



Building and testing infrastructure NAV style

Hans Kristian Flaatten

May the 4th 2023 // Cloud Native Oslo

Photo by Michael Marais on Unsplash





Hans Kristian Flaatten
Starefossen

Dad, husband, and outdoor 🏔 Kubernetes
Platform @nais and @navikt, Previously
@tietoevry @Turistforeningen



// NAV

The screenshot shows the GitHub repository page for 'nais'. The repository has a verified badge and 136 public repositories. It features pinned projects like 'doc', 'examples', 'wonderwall', 'naiserator', 'device', and 'tokendings'. The right sidebar displays a list of team members with their profile pictures.

nais
NAV Application Infrastructure Service

Oslo <https://nais.io> Verified

Overview Repositories 136 Projects 1 Packages People 3

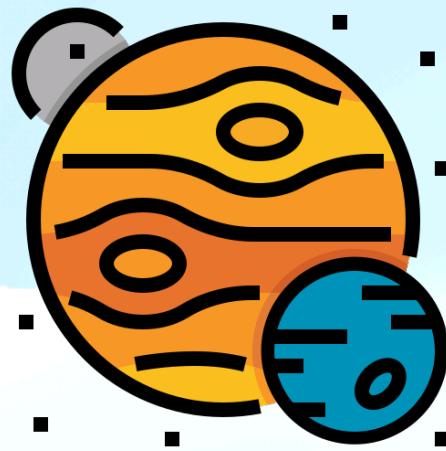
Pinned

- doc** Public
nais developer documentation
HCL ⭐ 28 ⚡ 49
- examples** Public
A collection of example apps demonstrating features and concepts in and around NAIS.
Kotlin ⭐ 4
- wonderwall** Public
openid connect relying party as a sidecar/service
Go ⭐ 5 ⚡ 1
- naiserator** Public
Naiserator creates a full set of Kubernetes application infrastructure based on a single application spec.
Go ⭐ 25 ⚡ 9
- device** Public
naisdevice is a application suite that enables NAV developers to connect to internal resources in a secure and friendly manner.
Go ⭐ 13 ⚡ 4
- tokendings** Public
Service implementing some parts of OAuth 2.0 Token Exchange (<https://www.rfc-editor.org/rfc/rfc8693.html>)
Kotlin ⭐ 11 ⚡ 1

