

Om GitOps og dets rolle i application deployment

Driving It | Kbh | 5/11

Henrik Løvborg

Sr Solutions Architect

Red Hat Danmark





Agenda

Om Red Hat og open source

GitOps

OpenShift GitOps & ArgoCD

GitOps og Application Delivery

Q&A

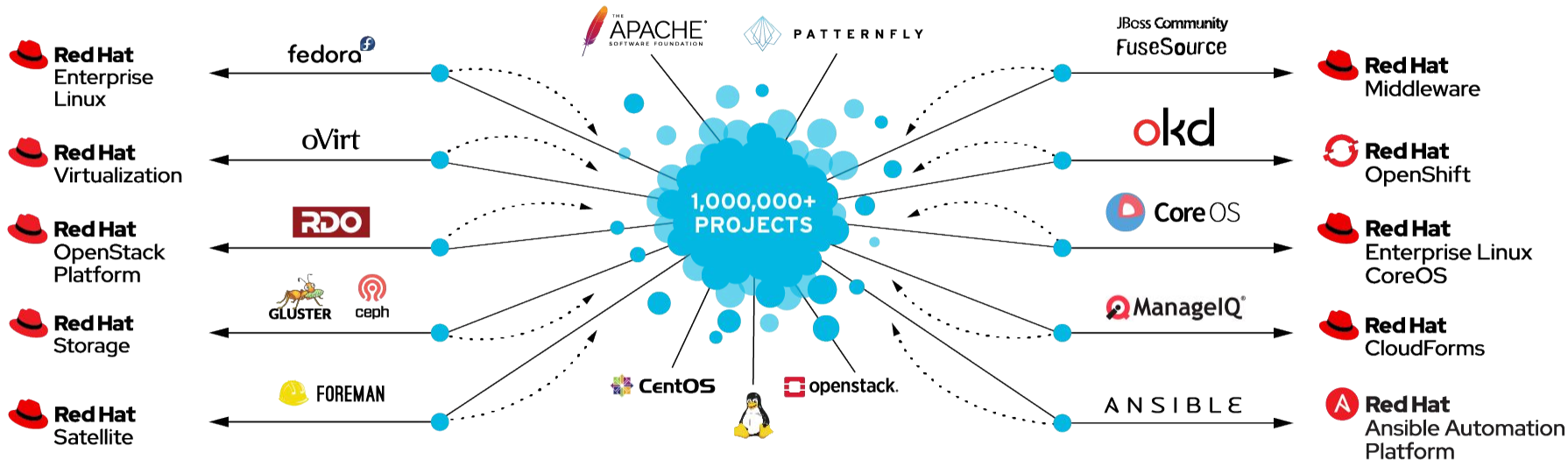


The world's leading developer and supplier of
open source software
for enterprise IT

Offering a comprehensive portfolio of
products and services

Globally active
in all vertical markets, with
strong channel and direct delivery models

