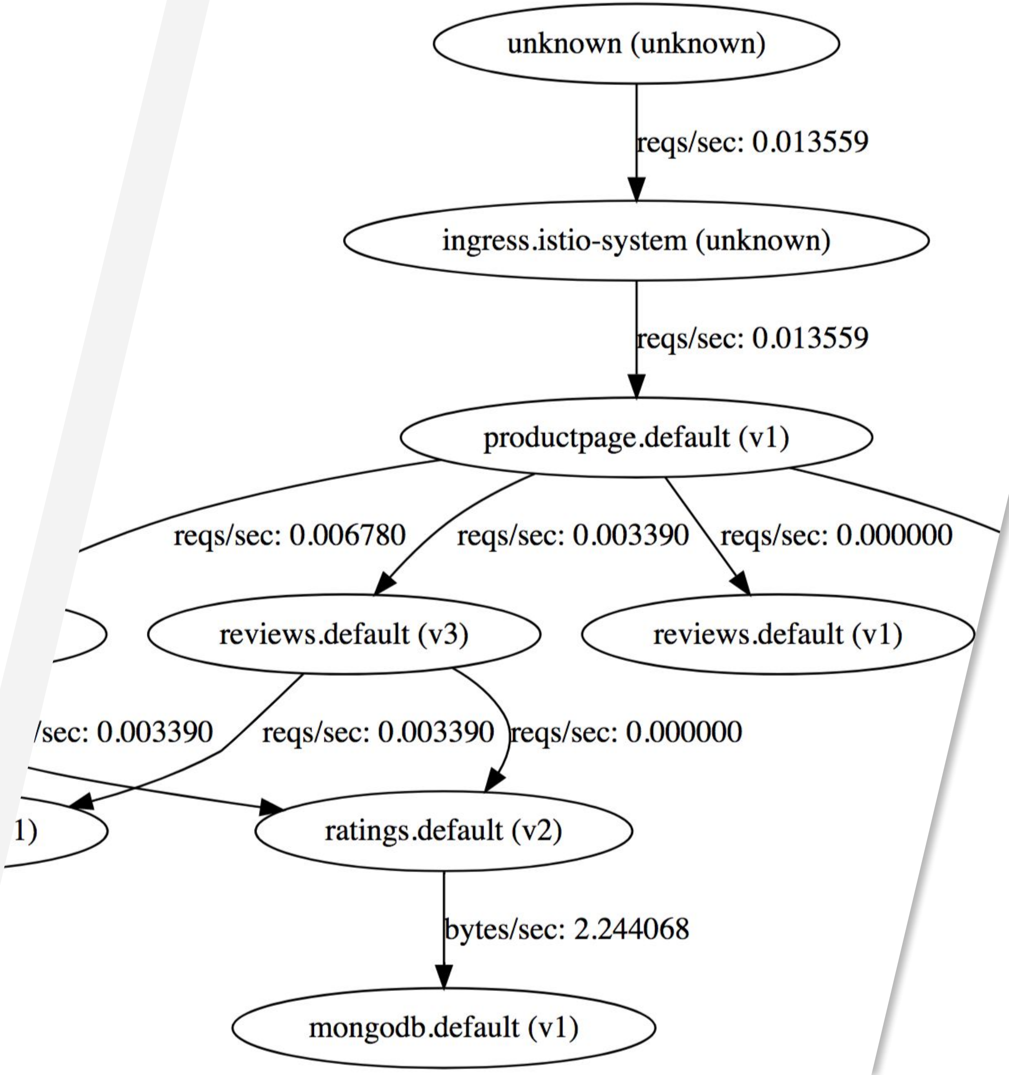
Istio.io

Service Mesh

Opentracing

Envoy

Metrics and overview



Monitoring

No default get-for-free monitoring

More logs than metrics

Infrastructure, not services