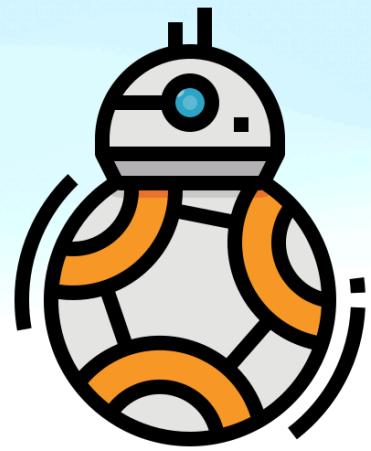
People

-
-
-
-
-
-
-
-

Agenda



Infrastructure @ NAV



Infrastructure as Code



Infrastructure as Software

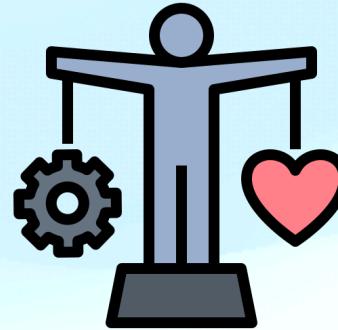
It is NAIS to work @ NAV



Common goal & purpose



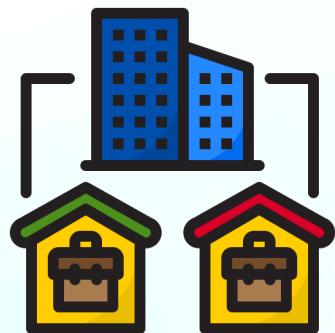
Best together



Work-life balance



Take responsibility



Hybrid work environment



Skilled colleagues

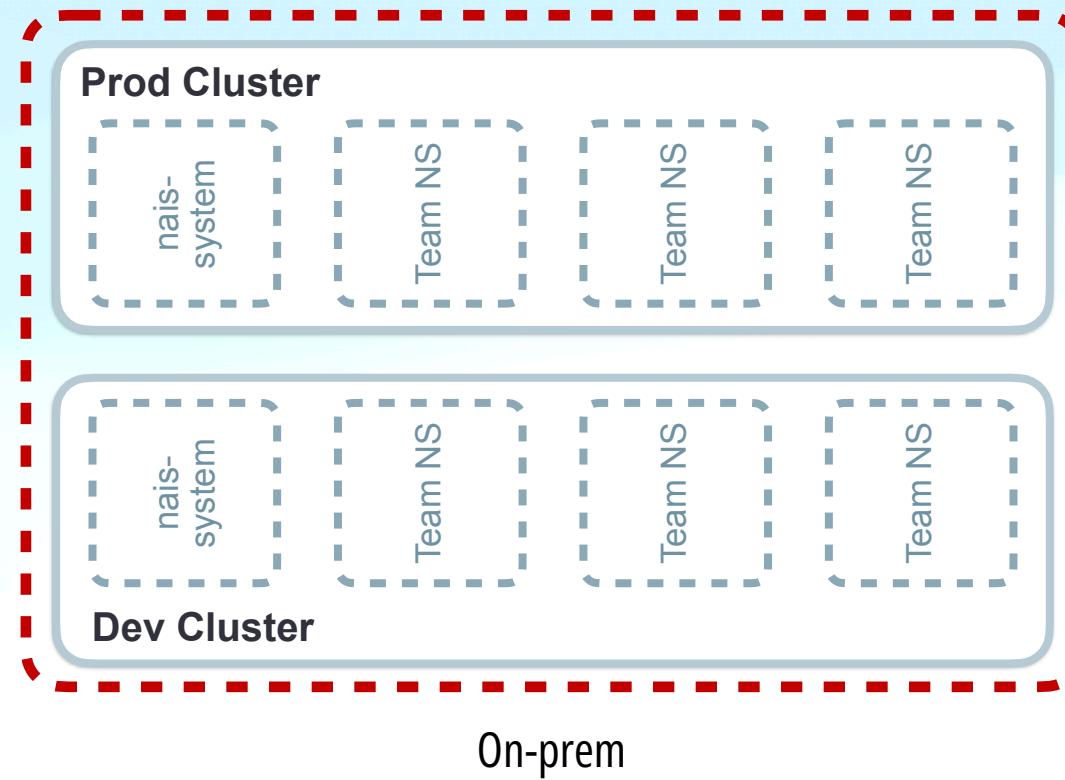


Right tools™

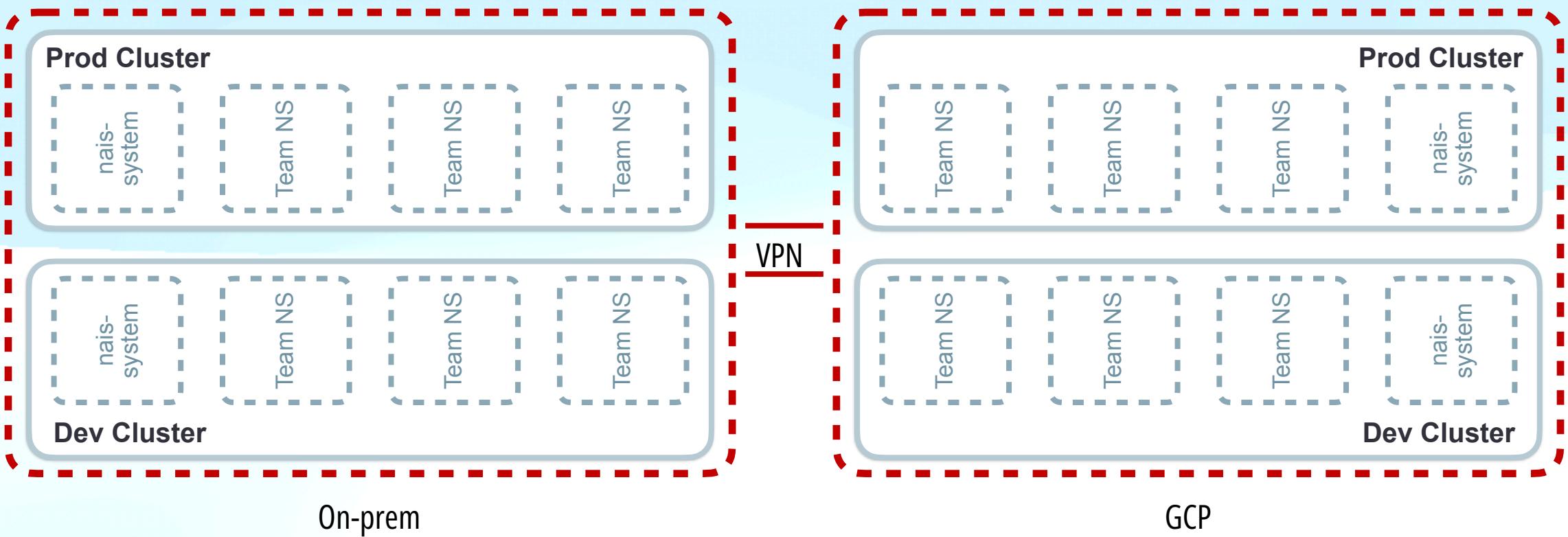