From community to enterprise

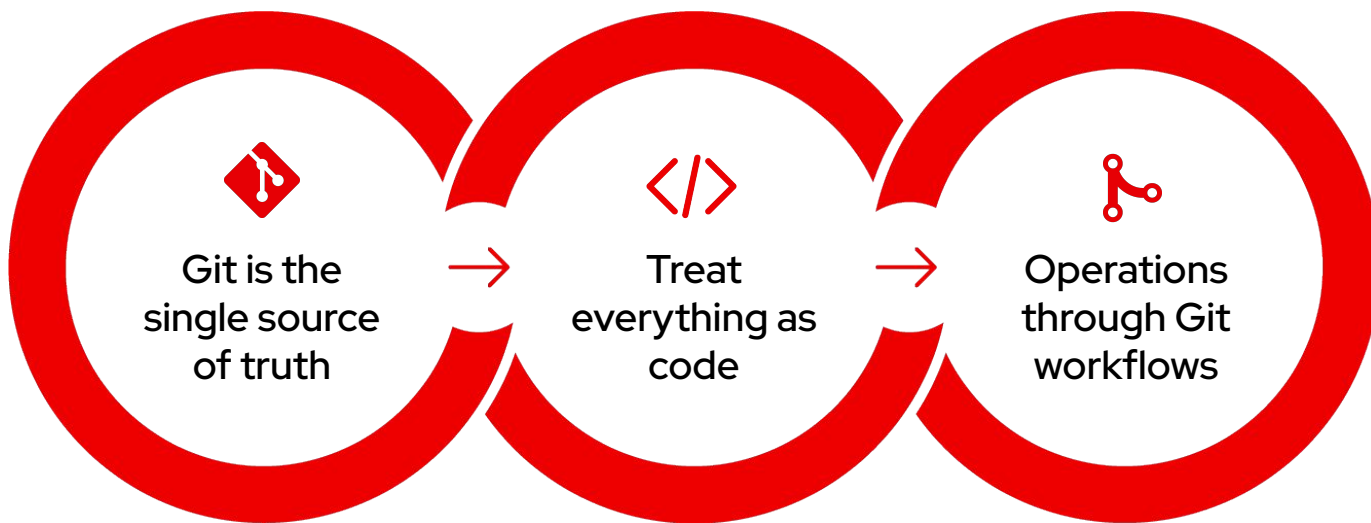


GitOps

5

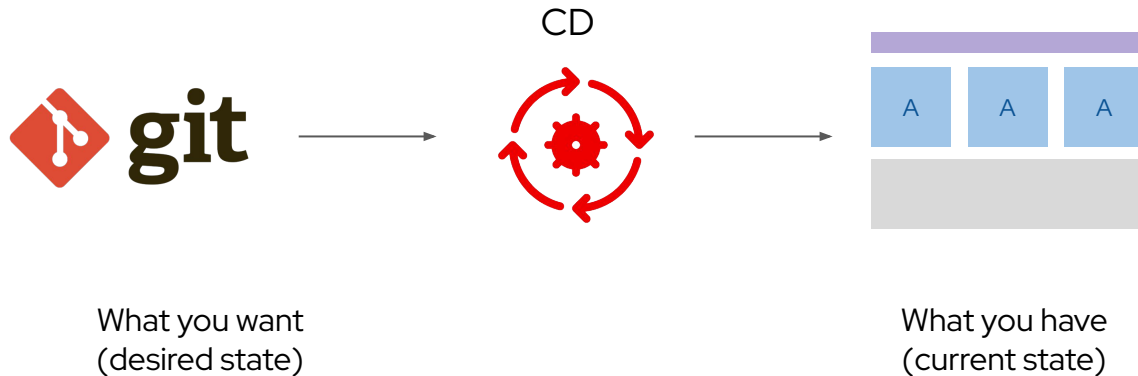
What is GitOps?

An developer-centric approach to Continuous Delivery and infrastructure operation



GitOps Workflow

a declarative approach to application/infra delivery



Why GitOps?

Standard Workflow

Familiar tools and Git workflows from application development teams

Enhanced Security

Review changes beforehand, detect configuration drifts, and take action

Visibility and Audit

Capturing and tracing any change to clusters through Git history

Multi-cluster consistency

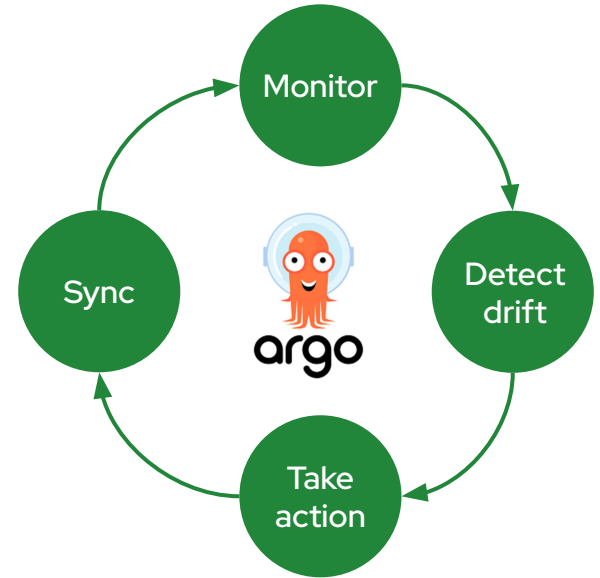
Reliably and consistently configure multiple Kubernetes clusters and deployment

OpenShift GitOps & ArgoCD

9

Argo CD

- Cluster and application configuration versioned in Git
- Automatically syncs configuration from Git to clusters
- Drift detection, visualization and correction
- Granular control over sync order for complex rollouts
- Rollback and rollforward to any Git commit
- Manifest templating support (Helm, Kustomize, etc)
- Visual insight into sync status and history



10

OpenShift GitOps



Multi-cluster config management

Declaratively manage cluster and application configurations across multi-cluster OpenShift and Kubernetes infrastructure with Argo CD



Automated Argo CD install and upgrade

Automated install, configurations and upgrade of Argo CD through OperatorHub



Opinionated GitOps bootstrapping

Bootstrap end-to-end GitOps workflows for application delivery using Argo CD and Tekton with GitOps Application Manager CLI