Monitoring made for management and ops, not for developers

Prometheus/ Grafana

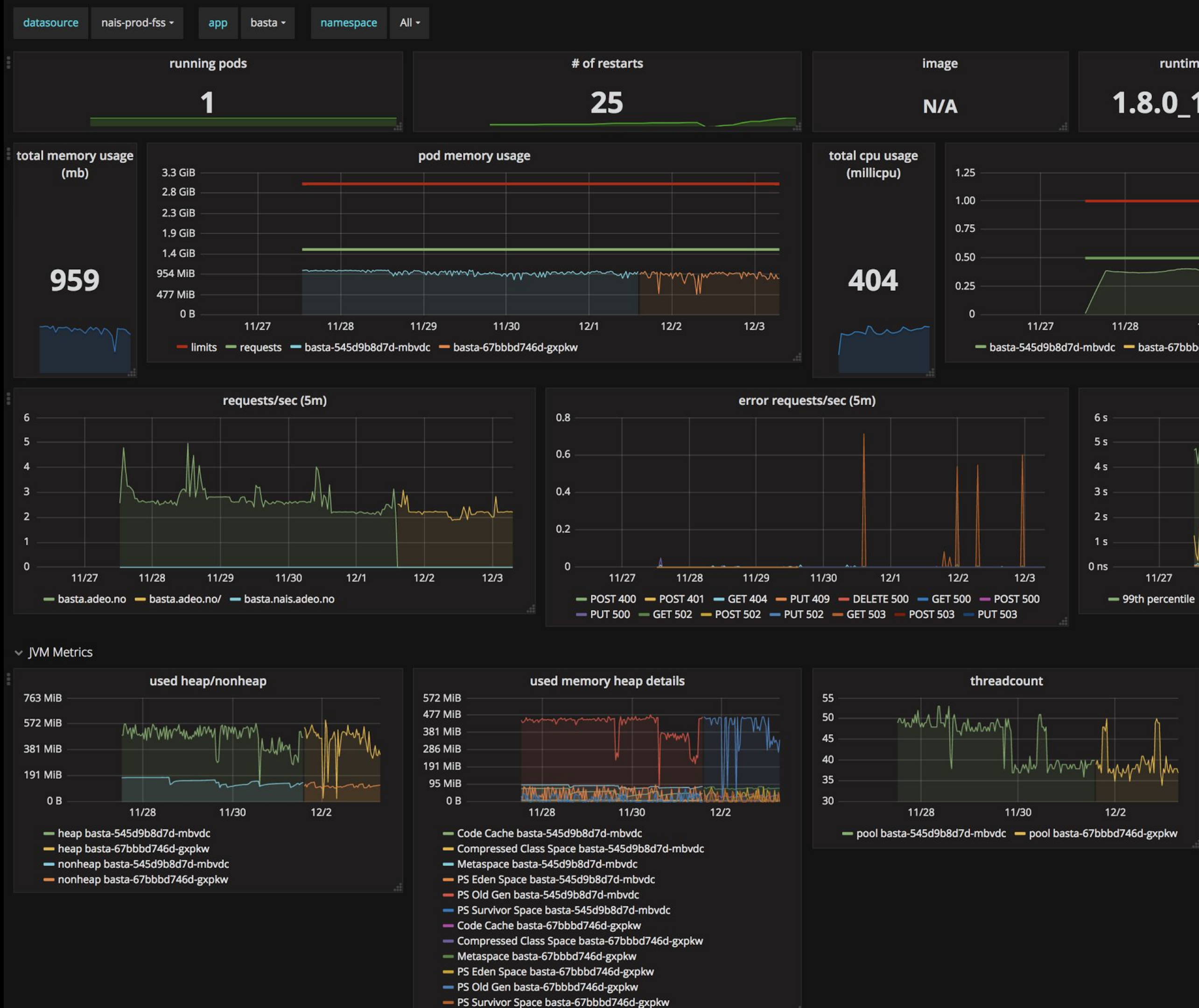
K8s metadata attached to timeseries in prometheus, and visualised in Grafana..

Default dashboard for every app

Prometheus DefaultExports

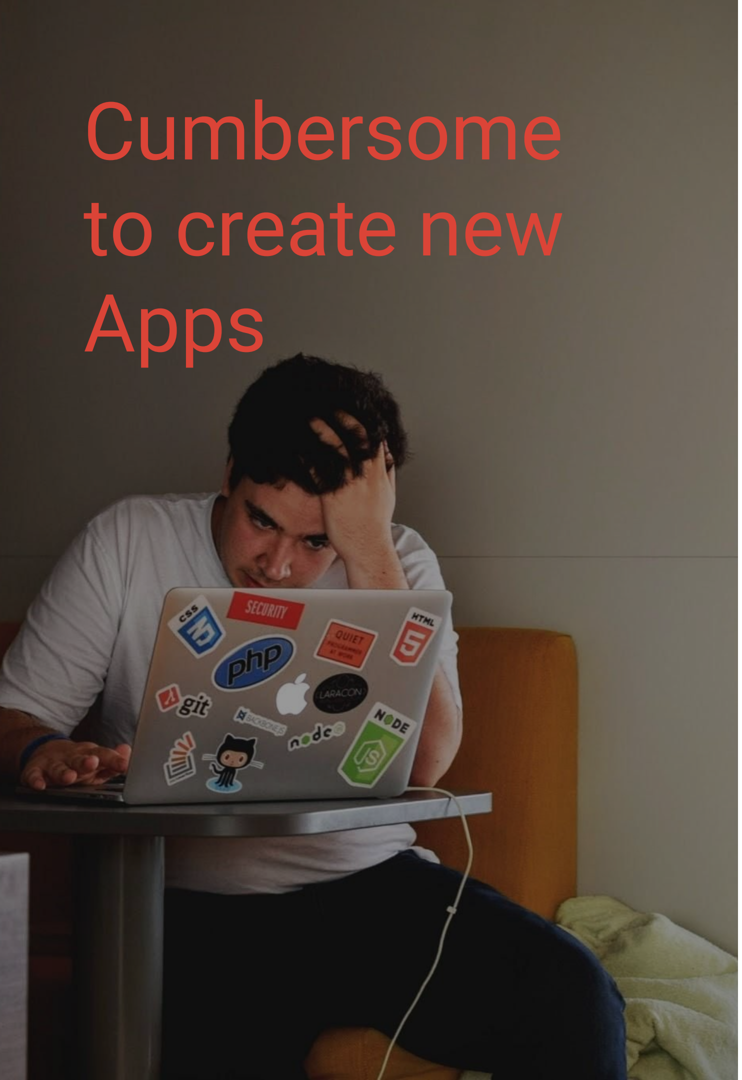
Heapster

“billing”



datasource nais-prod-fss app basta namespace All

Cumbersome to create new Apps



Manual provisioning of vmware servers

Manual setup of databases, loadbalancers, service discovery.

Steps

- Create a VM of a special type
- Create a Pipeline
- Create Config in another web-app
- Order deployment in Jira

Nais Deployment Daemon

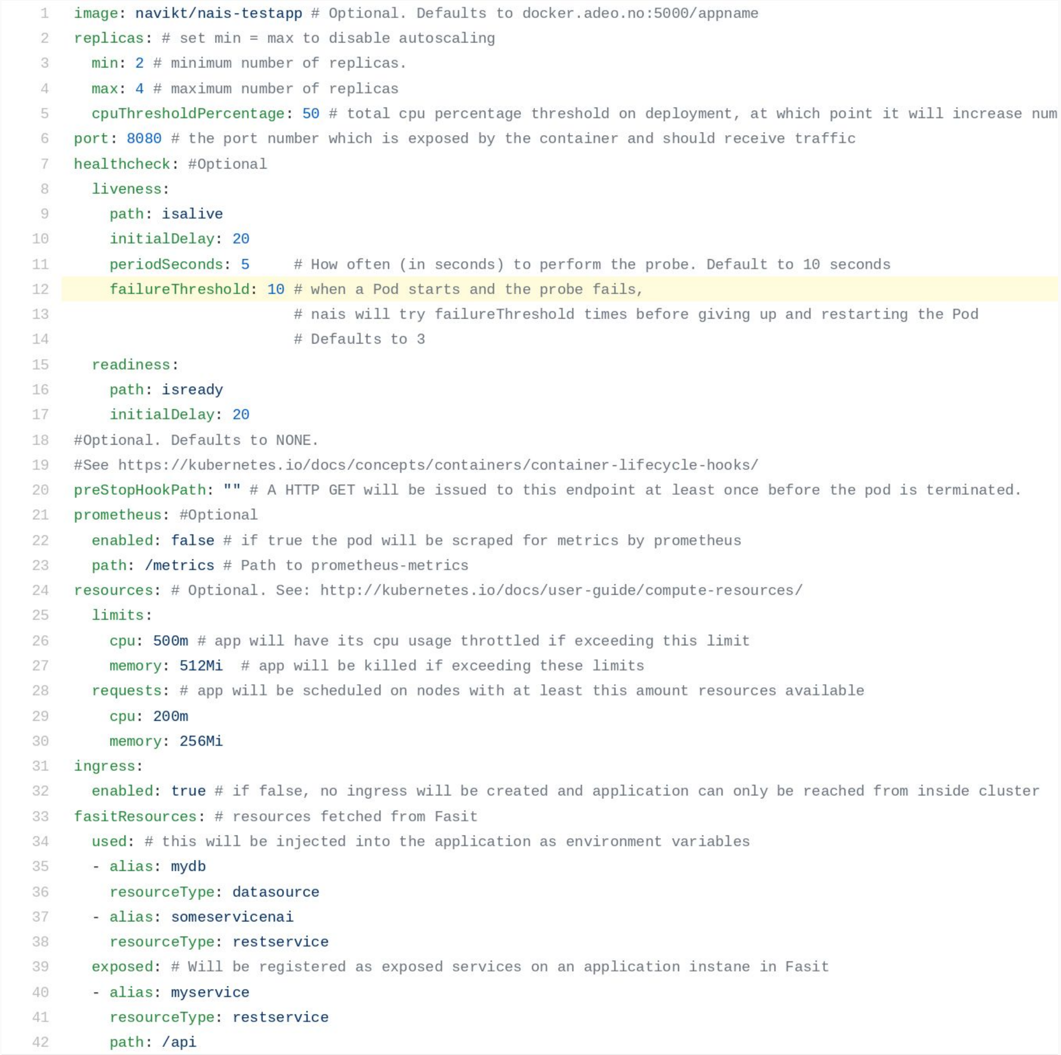
Nais.yaml

Opinionated config applications on a kubernetes cluster with sensible defaults

Integrates with platform applications for metrics, log-aggregation, ingress

Defaults

- Autoscaler
- isAlive/isReady



```yaml
image: navikt/nais-testapp # Optional. Defaults to docker.adeo.no:5000/appname
replicas: # set min = max to disable autoscaling
  min: 2 # minimum number of replicas.
  max: 4 # maximum number of replicas
  cpuThresholdPercentage: 50 # total cpu percentage threshold on deployment, at which point it will increase num
port: 8080 # the port number which is exposed by the container and should receive traffic
healthcheck: #Optional
  liveness:
    path: isalive
    initialDelay: 20
    periodSeconds: 5     # How often (in seconds) to perform the probe. Default to 10 seconds
    failureThreshold: 10 # when a Pod starts and the probe fails,
                         # nais will try failureThreshold times before giving up and restarting the Pod
                         # Defaults to 3
  readiness:
    path: isready
    initialDelay: 20
#Optional. Defaults to NONE.
#See https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/
preStopHookPath: "" # A HTTP GET will be issued to this endpoint at least once before the pod is terminated.
prometheus: #Optional
  enabled: false # if true the pod will be scraped for metrics by prometheus
  path: /metrics # Path to prometheus-metrics
resources: # Optional. See: http://kubernetes.io/docs/user-guide/compute-resources/
  limits:
    cpu: 500m # app will have its cpu usage throttled if exceeding this limit
    memory: 512Mi  # app will be killed if exceeding these limits
  requests: # app will be scheduled on nodes with at least this amount resources available
    cpu: 200m
    memory: 256Mi
ingress:
  enabled: true # if false, no ingress will be created and application can only be reached from inside cluster
fasitResources: # resources fetched from Fasit
  used: # this will be injected into the application as environment variables
  - alias: mydb
    resourceType: datasource
  - alias: someservicenai
    resourceType: restservice
  exposed: # Will be registered as exposed services on an application instane in Fasit
  - alias: myservice
    resourceType: restservice
    path: /api
```

Batch jobs

Batch and the application in the same artifact is a common pattern

Resource contention

Difficult to scale

Run at night, with dedicated operators (as in people)

Batch jobs in Nais

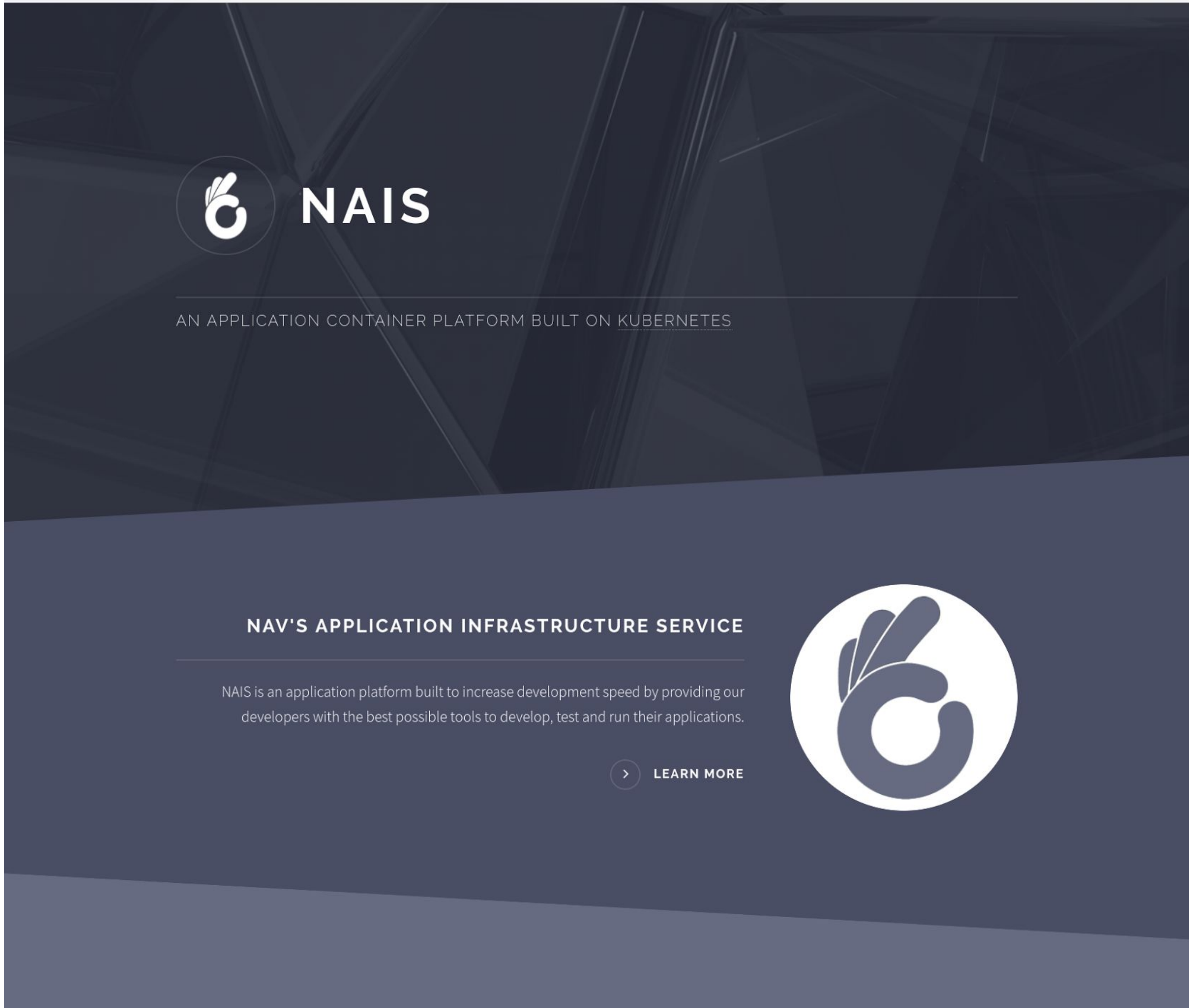
Separate containers

nais.job.yaml

Run when capacity is available

Subset of k8s functionality

Nais.io

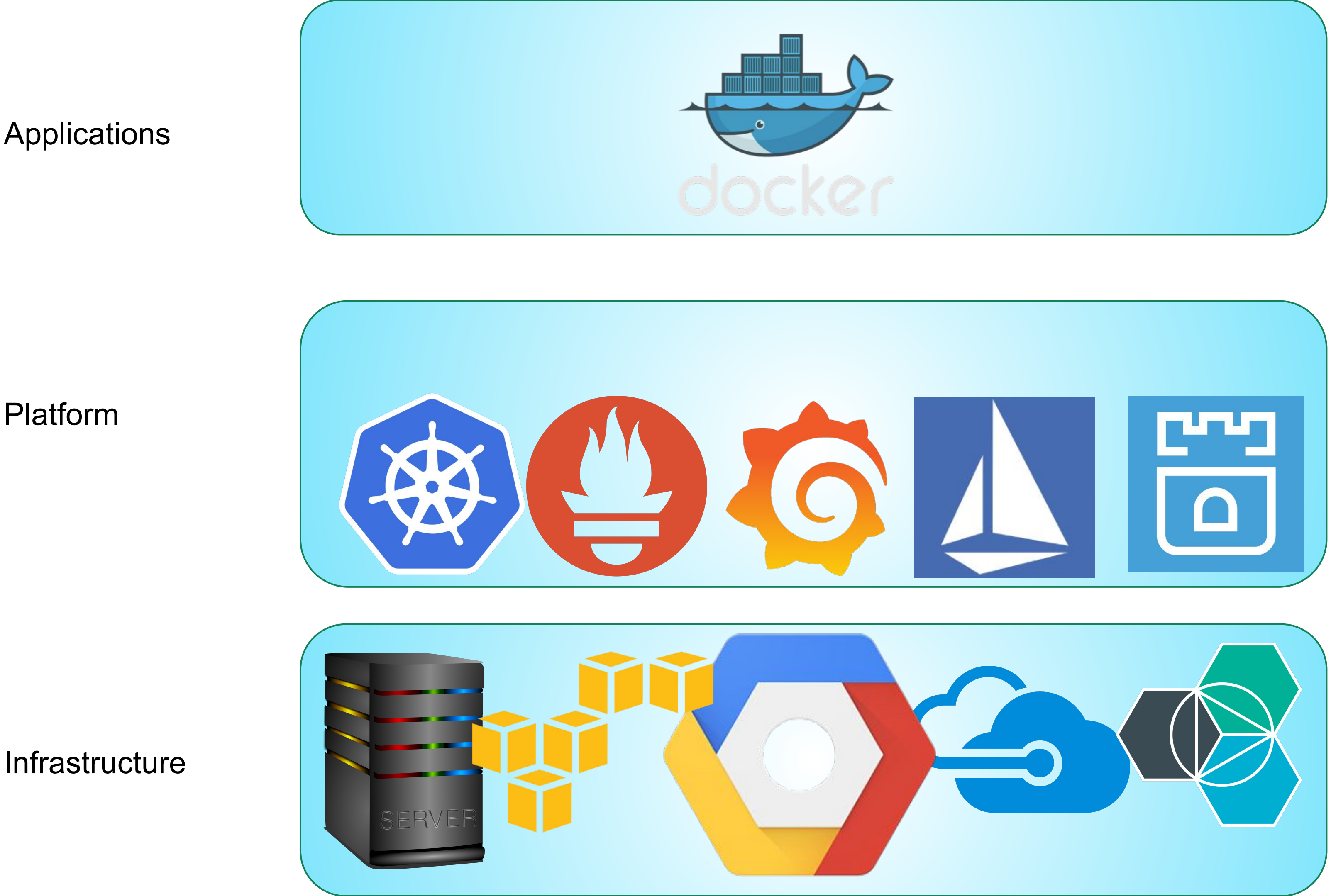


NAIS

AN APPLICATION CONTAINER PLATFORM BUILT ON KUBERNETES

NAV'S APPLICATION INFRASTRUCTURE SERVICE

NAIS is an application platform built to increase development speed by providing our developers with the best possible tools to develop, test and run their applications.



Applications

Platform

Infrastructure

Migration of Apps

Migrating apps are more difficult than building a platform

Nais is built to solve migration of legacy applications not building the perfect platform (that is step 2)

Reuse parts of the private cloud

Continuous delivery of cluster and platform components

Ansible

Helm with Landscaper

Nais deploy daemon

Storage

Rook used to set up Ceph

Postgres Operator in the future

Neo4j

Open Source

Creates community and enthusiasm

Pull request gives you cake

Government funded code, should be open

Conclusions

Conclusions

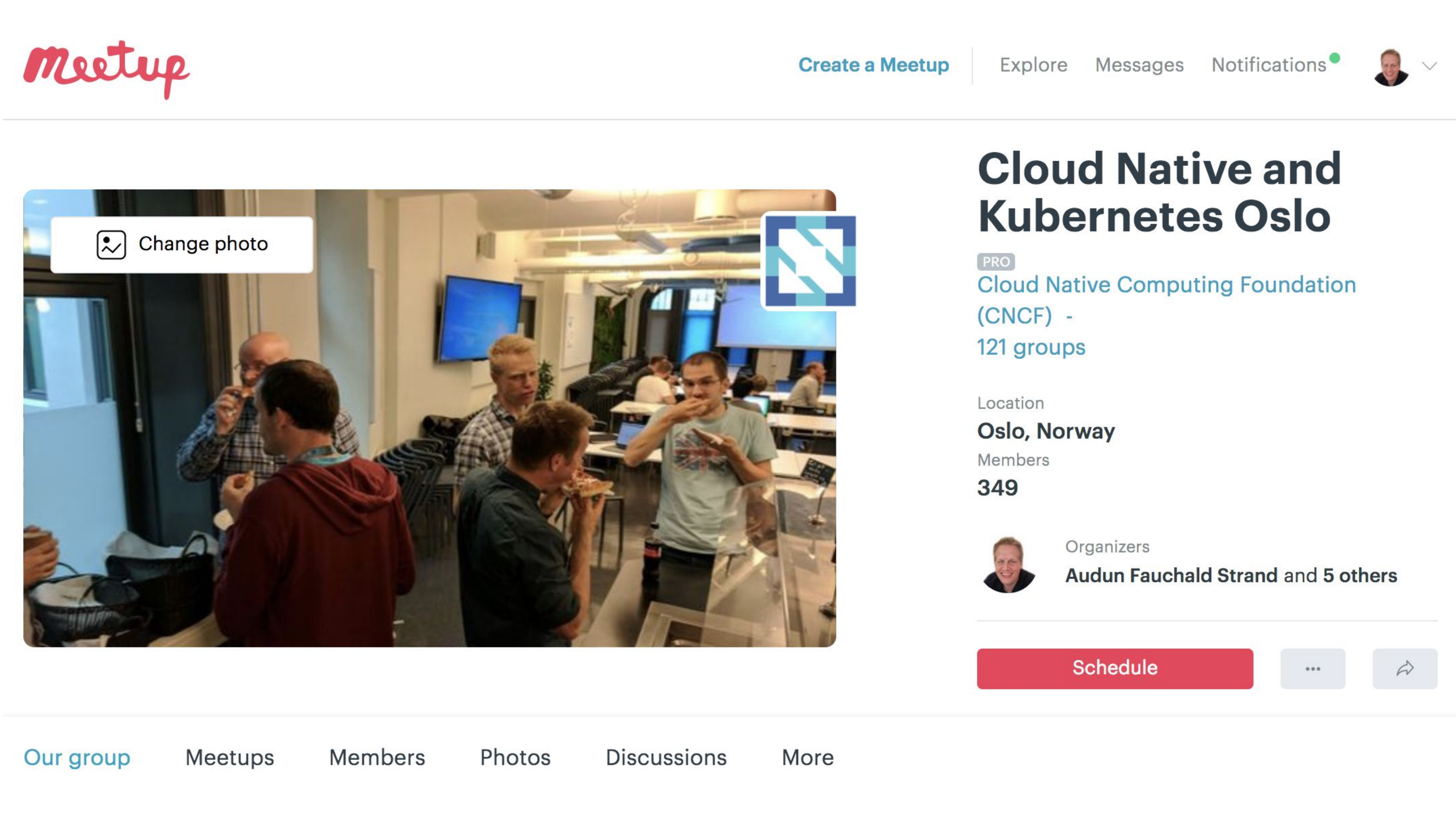
Kubernetes is great for building a PAAS that support migrations of legacy

Focus on migration, not on building a perfect platform

Build a brand around your internal platform

Open Source you code

Kubernetes helps build a you build it, you run it culture



Cloud Native and Kubernetes Oslo

PRO
Cloud Native Computing Foundation (CNCF) -
121 groups

Location
Oslo, Norway

Members
349

Organizers
Audun Fauchald Strand and **5 others**

Questions

HOW
HY
WHEN
WHERE
WHO
HERE
WHERE
WHY
WHO
WHEN
WHAT
WHY
WHAT
WHEN
WHEN
WHO
WHERE
WHY
HO