detsombetyrnoe.no

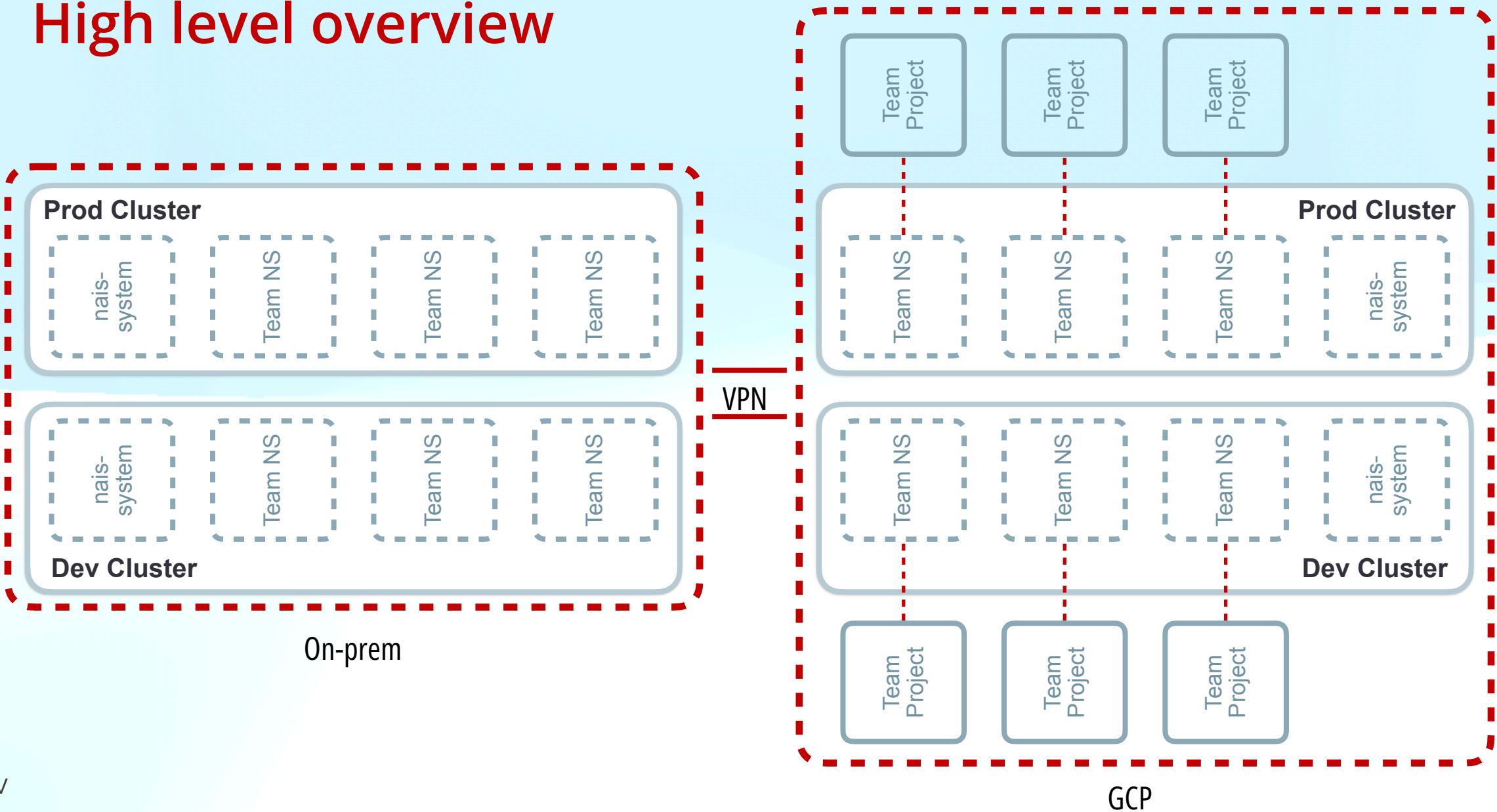
High level overview



High level overview



High level overview



Some NAIS stats

204

Teams

1641

Apps

59

Buckets

262

Databases

594

Kafka topics

Some NAIS stats

204

Teams

3282

Apps

118

Buckets

524

Databases

1188

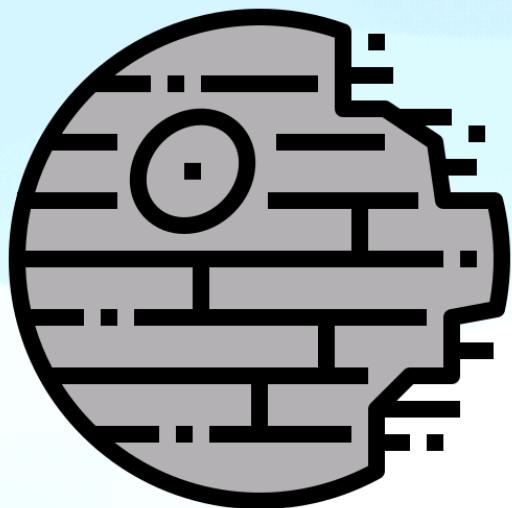
Kafka topics



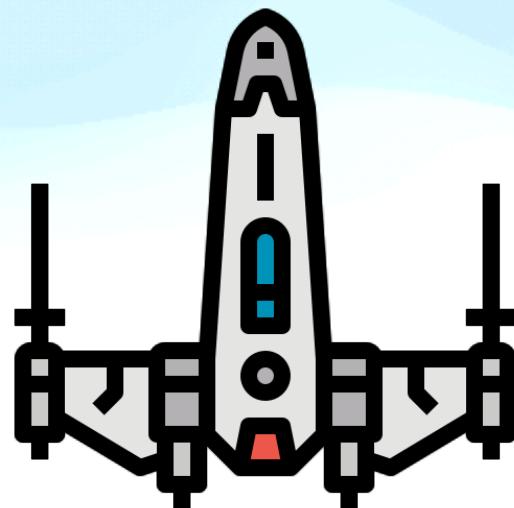
// NAV

Source: giphy.com

Two types of infrastructure

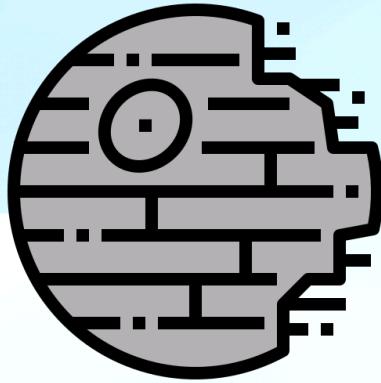


Infrastructure as Code