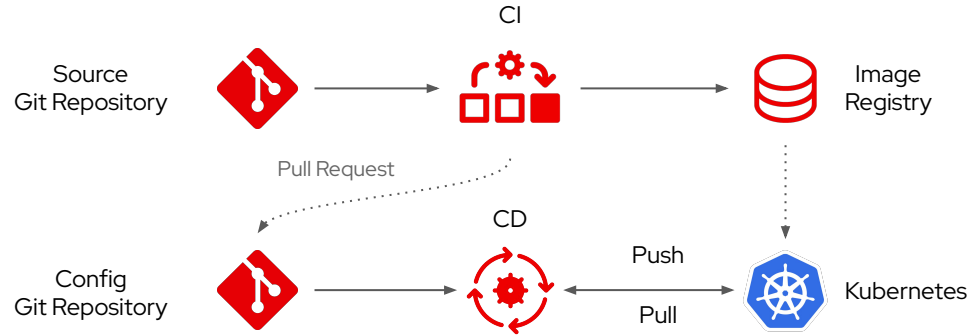
Deployments and environments insights

Visibility into application deployments across environments and the history of deployments in the OpenShift Console

GitOps og application delivery

12

The GitOps Application Delivery Model

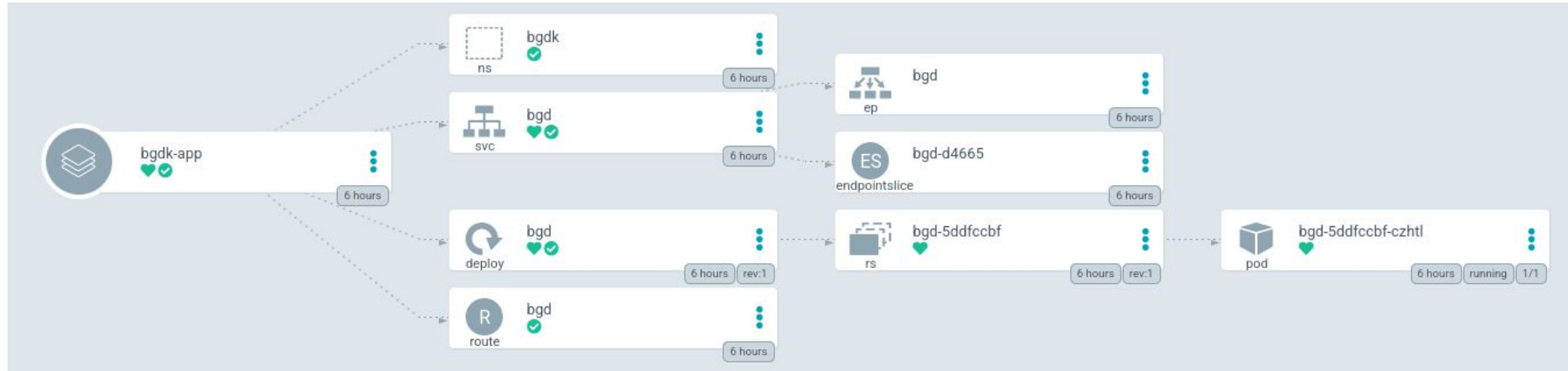


Argo CD Application

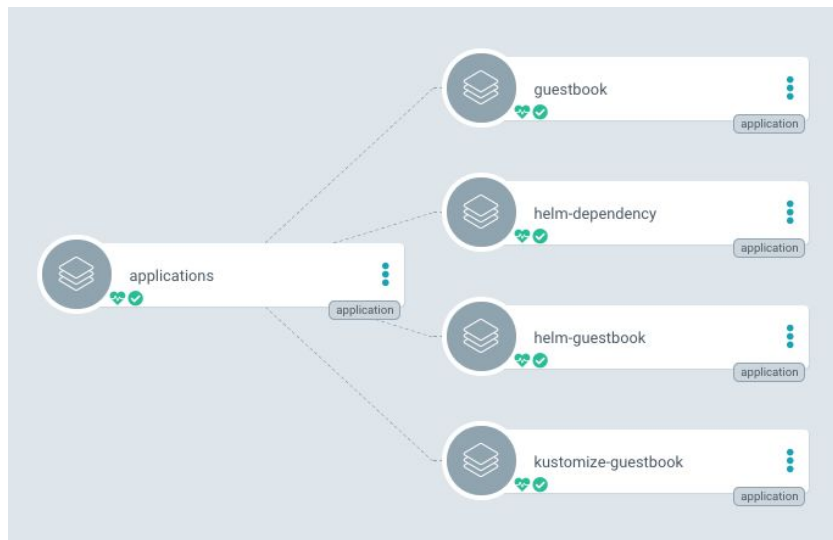
- “Application” is a custom resource provided by Argo CD
- Unit of work within Argo CD
- Logical collection of YAML
- Support for Helm and Kustomize
- Configuration specs
 - Git Repository
 - Revision/Branch
 - What cluster and namespace

