From DevOps to GitOps

Supercharging Your Kubernetes Workload Deployments

Dan Skaggs

OCTOBER 27-29
Raleigh, NC USA

https://tinyurl.com/skaggs-ato24

This Talk…

Is *not*

- A Kubernetes master class
- A step-by-step “How To” session

Is

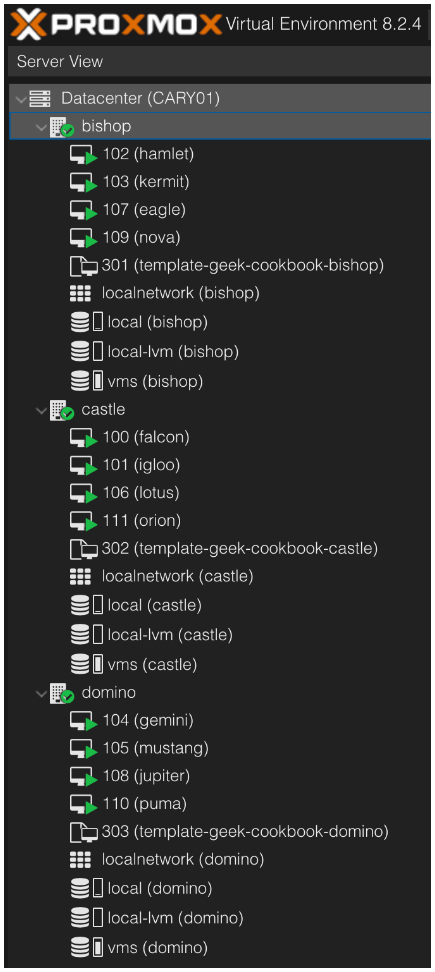
- Introduction to GitOps / Infrastructure as Code (IaC)
- Discussion of software engineering processes useful to IaC
- Discussion of tooling used in the GitOps process
- (Hopefully) Thought-provoking

Who Am I?

- Software engineer turned manager
- US Air Force veteran
- Life-long geek
- I *LOVE* automating things
- Very interested in self-hosting

BLOG https://dan.skaggsfamily.us

in https://www.linkedin.com/in/danskaggs/



PROXMOX Virtual Environment 8.2.4

Server View

```
Datacenter (CARY01)
  bishop
    102 (hamlet)
    103 (kermit)
    107 (eagle)
    109 (nova)
    301 (template-geek-cookbook-bishop)
    localnetwork (bishop)
    local (bishop)
    local-lvm (bishop)
    vms (bishop)
  castle
    100 (falcon)
    101 (igloo)
    106 (lotus)
    111 (orion)
    302 (template-geek-cookbook-castle)
    localnetwork (castle)
    local (castle)
    local-lvm (castle)
    vms (castle)
  domino
    104 (gemini)
    105 (mustang)
    108 (jupiter)
    110 (puma)
    303 (template-geek-cookbook-domino)
    localnetwork (domino)
    local (domino)
    local-lvm (domino)
    vms (domino)
```

GitOps?!?

What is it and why should I care?

Why GitOps

- Next evolution in automation
- Consistency and repeatability
- One source of truth for configuration of your workloads
- Transparent history of changes
- Peer reviews for quality control and consistency*
- Enforced approval workflows*
- Easily roll back bad changes

* with additional tooling

Deploying

Not *That* Flux

flux

Automates synchronizing cluster state to config files in your configuration source

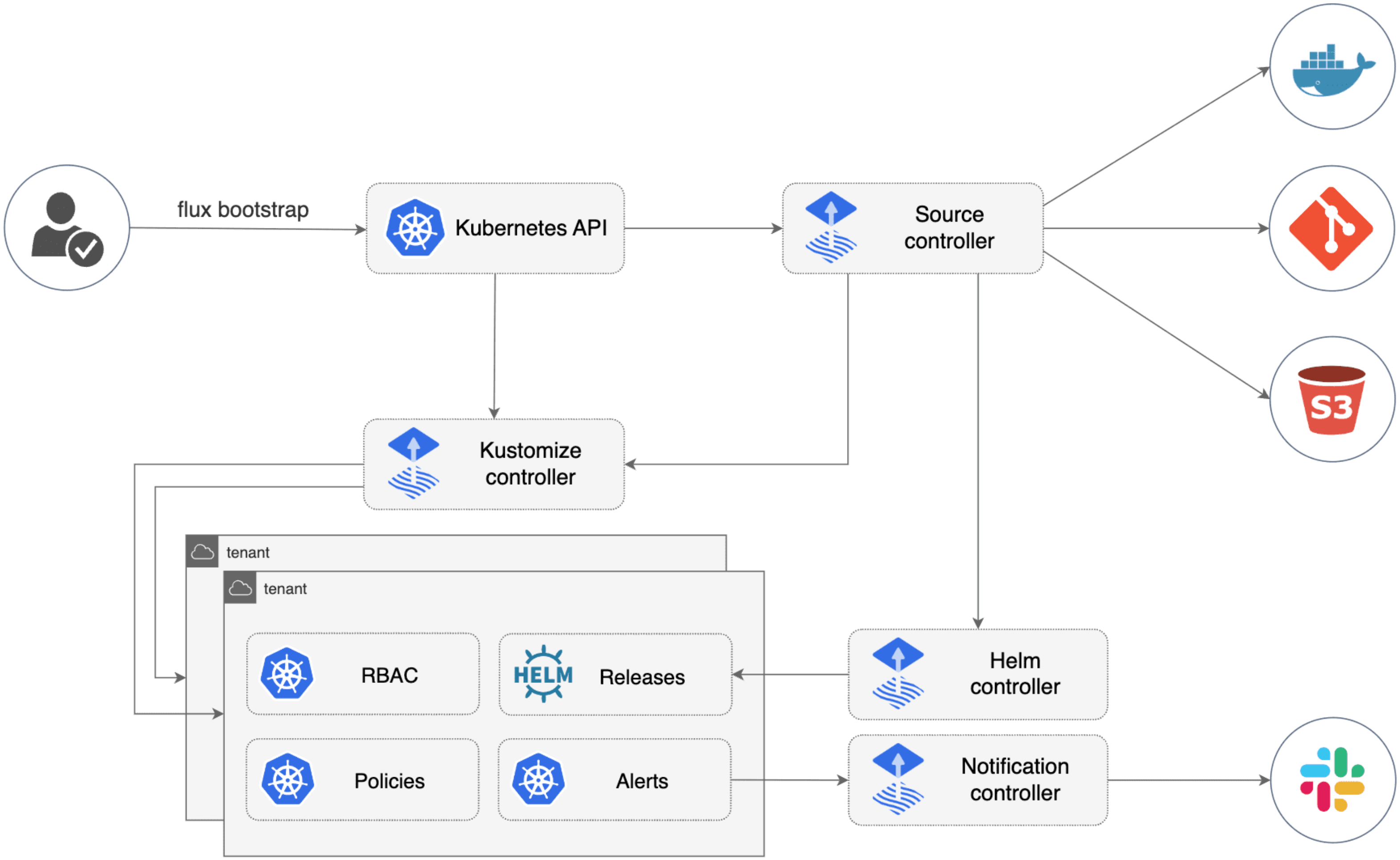
Does not cover build / test phases

Reconciles configuration stored in repo vs the state of the cluster to apply changes