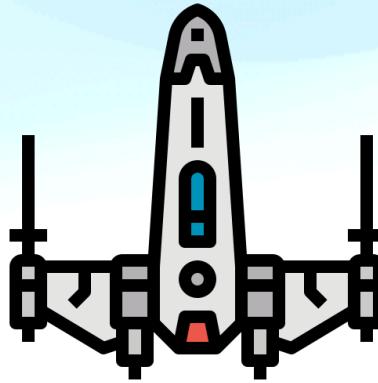


Infrastructure as Software

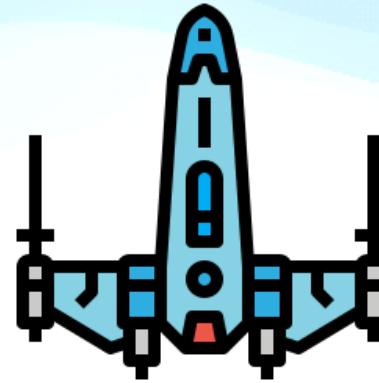
Two types of infrastructure



Infrastructure as Code

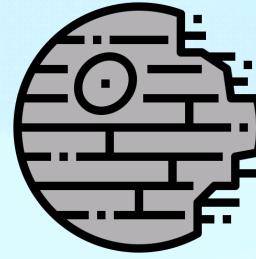
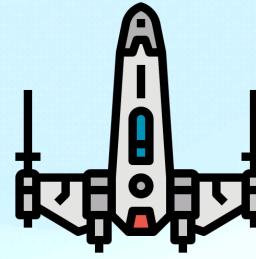
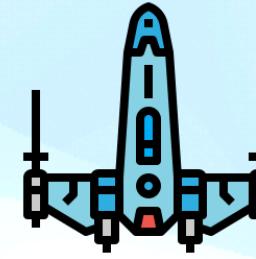


Portals



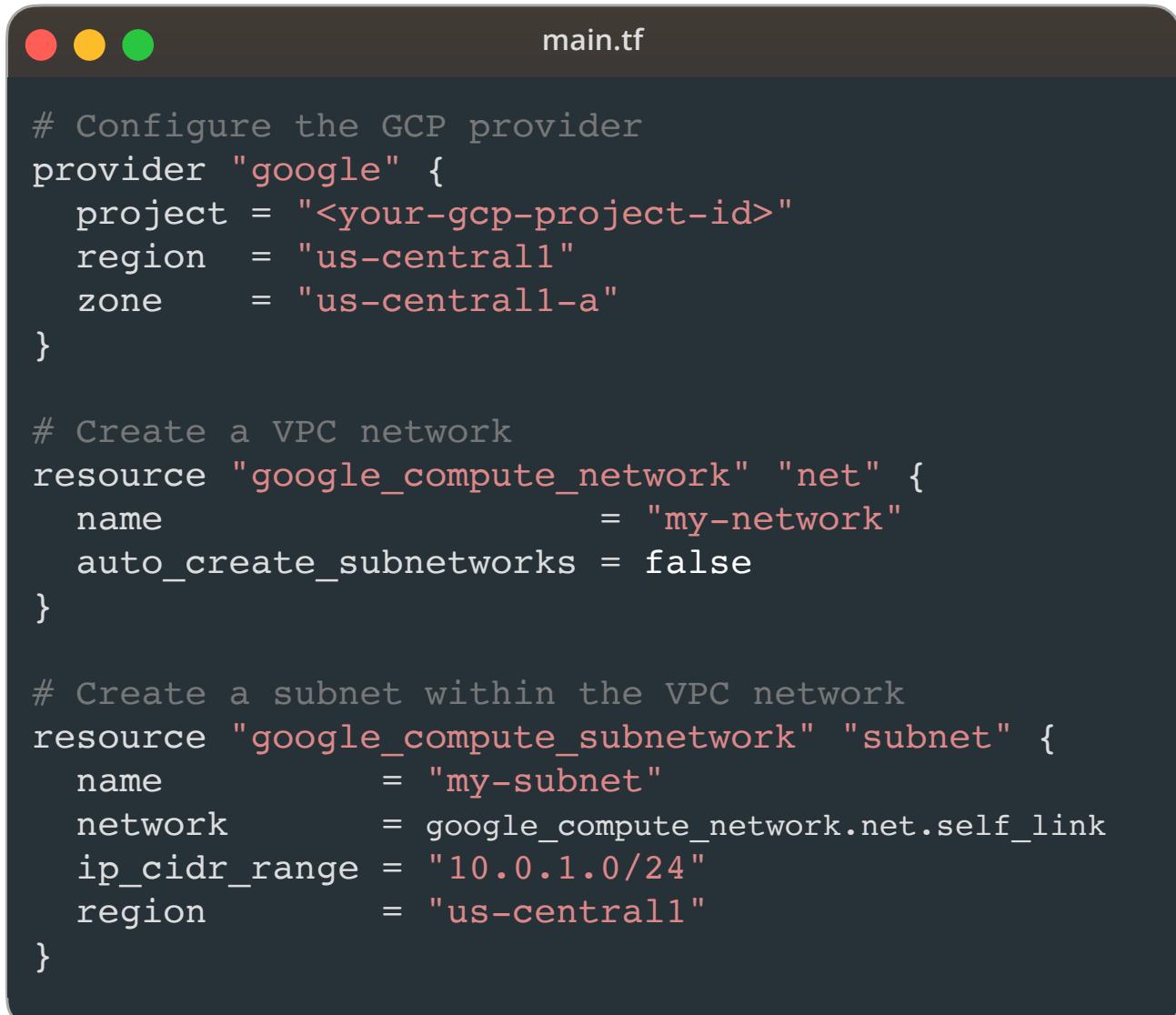
Operators

Two types of infrastructure

			
	Infrastructure as Code	Portals	Operators
Scope	Organisation	Team	Application
Change rate	Slow-ish	Medium	High