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: product-catalog-dev
  namespace: argocd
spec:
  destination:
    namespace: argocd
    server: https://kubernetes.default.svc
    project: product-catalog
  source:
    path: manifests/app/overlays/dev-quay
    repoURL:
      https://github.com/gnunn-gitops/product-catalog.git
    targetRevision: main
  syncPolicy:
    automated:
```

Sample app deployment in ArgoCD



App of Apps Pattern



- Application that deploys Applications
- Method of Bootstrapping
 - Hundreds of Applications can be deployed in “one shot”
- Logically group Applications made up of YAML **and** Helm.
- Watchdog for Applications.

ApplicationSets

- Use a single manifest to target multiple clusters
- Deploy Application from a single or multiple git repos
- Manages lifecycle of the Application CR
- Can be thought of as sort of a “factory” for building Application CRs
 - You still end up with Application definitions, it’s just managed now from a central configuration.

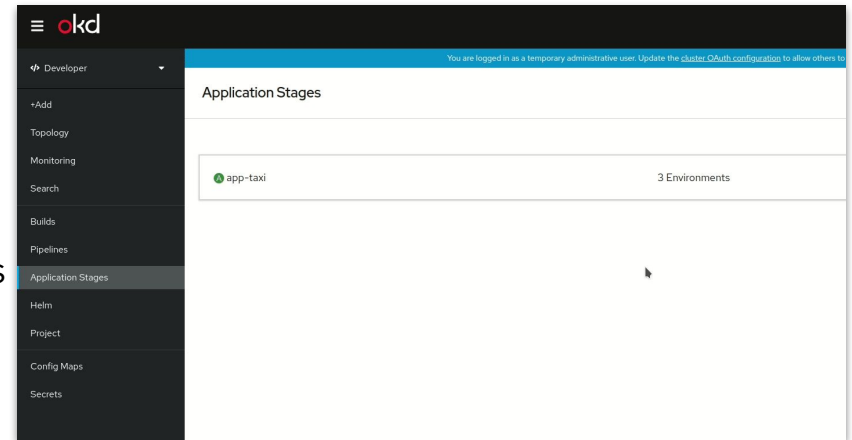


GitOps Application Manager CLI

- Bootstraps Git repos for GitOps
- Configures deployment environments
- Configures webhooks for Tekton Pipelines for CI
- Configures Argo CD for deployment to environments
- Kustomize for environment-specific configs
- Integration with secret managers

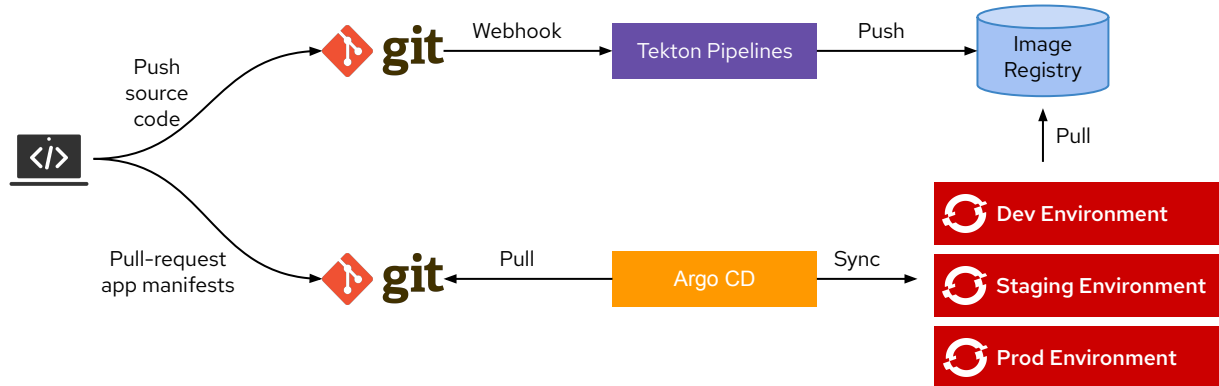
```
$ kam bootstrap
```

```
$ kam environment add stage
```



GitOps Application Manager CLI

```
$ kam bootstrap
```



Working with Kustomize

- Traverses a Kubernetes manifest to add, remove or update configuration options without forking.
- Native feature of kubectl (and by extension oc)
- Purely declarative approach to configuration customization
- Manage an arbitrary number of distinctly customized Kubernetes configurations
- Every artifact that kustomize uses is plain YAML and can be validated and processed as such
- As a "templateless" templating system; it encourages using YAML without forking the repo it.



Q&A og resources

- Topic Pages: [GitOps](#), [Pipelines](#)
- RH Keynote: GitOps Con - [GitOps: The Essential Kubernetes Workflow](#)
- [Learn.OpenShift.com/GitOps](#)
- Red Hat Developers: [Pipelines](#), [GitOps](#)
- GitOps Guide to the Galaxy [Stream](#)
- Learn kubernetes: <https://learnk8s.io/>
- ArgoCD community page: <https://argo-cd.readthedocs.io/>
- KAM cli: <https://github.com/redhat-developer/kam>
- <https://argoproj.github.io/argocon21/>

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat