Cloud Native CI/CD with "GitOps"

DevOpsDays Copenhagen 2019

Kasper Nissen, Cloud Architect/SRE @lunarway
kni@lunarway.com @phennex

lunar way

$ whoami

Kasper Nissen (@phennex)

- Cloud Architect / Site Reliability Engineer @lunarway
- Organizer and Co-Founder at Cloud Native Aarhus
- Founder of Cloud Native Nordics Slack Community
- Blogger at kubecloud.io

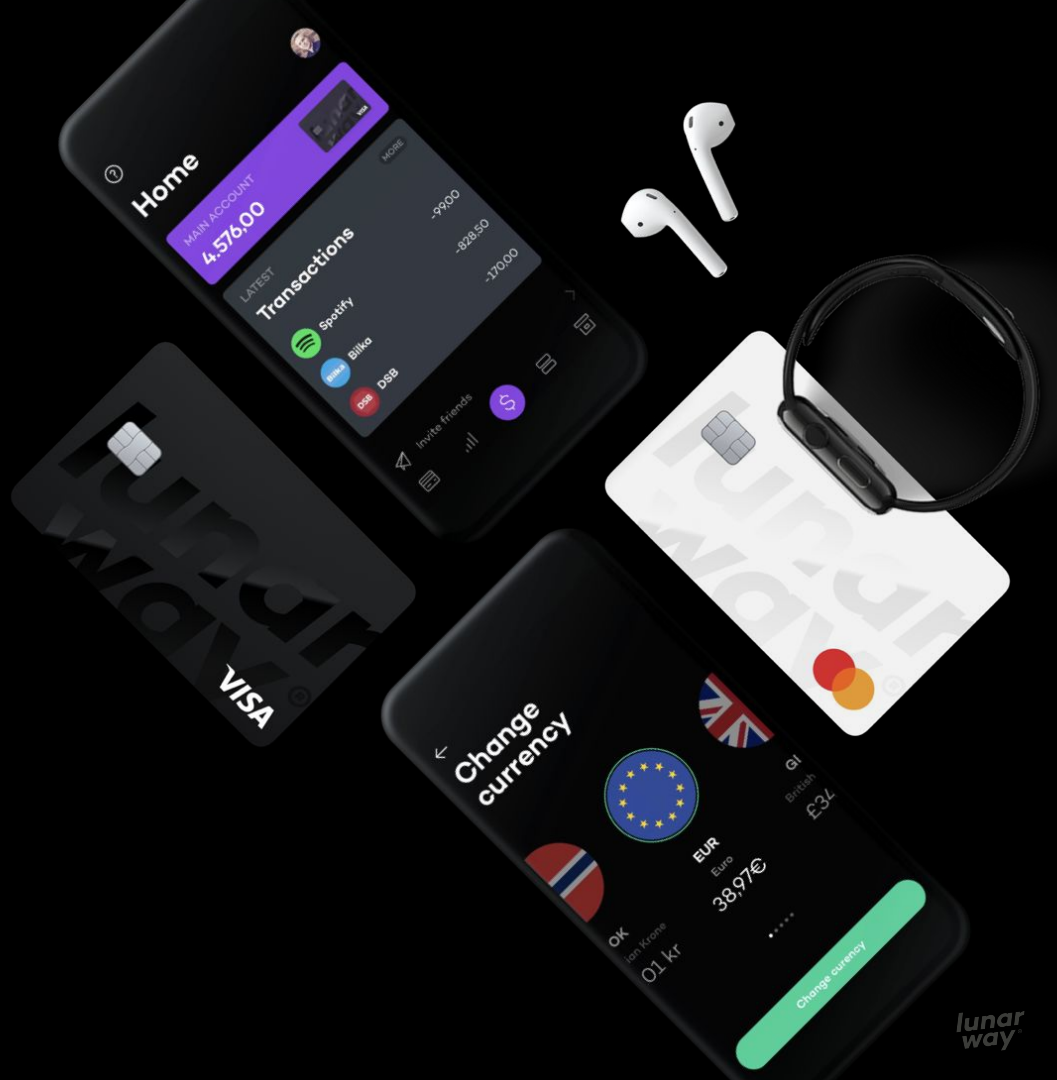
What is Lunar Way?

Our vision is to rethink the interaction with money.

Live in Sweden, Norway, and Denmark

3 Kubernetes clusters in AWS

80+ microservices in production



aws

Continuous integration is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

- - -

Continuous delivery is the ability to get changes of all types - including new features, configuration changes, bug fixes and experiments - into production, or into the hands of users, *safely* and *quickly* in a *sustainable* way.

This sounds like two different concerns?

… aren’t we all about decomposing, into smaller components with single responsibility?

The monolith

CI/CD

do we trust this guy with everything related to our pipeline, and basically with access to all our environments?

Single Responsibility

(microservices?)

CI

CD

Separating CI/CD into separate concerns as per the definition

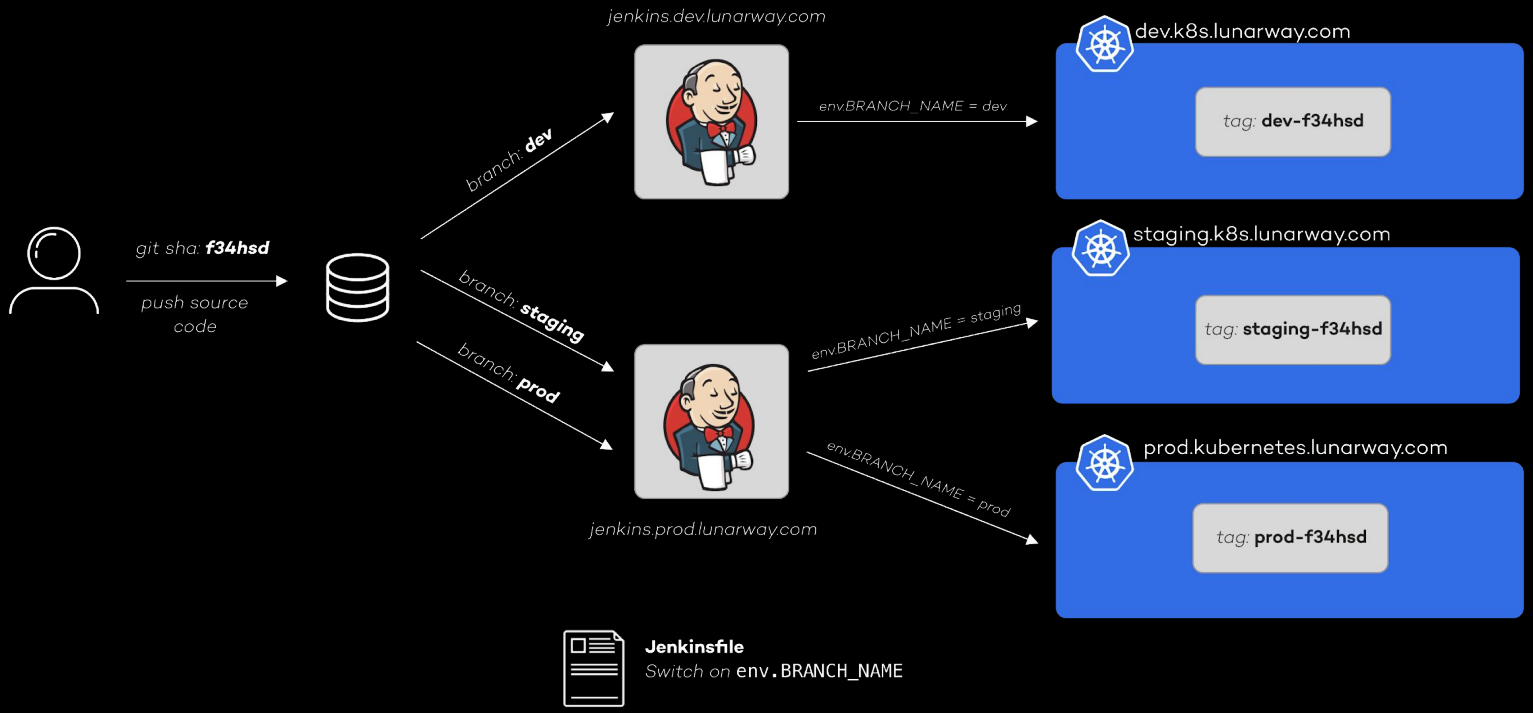
Introducing GitOps

GitOps is an extension of infrastructure as code that can be applied to Kubernetes workloads.

Configuration of applications is stored in Git that can be deployed automatically from Git and left untouched by manual operator intervention.

The term was coined by Alexis Richardson and the great folks at weaveworks

CI/CD at Lunar Way (before)



Problems?

- Long change lead time - because of multiple builds
- Building images for each branch is not feasible
- Jenkins has R/W access to the Kubernetes cluster
- No audit trail of deployments
- Branching hell

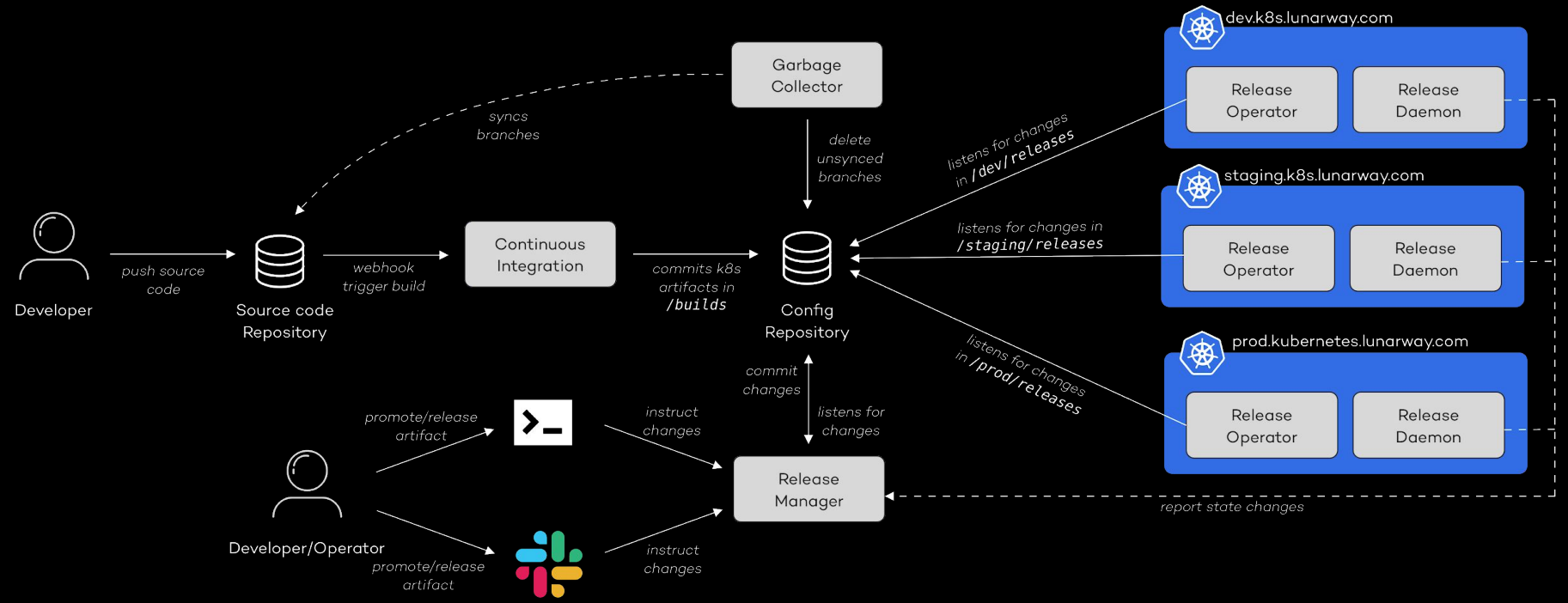
“Awesome” promote command to all environments

```
git checkout dev && git pull origin dev && git merge master && git push origin dev
&& git checkout staging && git pull origin staging && git merge dev && git push
origin staging && git checkout prod && git pull origin prod && git merge staging &&
git push origin prod && git checkout master
```

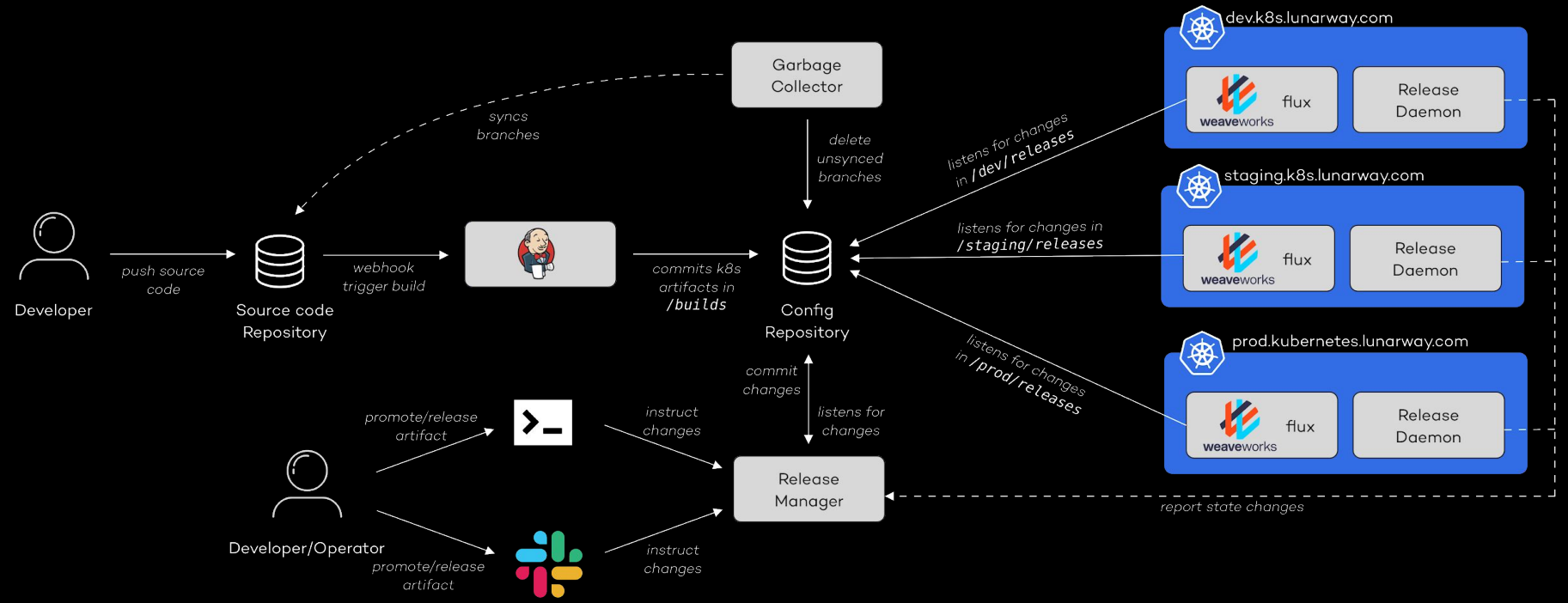
Problems?

- No “source of truth” of what is running making it hard in a disaster recovery situation

Separation of concerns

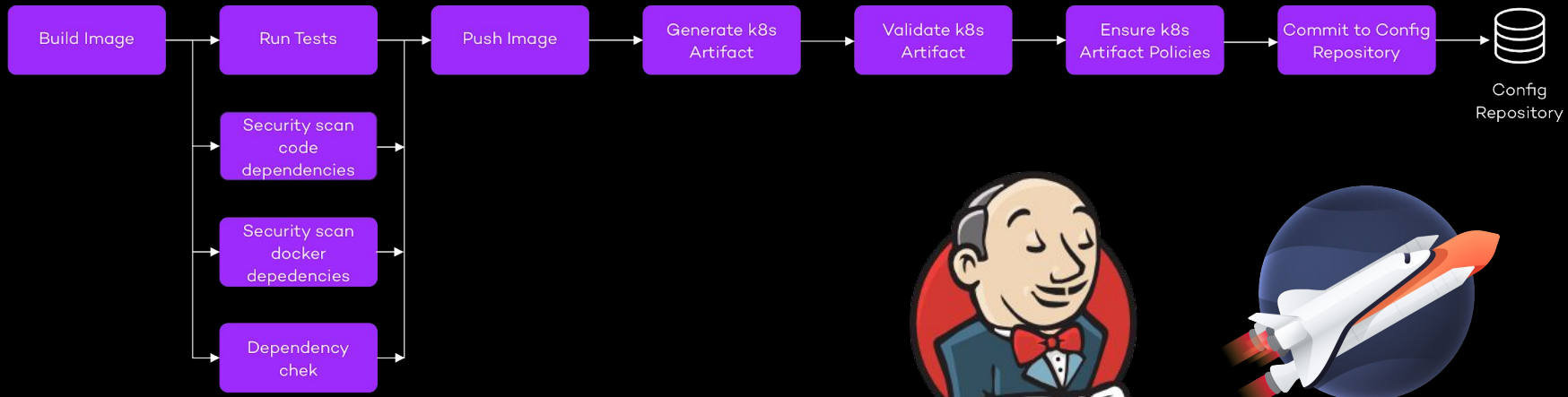


Cots vs custom



Continuous Integration

The responsibility of Continuous Integration is to ensure quality by executing test, checks, and scans of the code before handing it over.



shuttle

https://github.com/lunarway/shuttle

```
shuttle run build
shuttle run push tag=tag
```

Minimize groovy code to allow for easy migration

Build Artifacts

A build artifact is a JSON-blob that follows all builds from the CI pipeline. It contains all relevant information about the build.

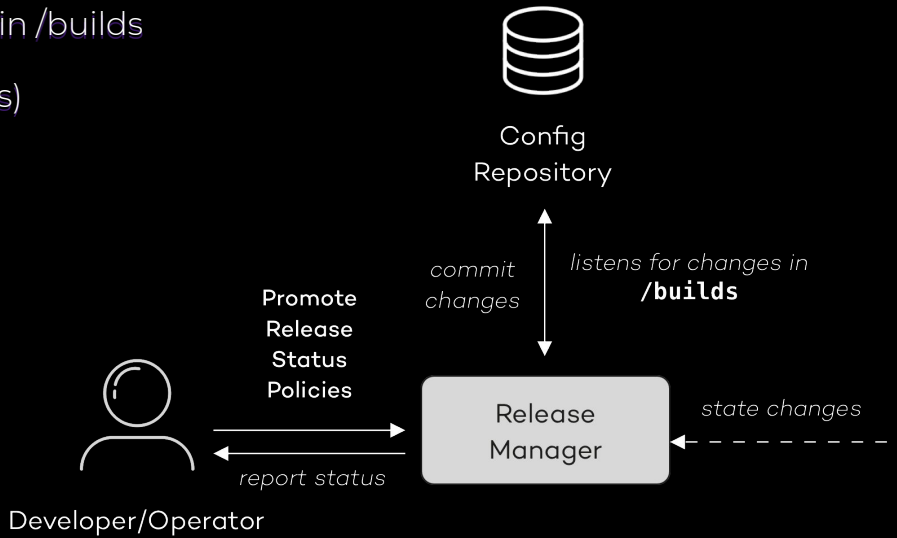
- Shuttle plan information
- CI info and links
- Stage information
  - Test information
  - Security Scan information


```
{
  "id": "master-3b034bc046-8e37ac4819",
  "application": {
    "sha": "3b034bc04638555ee6c3b387c7cbbcc4a499bda4",
    "authorName": "Alexander Kledal",
    "authorEmail": "ark@lunarway.com",
    "committerName": "Alexander Kledal",
    "committerEmail": "ark@lunarway.com",
    "message": "Merged-in-feature-add-loan-product-pull-request-7",
    "name": "lunar-way-product-service",
    "url": "https://bitbucket.org/LunarWay/lunar-way-product-
service/commits/3b034bc04638555ee6c3b387c7cbbcc4a499bda4",
    "provider": "BitBucket"
  },
  "ci": {
    "jobUrl": "https://jenkins.dev.lunarway.com/job/bitbucket/job/lunar-way-product-
service/job/master/104/display/redirect",
    "start": "2019-03-12T12:28:15.246291263+01:00",
    "end": "2019-03-12T12:29:37.951013992+01:00"
  },
  "shuttle": {
    "plan": {
      "sha": "8e37ac48194e5cf0fa85ba23b7090c9ccb0908fe",
      "message": "dont-commit-if-gitops-is-not-enabled",
      "url": "git://git@bitbucket.org:LunarWay/lw-shuttle-go-plan.git"
    }
  },
  "stages": [
    {
      "id": "build",
      "name": "Build",
      "data": {
        "dockerVersion": "17.06.2-ce",
        "image": "quay.io/lunarway/product",
        "tag": "master-3b034bc046-8e37ac4819"
      }
    },
    {
      "id": "snyk-code",
      "name": "Security Scan - Code",
      "data": {
        "vulnerabilities": {
          "high": 0,
          "low": 0,
          "medium": 0
        }
      }
    },
    {
      "id": "test",
      "name": "Test",
      "data": {
        "results": {
          "failed": 0,
          "passed": 166,
          "skipped": 0
        },
        "url": "https://jenkins.dev.lunarway.com/job/bitbucket/job/lunar-way-product-
service/job/master/104/display/redirect"
      }
    },
    {
      "id": "push",
      "name": "Push",
      "data": {
        "dockerVersion": "17.06.2-ce",
        "image": "quay.io/lunarway/product",
        "tag": "master-3b034bc046-8e37ac4819"
      }
    }
  ]
}
```

Release Manager

The responsibility of Release Manager is to control promotion of artifacts between environments, and enable developers/operators to easily operate their services.

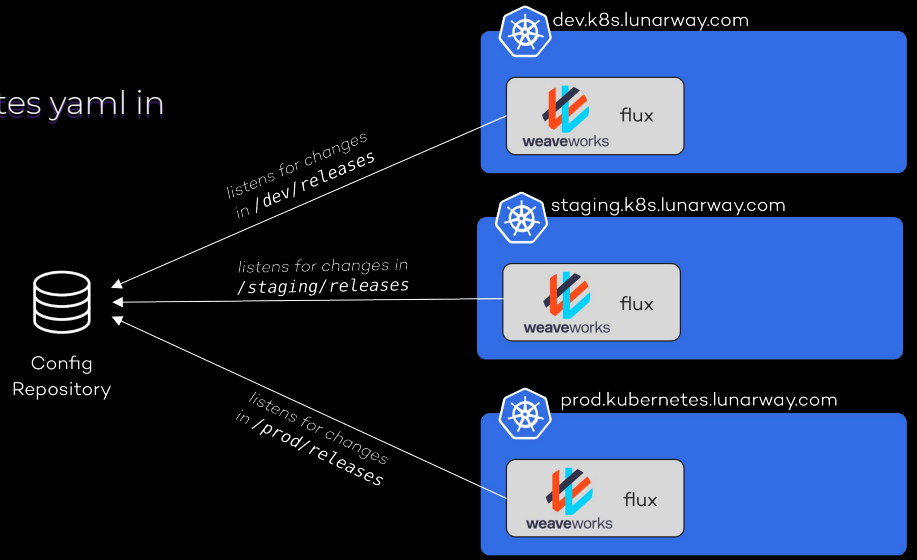
- Listen for changes in a git repository, e.g. in /builds
- Move files between folders (environments)
- Report state changes back to clients



Release Operator

The responsibility of Release Operator is to ensure that the environment is synchronized with the configuration repository

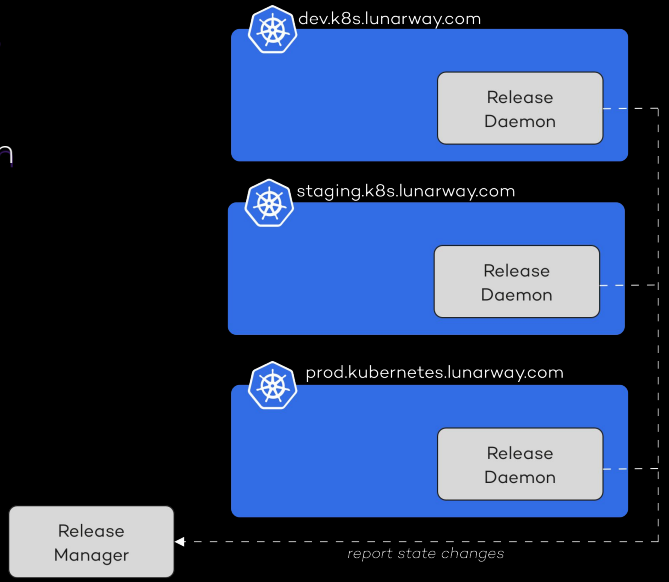
- Weaveworks flux (OSS)
- Listens for changes and applies kubernetes yaml in the cluster it is deployed



Release Daemon

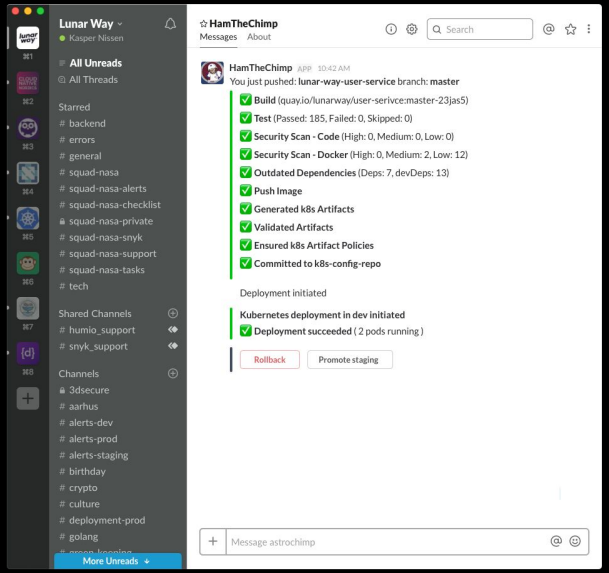
The responsibility of the Release Daemon is to communicate changes in the environment back to the release manager. Both successful releases and failures.

- Listens for changes in the kubernetes environment it is running in
- Reports the state back, and extract relevant information



Developer Workflow

The goal is to provide our developers with the information and tooling they need. Minimizing the feedback loop.



Slack Integration

```
# Assumes that the service is running in the "default" environment namespace.
# Assumes master branch as default branch to release
# Optional arguments:
# '--namespace logging' - to support services running in other namespaces.
$ hamctl promote --service authentication --env staging
  [✓] Promoting master-18c0a8 to k8s.staging.lunarway.com
  [✓] 2 pods successful deployed to k8s.staging.lunarway.com
        > authentication-8449fd9f5-76ltr
        > authentication-8449fd9f5-9lxnv

# List the current status of the deployments for the given service
$ hamctl list --service authentication

  k8s.dev.lunarway.com:
    - image-tag: master-18c0a8
    - author: kni@lunarway.com
    - message: "bug fix for authentication middleware and base image changed to alpine"
    - link: https://bitbucket.org/LunarWay/lunar-way-authentication-service/commits/18c0a84d5c5d5db87b6aef5fc0762844c56ad90a
    - vulnerabilities: 0 high, 1 medium, 0 low

  k8s.staging.lunarway.com:
    - image-tag: master-327ac7
    - author: bso@lunarway.com
    - message: "implemented new authentication model for lunar way supporters"
    - link: https://bitbucket.org/LunarWay/lunar-way-authentication-service/commits/327ac7ae0e4728dee624722fd6fdbfc8c9ad1016
    - vulnerabilities: 5 high, 18 medium, 136 low

  kubernetes.prod.lunarway.com:
    - image-tag: master-327ac7
    - author: bso@lunarway.com
    - message: "implemented new authentication model for lunar way supporters"
    - link: https://bitbucket.org/LunarWay/lunar-way-authentication-service/commits/327ac7ae0e4728dee624722fd6fdbfc8c9ad1016
    - vulnerabilities: 5 high, 18 medium, 136 low
```

CLI

Wrapping up, what did we get?

Adopting a modified GitOps pattern has improved a lot of the previous pain points

- Faster lead time to production
- Promotion of artifacts instead of code (reproducible builds)
- Current state of the cluster is always stored in git and can easily be recreated

Wrapping up, what did we get?

Continued

- Minimizing the need for developer access to the environments (minimize the risk of exposing credentials and human errors)
- CI no longer have access to the cluster, minimizing the surface of attack
- Audit trail of changes

Thank You!

we are hiring https://jobs.lunarway.com/