Infrastructure as Code

- Pros
 - Quick to set up
 - For the simple things
- Cons
 - Messy to maintain
 - Becomes complex



The code block displays a Terraform configuration file named 'main.tf'. It starts by configuring the GCP provider with a project ID, region, and zone. Then, it creates a VPC network named 'my-network' with auto-create-subnetworks set to false. Finally, it creates a subnet named 'my-subnet' within the network, specifying an IP range of 10.0.1.0/24 and the same region as the network.

```
# Configure the GCP provider
provider "google" {
  project = "<your-gcp-project-id>"
  region  = "us-central1"
  zone    = "us-central1-a"
}

# Create a VPC network
resource "google_compute_network" "net" {
  name            = "my-network"
  auto_create_subnetworks = false
}

# Create a subnet within the VPC network
resource "google_compute_subnetwork" "subnet" {
  name            = "my-subnet"
  network         = google_compute_network.net.self_link
  ip_cidr_range  = "10.0.1.0/24"
  region          = "us-central1"
}
```

Infrastructure as Code



Atlantis – Terraform Pull Request Automation



STEP 1 Open a Terraform pull request

A screenshot of a GitHub pull request interface. The title bar says "Open lkysow wants to merge 1 commit into master from atlantis-workflow-demo". Below the title are buttons for Conversation (0), Commits (1), Checks (0), and Files changed (1). A "Review changes" button is at the top right. The main area shows a diff for "main.tf" with one addition and one deletion. The addition is "+ resource "null_resource" "demo" {}". The URL of the pull request is <https://github.com/lkysow/atlantis-workflow-demo/pull/1>.

```
diff --git a/main.tf b/main.tf
--- a/main.tf
+++ b/main.tf
@@ -1 +1 @@
+resource "null_resource" "demo" {}
```



STEP 2

Atlantis automatically runs `terraform plan` and comments back on the pull request



atlantisbot commented

Ran Plan in dir: . workspace: default

```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```

```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
```

```
Terraform will perform the following actions:
```

```
+ null_resource.demo
    id: <computed>
Plan: 1 to add, 0 to change, 0 to destroy.
```

- To apply this plan, comment:
 - `atlantis apply -d .`
- To delete this plan click [here](#)
- To plan this project again, comment:
 - `atlantis plan -d .`

- To apply this plan, comment:
 - atlantis apply -d .
- To delete this plan click [here](#)
- To plan this project again, comment:
 - atlantis plan -d .

-
- To apply all unapplied plans, comment:
 - atlantis apply



STEP 3 Someone on your team reviews the plan and approves the pull request



alice approved these changes

[View changes](#)



STEP 4 You comment 'atlantis apply'



Ikysow commented

atlantis apply



Ikysow commented

atlantis apply



STEP 5 Atlantis runs `terraform apply` and comments back on the pull request



atlantisbot commented

Ran Apply in dir: . workspace: default

```
null_resource.demo: Creating...
null_resource.demo: Creation complete after 0s (ID: 4542221565395344699)
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.



STEP 6 Merge the pull request



Pull request successfully merged and closed

You're all set—the atlantis-workflow... branch can be safely deleted.

Delete branch

A step in the right direction...

Terratest
by Gruntwork.io



Pulumi



You must *unlearn* what you have learned.

Infrastructure as Software

- Real programming language
 - Go is a good fit 😊
- Test framework of choice
- Maintainable code
- Lifecycle management

```
package main

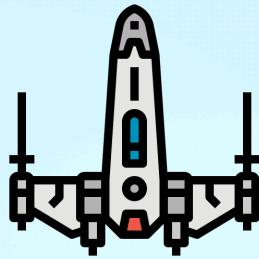
// Imported packages
import (
    "fmt"
    "log"
    "net/http"
)

// Server handler function
func handler(w http.ResponseWriter, r *http.Request)
    fmt.Fprintf(w, "Hello, World!")
}

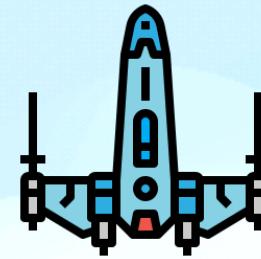
// Main program function
func main() {
    http.HandleFunc("/", handler)

    log.Fatal(http.ListenAndServe(":8080", nil))
}
```

Portals and Operators



Portals



Operators

Web UI

Kubernetes API

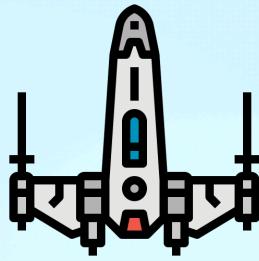
Team centric

App centric

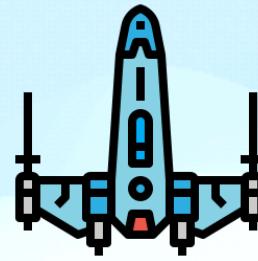
Uses operators

Uses (other) operators

Portals and Operators



Portals



Operators

Manage my team

Create a Kafka topic

Unleash instance

ID-porten client

NAIS Console (Portal)

<https://github.com/nais/console>

The screenshot shows the NAIS Console (Portal) interface. At the top, there's a header bar with the NAIS logo, a title bar showing 'NAIS console', and a navigation bar with back, forward, and search icons. The URL in the address bar is https://console.example.com/teams/basseng. The main content area has a dark header with the NAIS logo and the word 'CONSOLE'. On the left, a sidebar shows a 'Teams' section with a single item: 'basseng'. The main content area displays information for the 'Team basseng' page. It includes a summary message: 'Bassengteamet skal bade i bassenget', and sections for 'Slack channel' (#basseng), 'Slack alert channels' (ci #basseng, dev #basseng), and a status message: 'Status: last fully synchronized on 2023-05-04T13:20:03.647Z.' Below this is a 'Members' table with three rows:

Email	Role
admin@dev-nais.io	Owner
hans@dev-nais.io	Owner
kimt@dev-nais.io	Owner

At the bottom left, there's a red 'NAV' button.

NAIS Console (Portal)

The screenshot shows the NAIS Console interface for the team 'basseng'. At the top, there's a header bar with a back button, a plus sign for adding new teams, the URL 'https://console.example.com/teams/basseng', a star icon, and user information 'hans flaatten' and 'Logout'. Below the header, the main content area has a dark header with 'CONSOLE' and a 'Logout' button. The main content shows the 'Team basseng' details:

- Team basseng**: Bassengteamet skal bade i bassenget.
- Slack channel**: #basseng
- Slack alert channels**: Per-environment slack channels to be used for alerts sent by the platform.
 - ci : #basseng
 - dev : #basseng
- Status: last fully synchronized on 2023-05-04T13:20:03.647Z.*

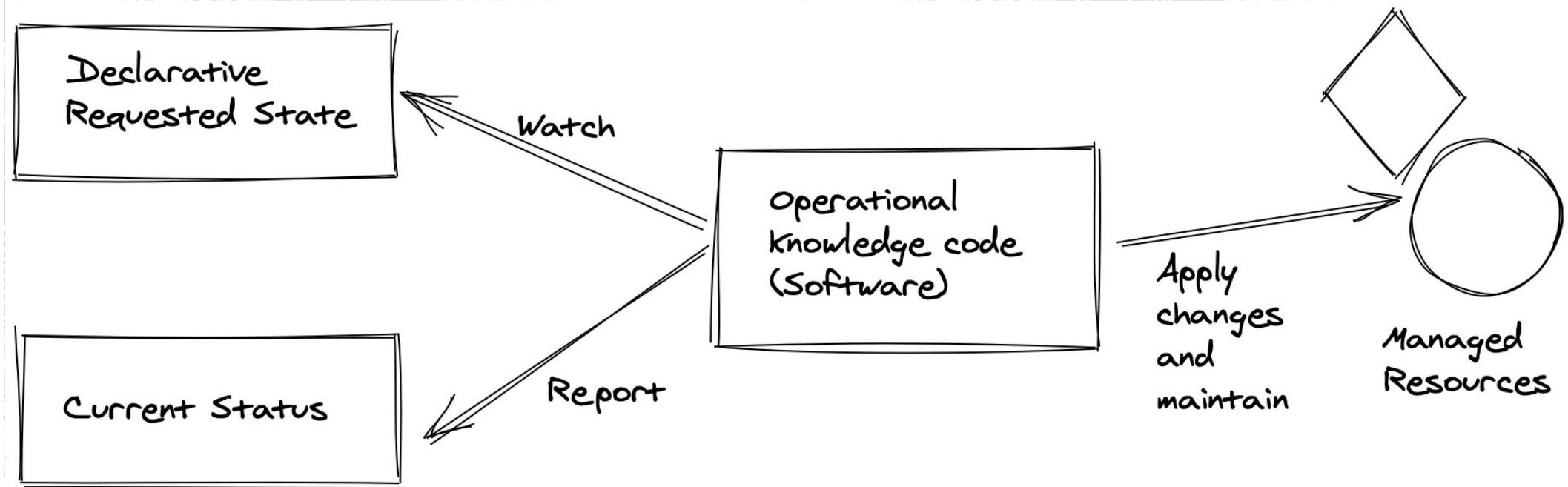
At the bottom, there's a 'Members' section with a table:

Email	Role	Edit
admin@dev-nais.io	Owner	Edit
hans@dev-nais.io	Owner	Edit
kimt@dev-nais.io	Owner	Edit

Responsible for

- AzureAD group
- GitHub team
- Google Cloud
 - Project, Registry
- Kubernetes
 - Namespace, Deploy

Kubernetes Operators



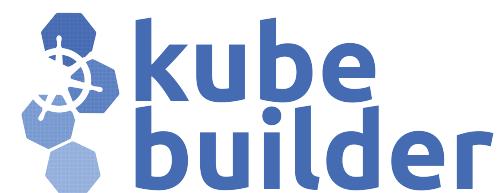
Infrastructure as Operators

<https://github.com/nais/naiserator>

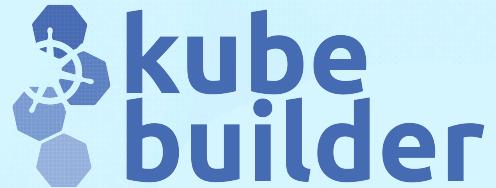
```
apiVersion: "nais.io/v1alpha1"
kind: "Application"
metadata:
  name: "my-app"
  namespace: "my-team"
  labels:
    team: "my-team"
spec:
  image: "ghcr.io/navikt/my-app:my-tag"
  ...
```



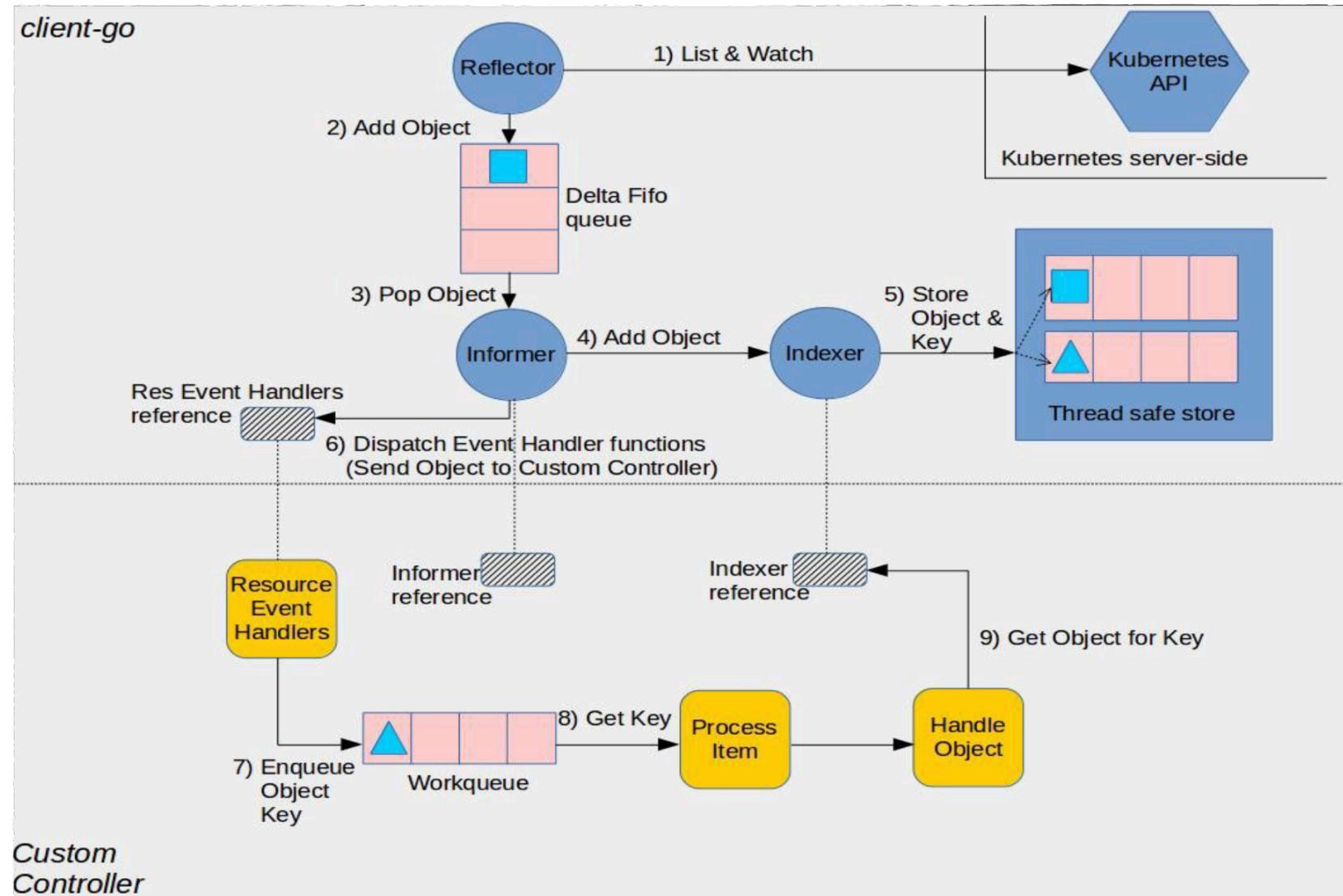
And more...



Kubebuilder

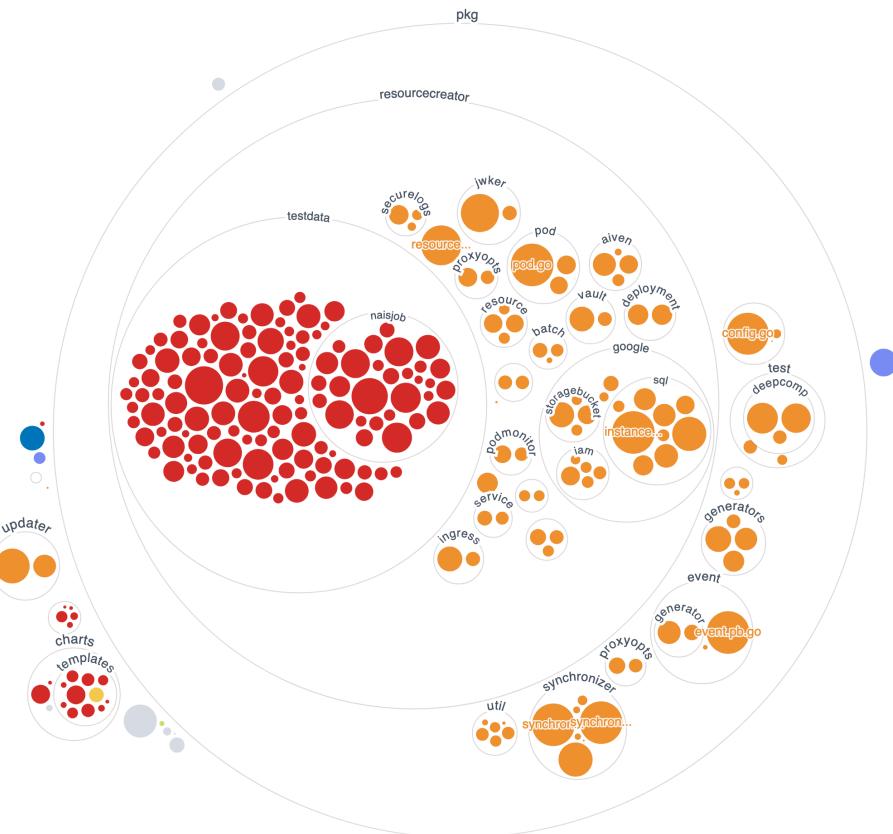


- [kubebuilder](#)
- [controller-runtime](#)
- [controller-tools](#)