Alternatives

Argo CD

JENKINS X

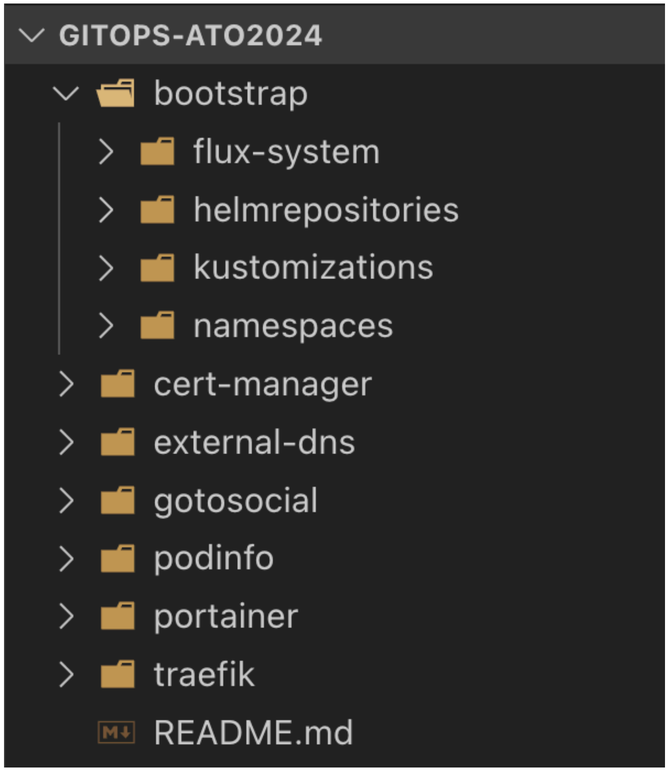


https://fluxcd.io/flux/components/

Organizing Your Repo

Simple Repo Example

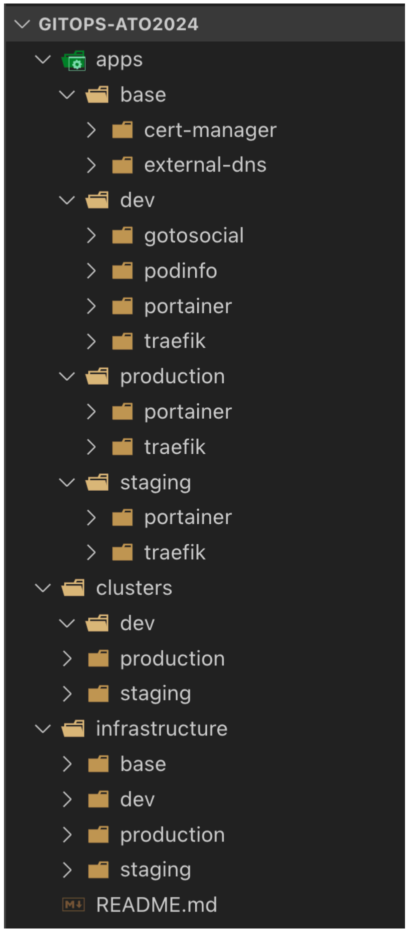
- Perfect for learning, experimenting, and testing
- Will get unwieldy as more workloads are added



```
GITOPS-ATO2024
  bootstrap
    flux-system
    helmrepositories
    kustomizations
    namespaces
  cert-manager
  external-dns
  gotosocial
  podinfo
  portainer
  traefik
  README.md
```

MonoRepo:
Folder Per Environment

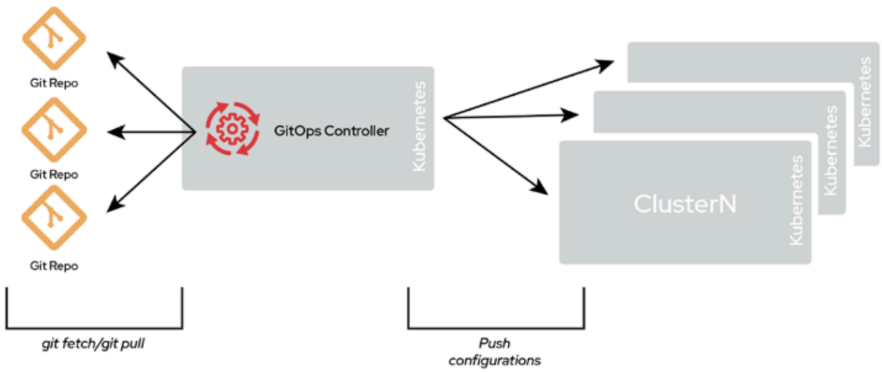
- Good for small to mid-sized teams
- All configuration is in one place and easy to trace
- CODEOWNERS file important for approval workflow
- Reconciliation performance can suffer as repository size increases



```
∨ GITOPS-ATO2024
  ∨ apps
    ∨ base
      › cert-manager
      › external-dns
    ∨ dev
      › gotosocial
      › podinfo
      › portainer
      › traefik
    ∨ production
      › portainer
      › traefik
    ∨ staging
      › portainer
      › traefik
  ∨ clusters
    ∨ dev
    › production
    › staging
  ∨ infrastructure
    › base
    › dev
    › production
    › staging
    README.md
```

Multi Repo:
Repo Per Environment

- Easier to restrict permissions for each environment
- Promoting changes between environments requires more effort
- Able to scale to many environments without sacrificing performance



Path to GitOps (Hernandez) p.28

So, Software Engineering…

Infrastructure as Code with git

Allows parallel distributed updates

Commit messages provide a detailed audit trail of changes

Allows system documentation to live with configuration files

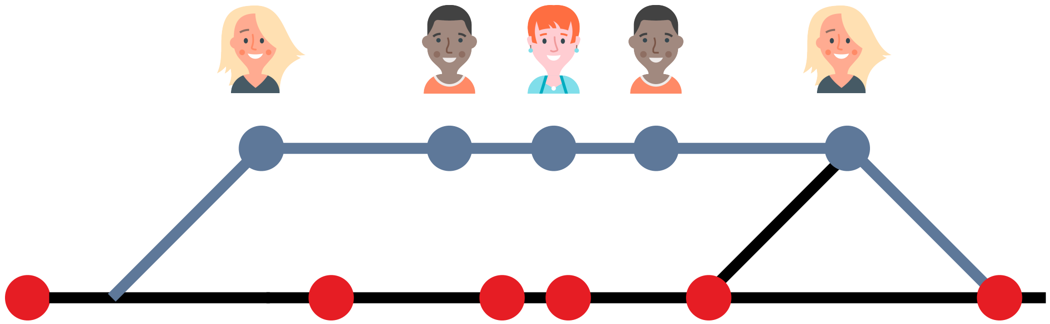
Encourages collaboration and approvals through pull requests*

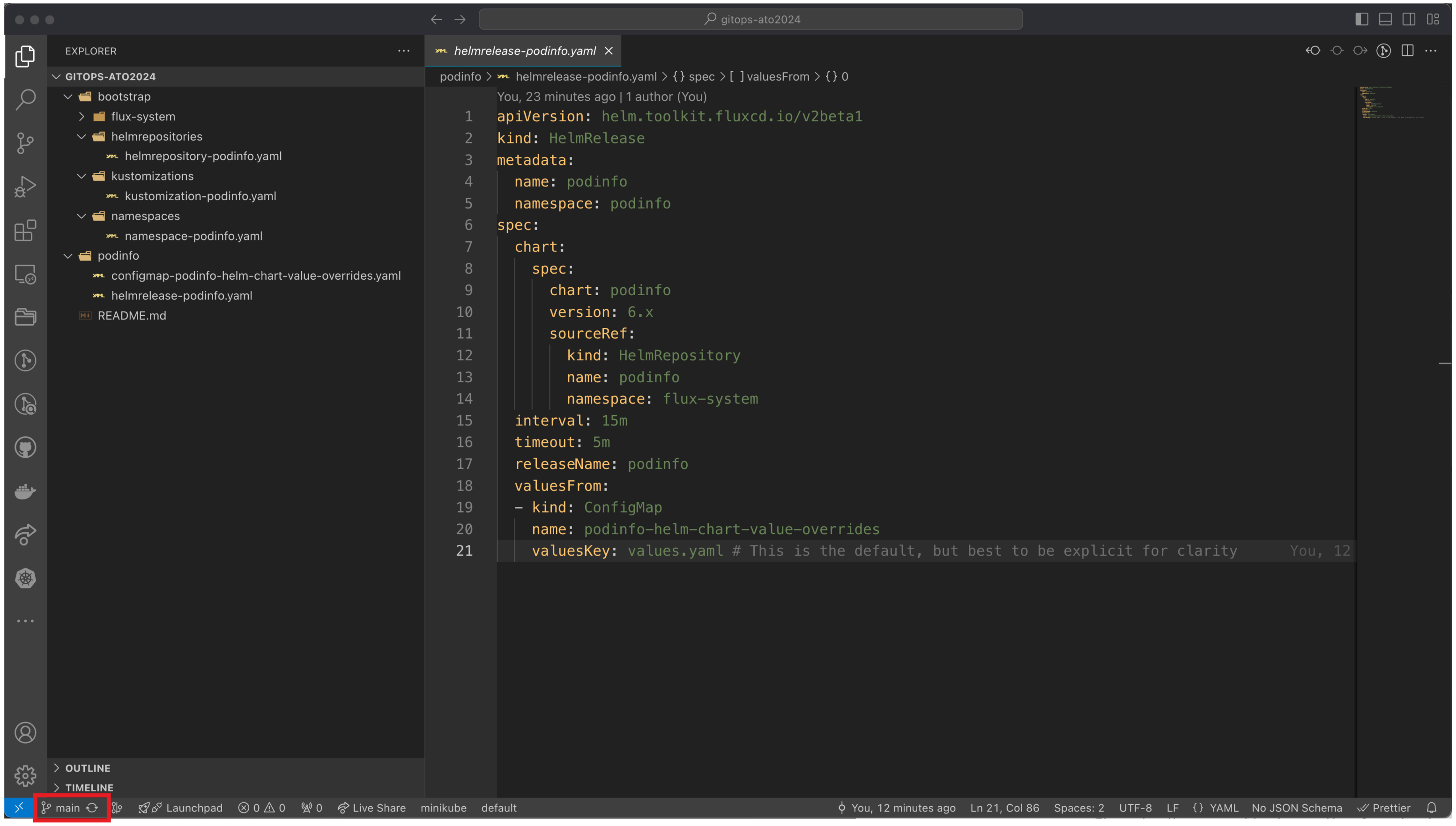
* with GitHub, Gitlab, etc

Branching

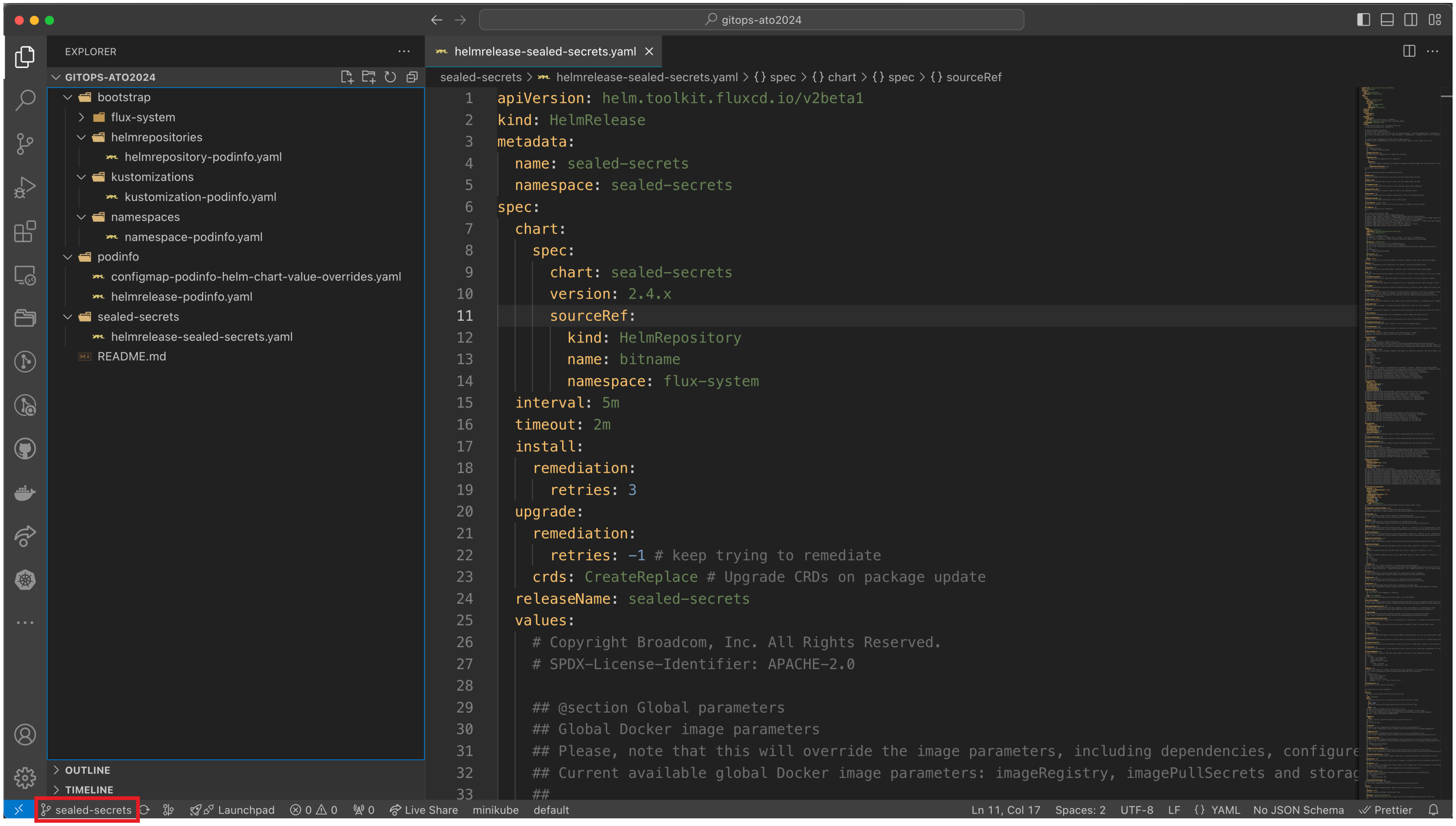
Allows work to happen in a space separate from the main code line

Allows others to continue their work without being affected by your work





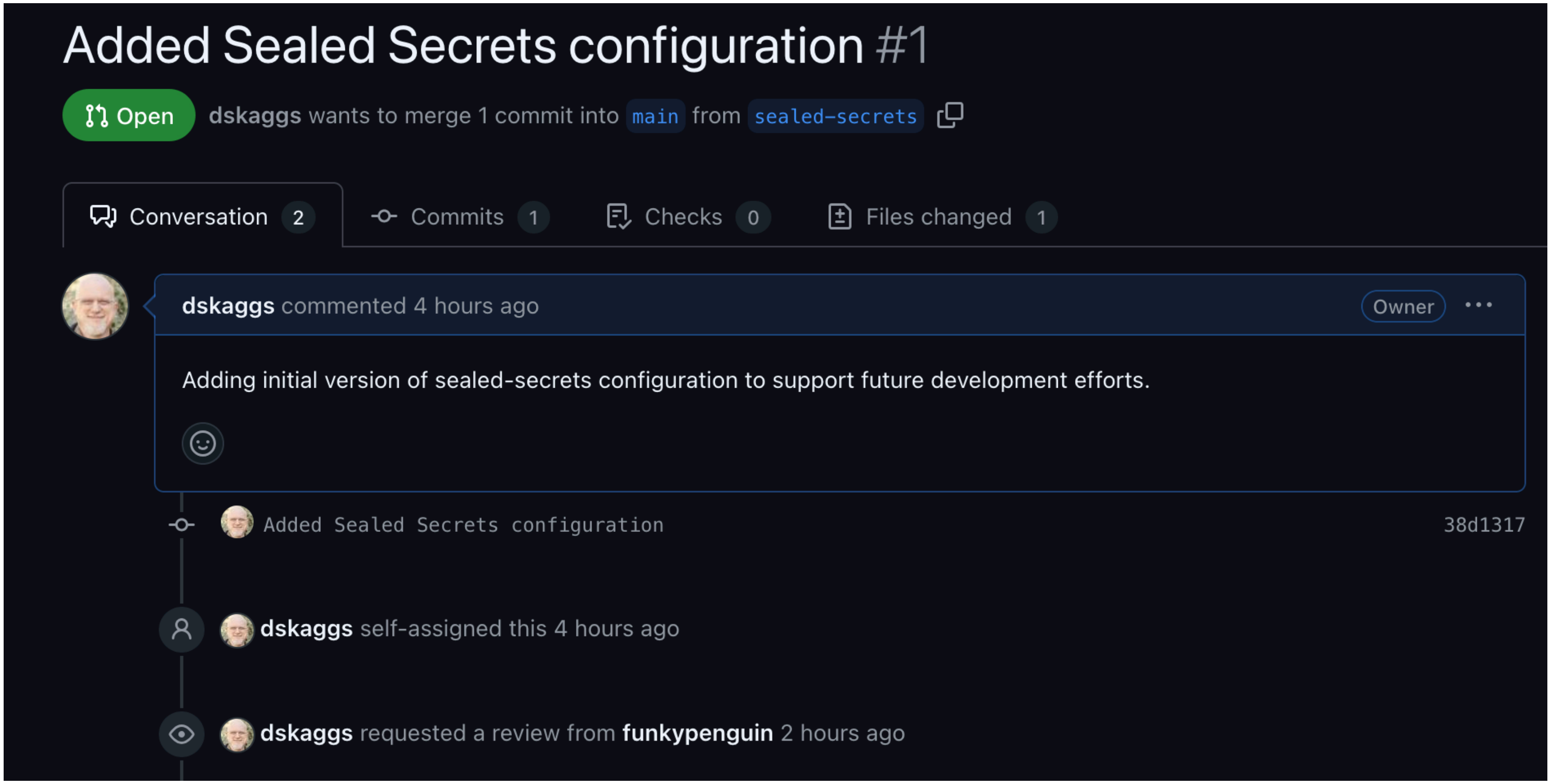
```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: podinfo
  namespace: podinfo
spec:
  chart:
    spec:
      chart: podinfo
      version: 6.x
      sourceRef:
        kind: HelmRepository
        name: podinfo
        namespace: flux-system
  interval: 15m
  timeout: 5m
  releaseName: podinfo
  valuesFrom:
  - kind: ConfigMap
    name: podinfo-helm-chart-value-overrides
    valuesKey: values.yaml # This is the default, but best to be explicit for clarity     You, 12
```



```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: sealed-secrets
  namespace: sealed-secrets
spec:
  chart:
    spec:
      chart: sealed-secrets
      version: 2.4.x
      sourceRef:
        kind: HelmRepository
        name: bitname
        namespace: flux-system
  interval: 5m
  timeout: 2m
  install:
    remediation:
      retries: 3
  upgrade:
    remediation:
      retries: -1 # keep trying to remediate
    crds: CreateReplace # Upgrade CRDs on package update
  releaseName: sealed-secrets
  values:
    # Copyright Broadcom, Inc. All Rights Reserved.
    # SPDX-License-Identifier: APACHE-2.0

    ## @section Global parameters
    ## Global Docker image parameters
    ## Please, note that this will override the image parameters, including dependencies, configure
    ## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storag
    ##
```

Pull Requests

- Invite collaboration between engineers to improve quality of solution
- Learning opportunity for all involved
- Key component in change approval workflow
- Feedback *must* be given and taken in a spirit of wanting to make solutions better
- Toxic comments, bullying, etc must not be tolerated



Added Sealed Secrets configuration #1

Open dskaggs wants to merge 1 commit into `main` from `sealed-secrets`

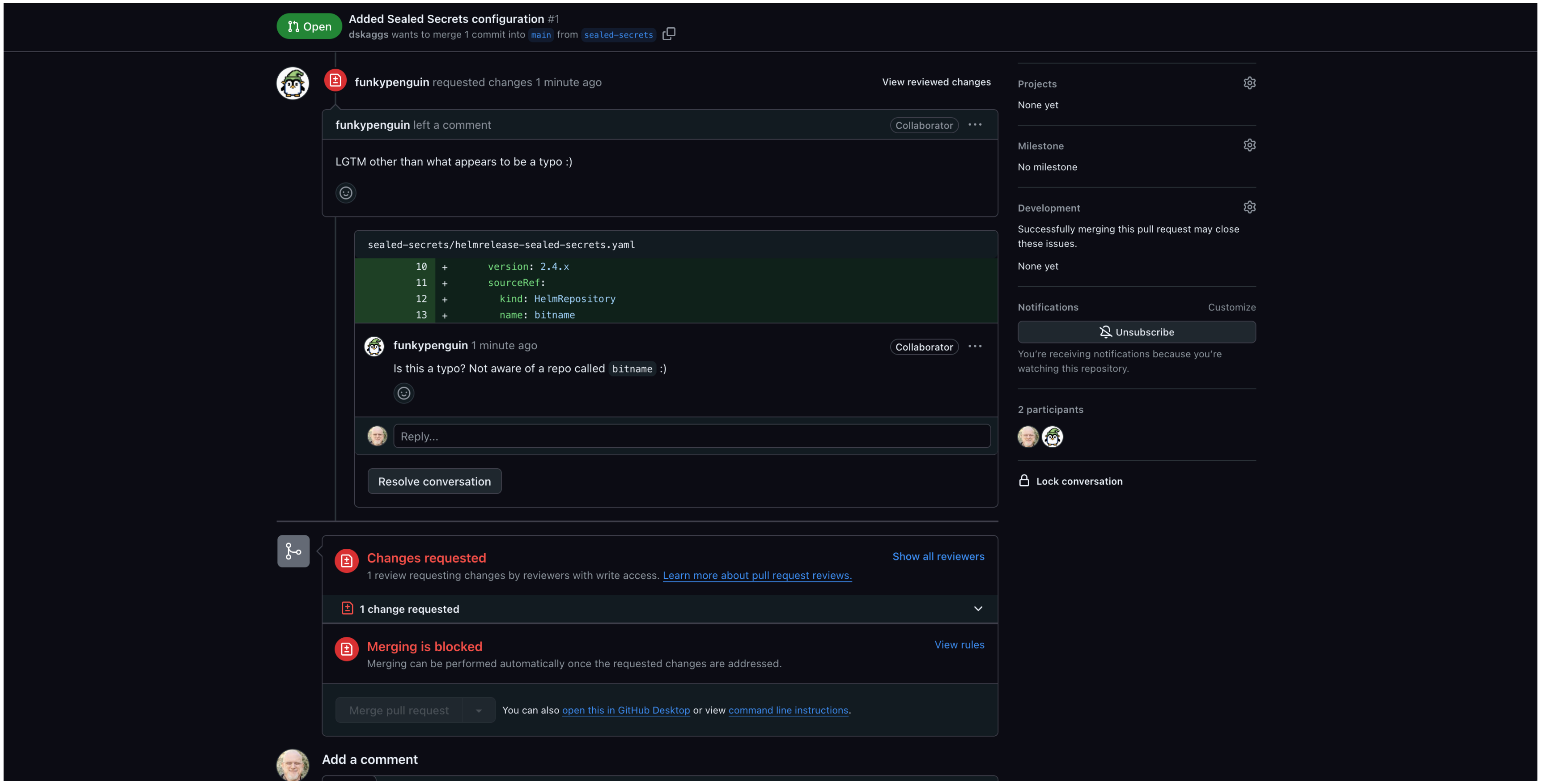
dskaggs commented 4 hours ago

Adding initial version of sealed-secrets configuration to support future development efforts.

`Added Sealed Secrets configuration` 38d1317

dskaggs self-assigned this 4 hours ago

dskaggs requested a review from funkypenguin 2 hours ago



Open **Added Sealed Secrets configuration** #1

dskaggs wants to merge 1 commit into `main` from `sealed-secrets`

funkypenguin requested changes 1 minute ago — View reviewed changes

funkypenguin left a comment — Collaborator

LGTM other than what appears to be a typo :)

sealed-secrets/helmrelease-sealed-secrets.yaml

```
+       version: 2.4.x
+       sourceRef:
+         kind: HelmRepository
+         name: bitname
```

funkypenguin 1 minute ago — Collaborator

Is this a typo? Not aware of a repo called `bitname` :)

Reply...

Resolve conversation

Changes requested

1 review requesting changes by reviewers with write access. Learn more about pull request reviews.

Show all reviewers

1 change requested

Merging is blocked

Merging can be performed automatically once the requested changes are addressed.

View rules

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Add a comment

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues. None yet

Notifications — Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

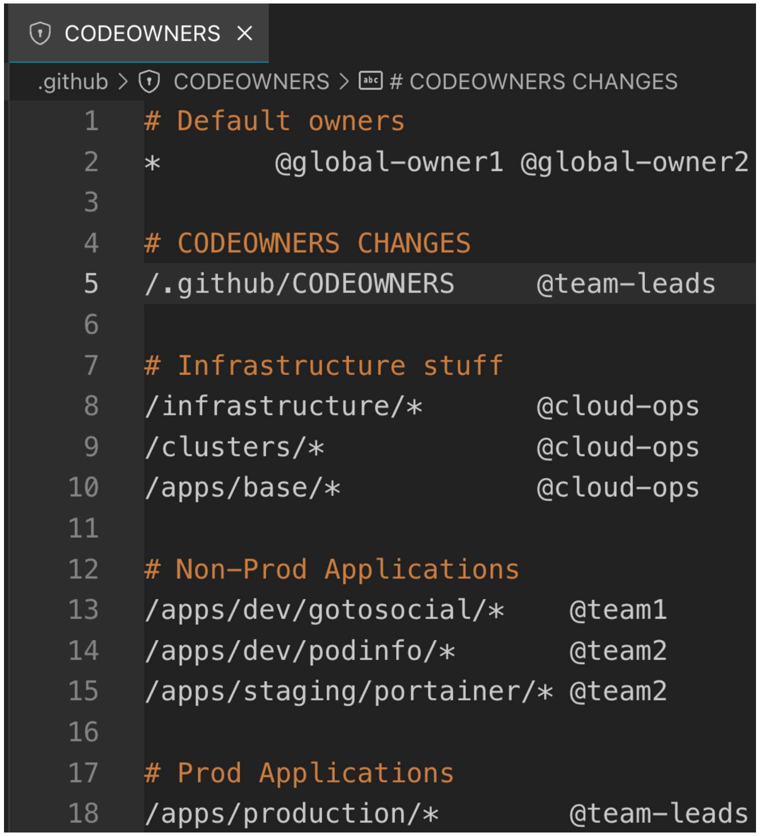
Code Owners

Specifies rules for who must approve changes to various parts of the repository

Can be individual accounts or groups

Combined with branch protection settings (GitHub feature), provides automatic required approvals for pull requests based on changed files

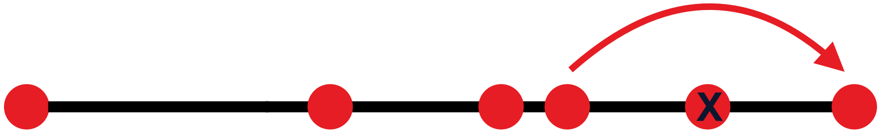
Supported by most Git hosting platforms (GitHub, BitBucket, GitLab, etc)



```
# Default owners
*       @global-owner1 @global-owner2

# CODEOWNERS CHANGES
/.github/CODEOWNERS     @team-leads

# Infrastructure stuff
/infrastructure/*       @cloud-ops
/clusters/*             @cloud-ops
/apps/base/*            @cloud-ops

# Non-Prod Applications
/apps/dev/gotosocial/*    @team1
/apps/dev/podinfo/*       @team2
/apps/staging/portainer/* @team2

# Prod Applications
/apps/production/*        @team-leads
```

Rolling Back

- Errors happen (we’re human)
- Not the same as deployment rollback on failure
- Git revert allows going back in time to the last stable version
- Bad commit still in history for reference
- Highly suggested to “Squash” your PR commits on merge
- Next reconcile reverts to last good configuration



Antipatterns and Pitfalls

- Don’t use long-lived branches for environments
- Don’t store unencrypted secrets in your repo(s)
- Don’t mix infrastructure deployment with application deployment
- Don’t deploy resources by hand with kubectl; trust your IaC
- Beware of monorepo performance as your cluster size grows
- Configure permissions to your repos early and review often

Next Steps

Get “DRY” with Overlays

- Allows reuse of configuration between clusters to improve consistency
- Builds a hierarchical view of how any given cluster is configured
- Larger mental model to understand
- Not as easy for less experienced personnel to work with

Combine All the Above & Profit

- Understand which approach is right for your organization
- Don’t try to eat the entire elephant at once
- Find meaningful progress measurements
- Never stop learning

Acknowledgements

David Young (aka The Funky Penguin)
https://geek-cookbook.funkypenguin.co.nz/

Flux
https://fluxcd.io

“The Path to GitOps”
Christian Hernandez
https://developers.redhat.com/e-books/path-gitops