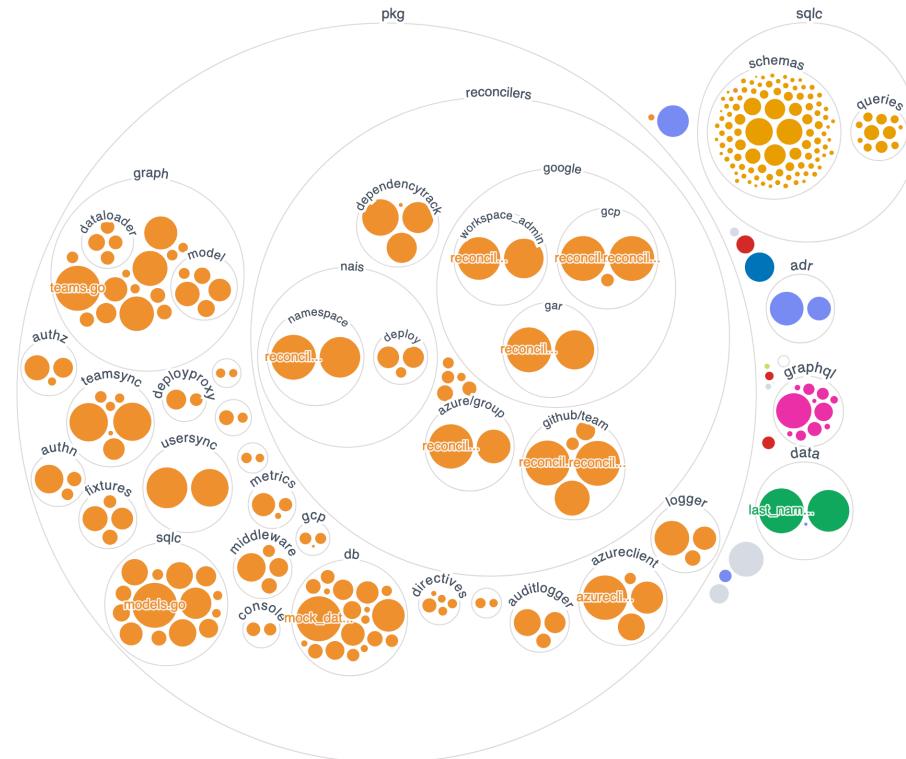


Building and testing Operators

nais/naiserator



nais/console



- .cmd
 - .go
 - .md
 - .mod
 - .sh
 - .sql
 - .tpl
 - .yaml
 - .yml

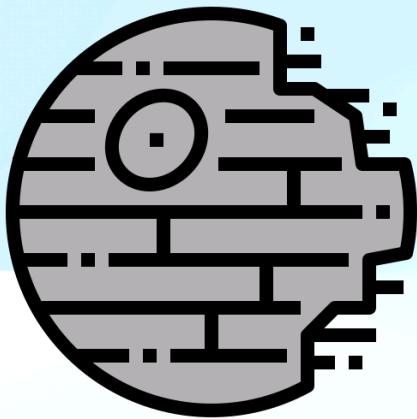
Testing Operators

- Unit level tests
- Integration tests (mocking)
- End-to-end test (Minikube)
- Canary deployment

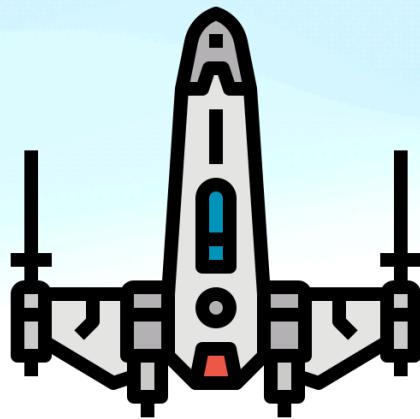


No more training do you require.

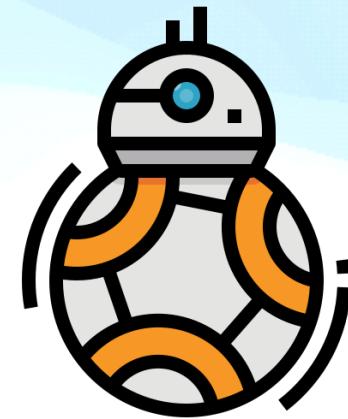
Summing it up



Infrastructure is hard



Apply software engineering



Full infrastructure lifecycle

NDC { Oslo }

22-26 May 2023

Wednesday

Room 6

Bulding a dedicated platform for frontend developers at NAV



Hans Kristian Flaatten

Andreas Nordahl

Talk (60 min)

Thursday

Workshop

Part 1/2: Securing Container Supply Chain Workshop



Hans Kristian Flaatten

Jan-Kåre Solbakken

Youssef Bel Mekki

Talk (60 min)

Friday

Room 2

Modelling vs Reality



Einar Høst

Talk (60 min)



github.com/nais

Hans Kristian Flaatten
Platform Engineer @NAV
hans.kristian.flattan@nav.no
@Starefossen