ArgoCon
NORTH AMERICA

GitOps Safety: Rendering Accurate Argo CD Diffs Directly on Pull Requests

Regina Voloshin

OSS Tech Lead @ Codefresh by Octopus Deploy

Argo CD Maintainer

CNCF Ambassador

Codefresh GitOps by Octopus Deploy

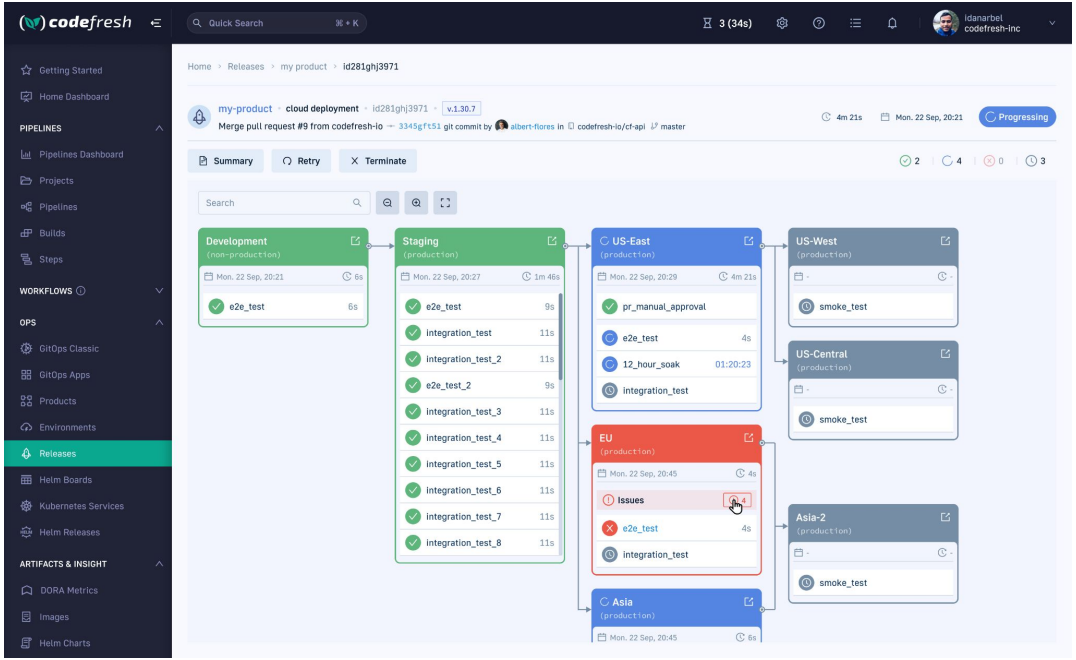
- Argo Control plane and Promotions

  Model app promotions
- Argo Enterprise Support & TAM

  Expert help from Argo Maintainers
- Hardened Security Distro for Argo

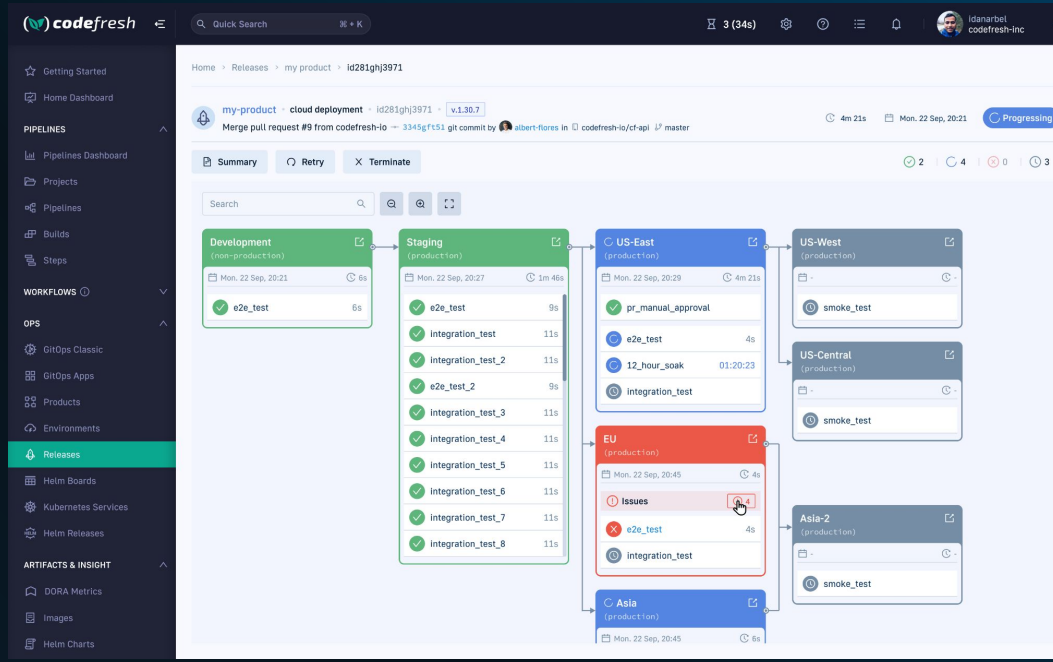
  Aggressive patching SLA form #1 security maintainers
- #1 GitOps Training & Certification

  Over 23k students…



Codefresh GitOps by Octopus Deploy

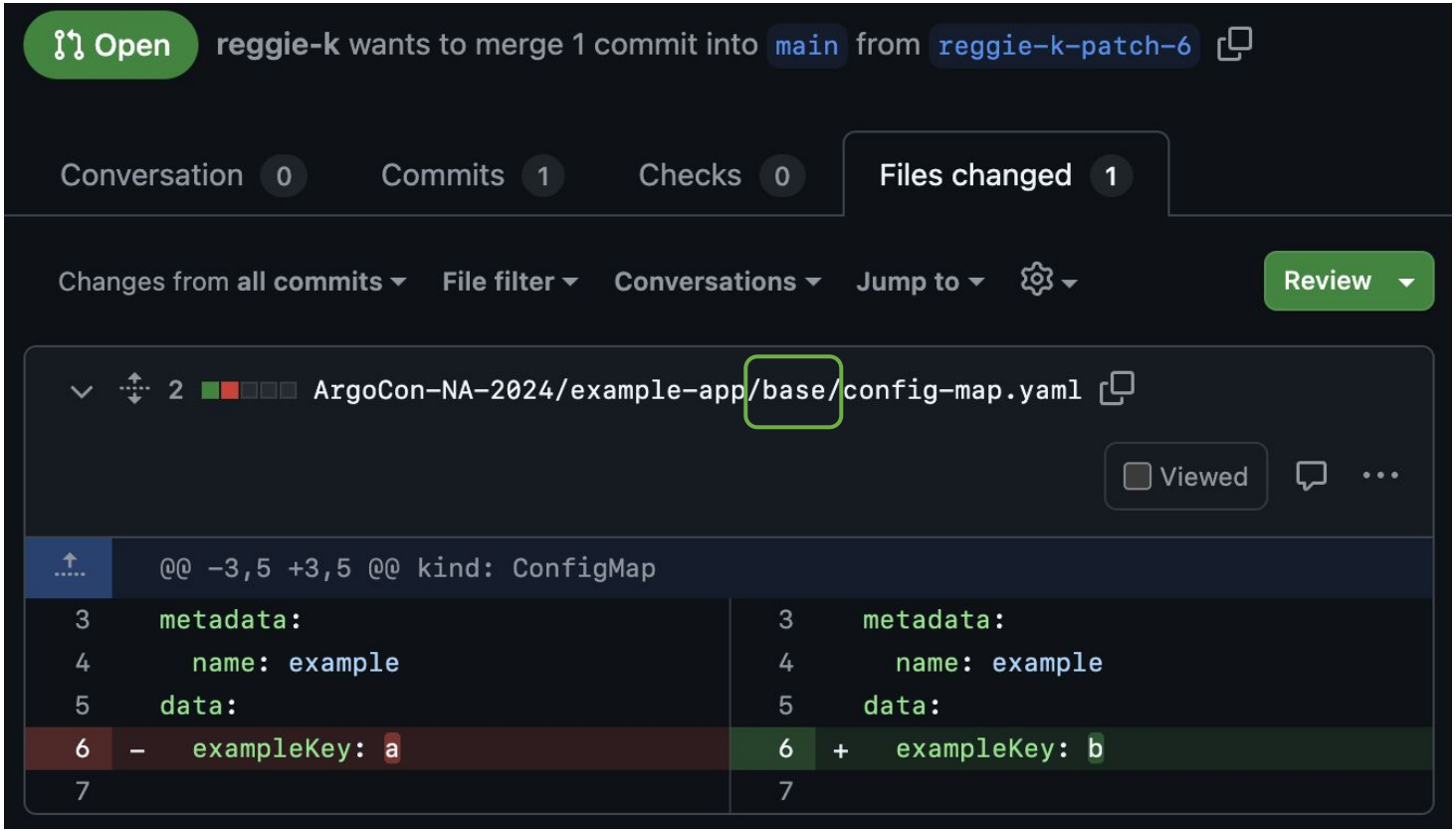
- Argo Control plane and Promotions
  Model app promotions
- Argo Enterprise Support & TAM
  Expert help from Argo Maintainers
- Hardened Security Distro for Argo
  Aggressive patching SLA form #1 security maintainers
- #1 GitOps Training & Certification
  Over 23k students…



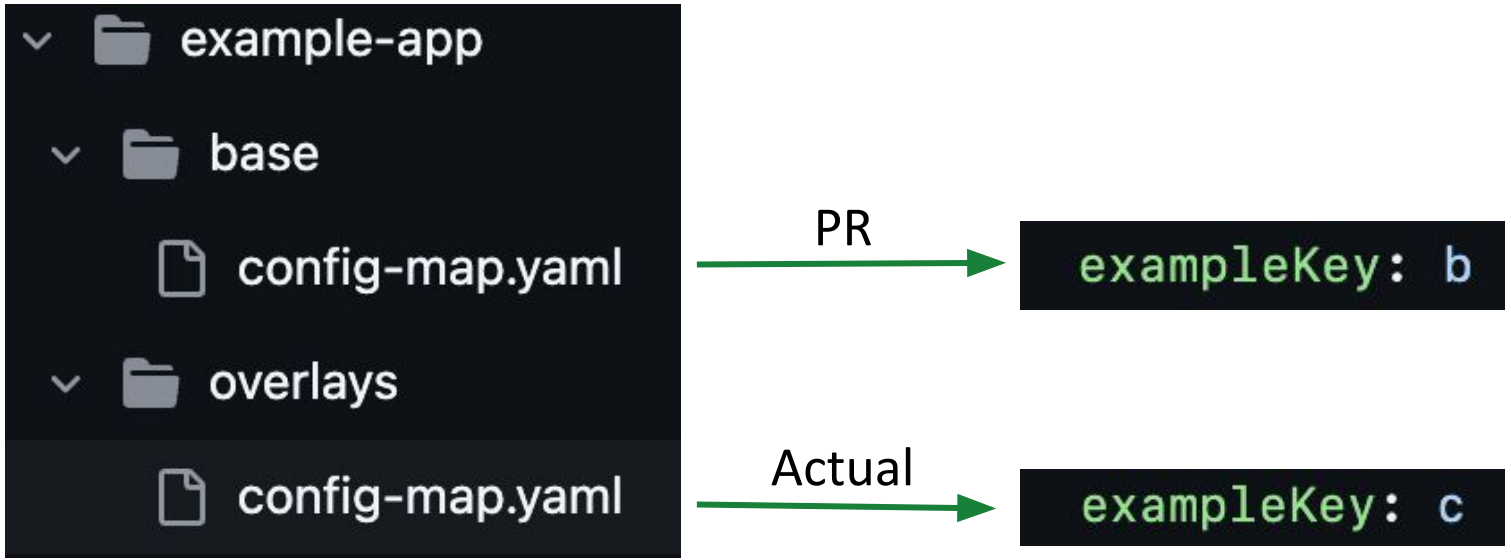
Fun stats

- Other girls in my high school class:
  - 0
- Hours on a plane on my way here:
  - 18
- Times in KubeCon so far:
  - 0
- Number of things I can do in parallel:
  - 1

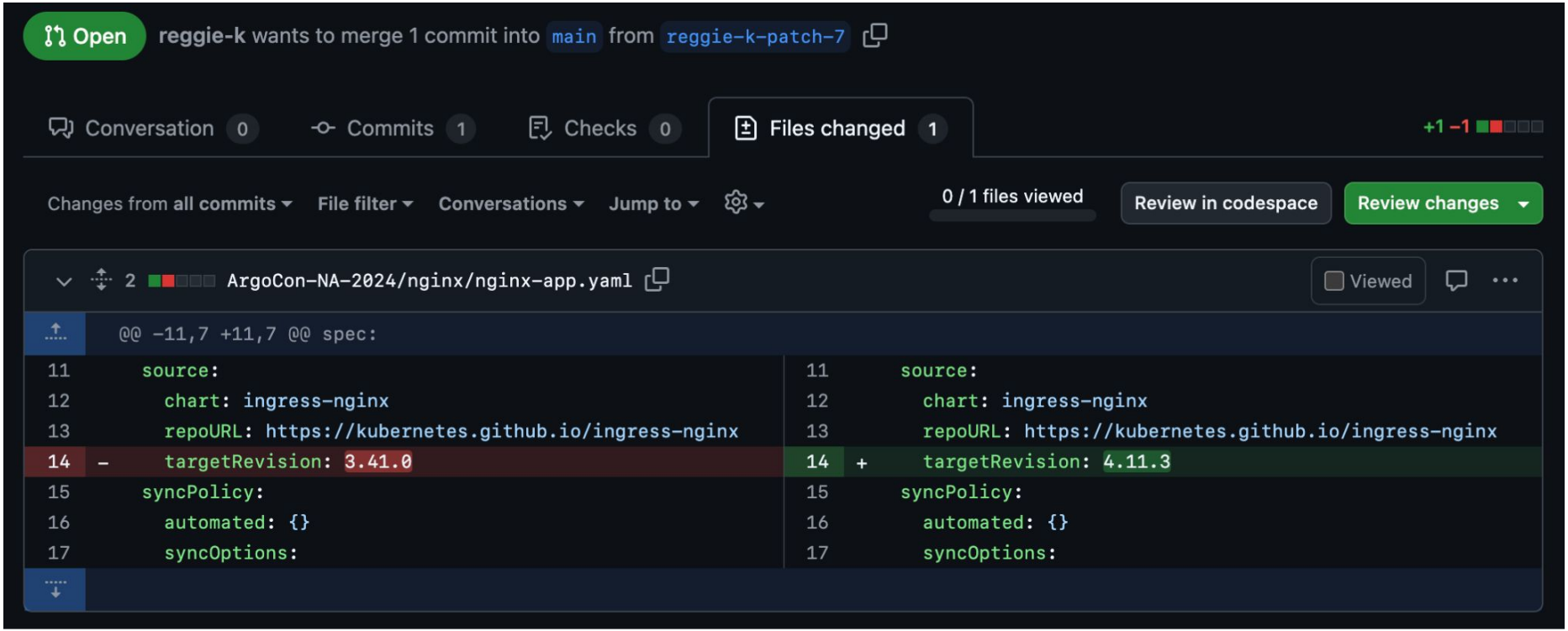
Problem: Kustomize Base Change



Problem: Kustomize Base Change



Problem: Helm Chart Version Change



Needed

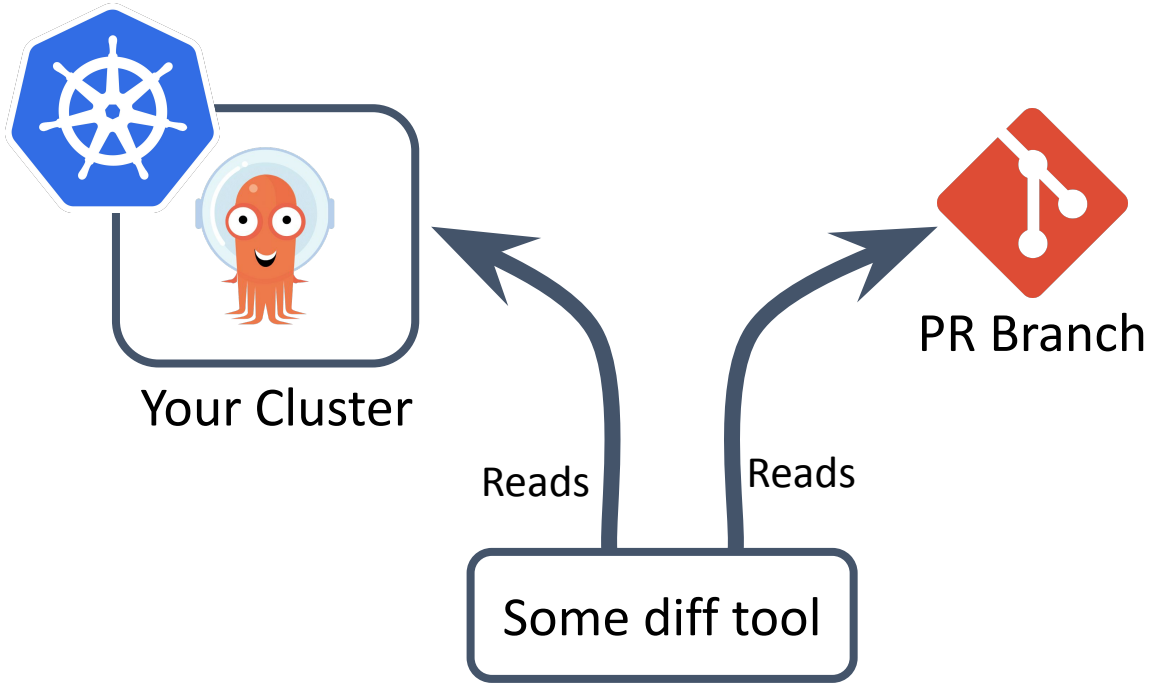
ArgoCon
NORTH AMERICA

Full visibility on the diff!

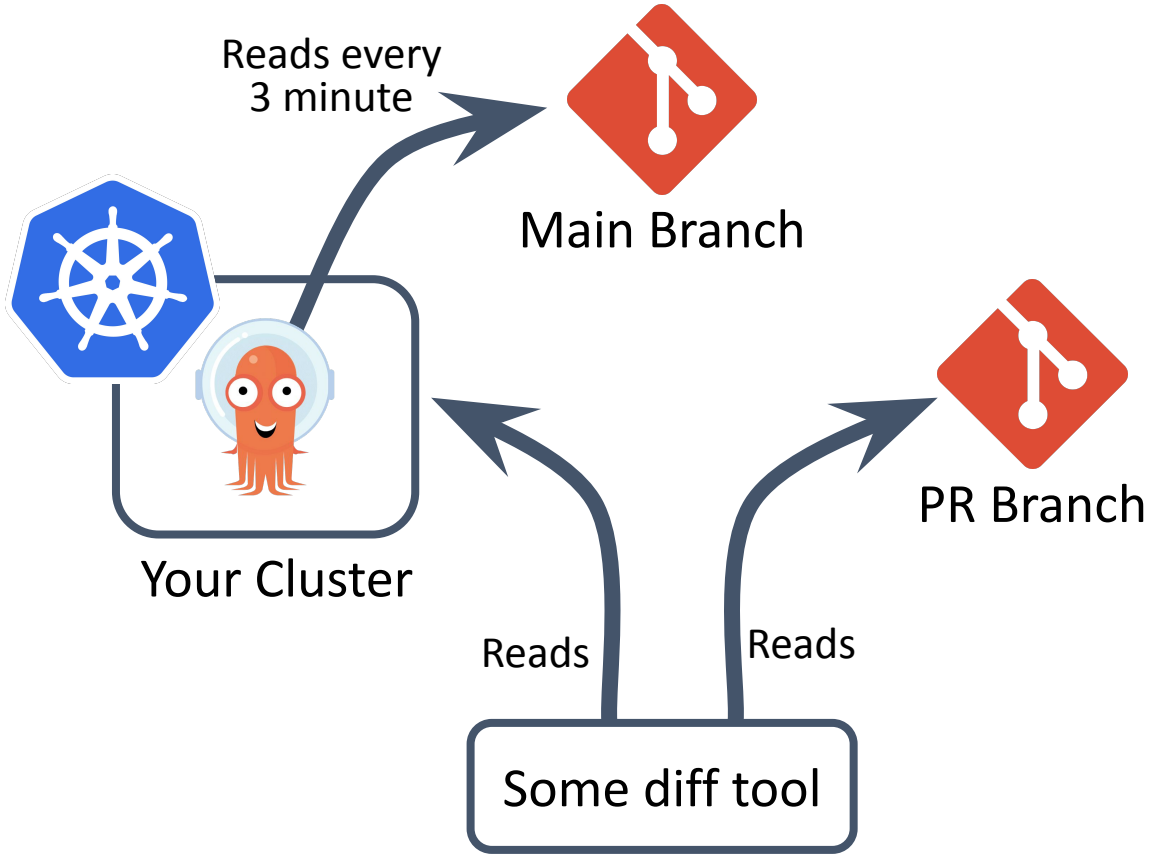
Desired state vs live state

When rendering diffs

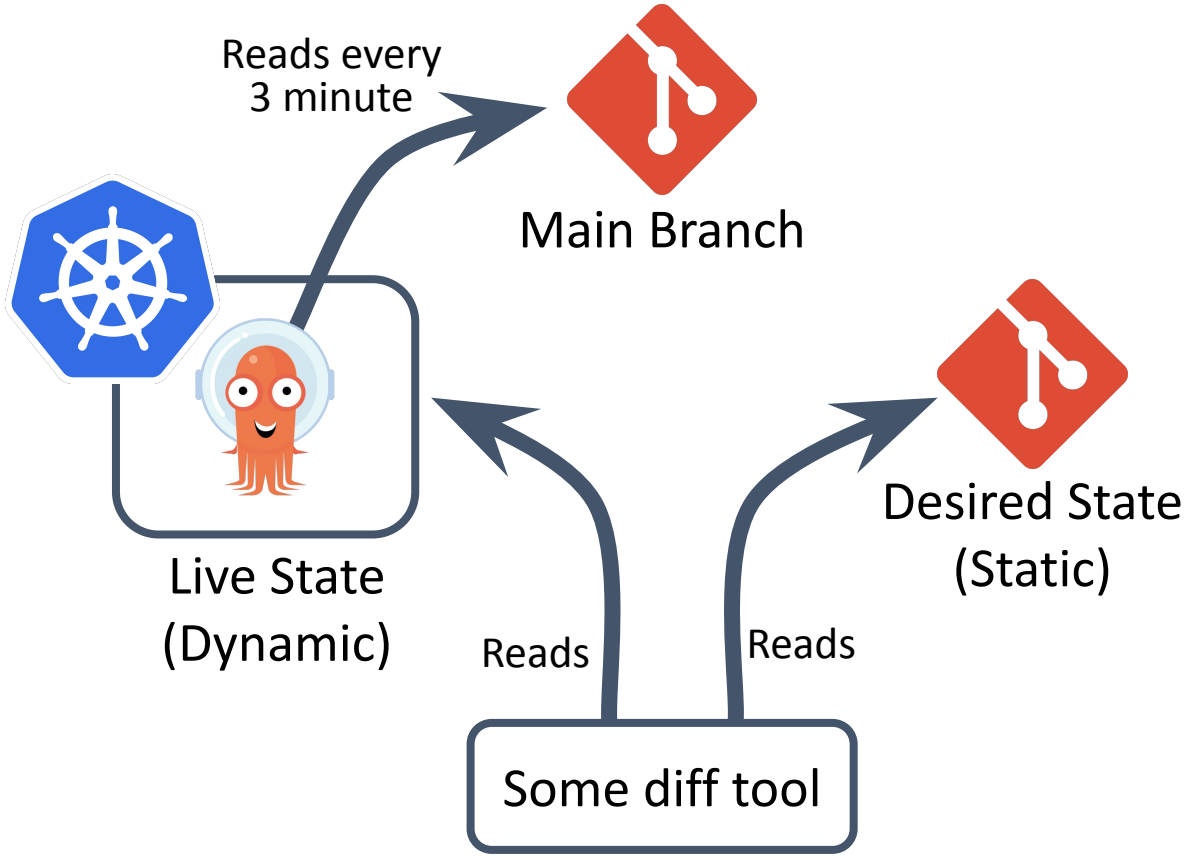
Desired state vs live state



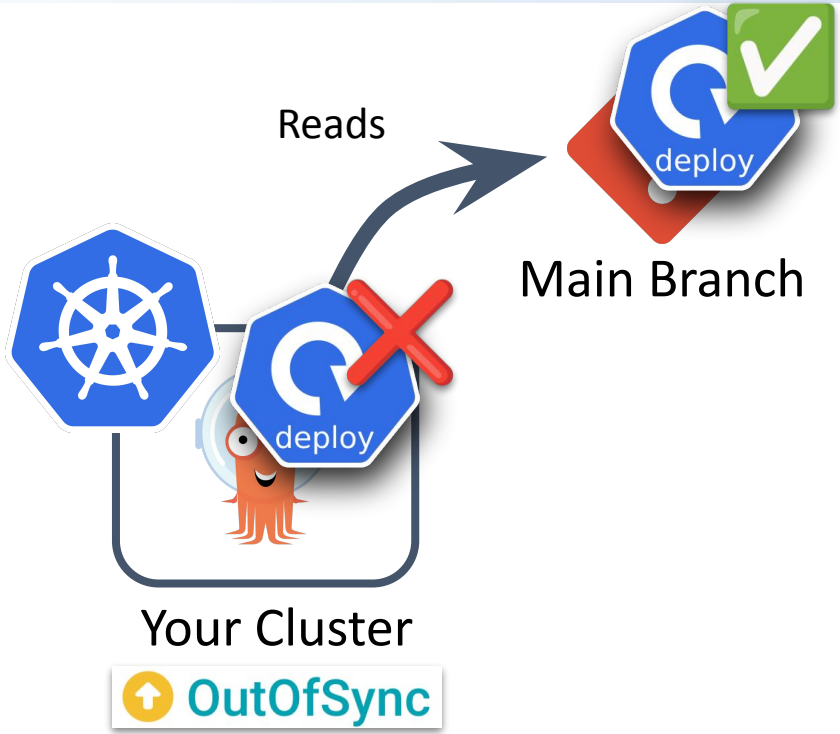
Desired state vs live state



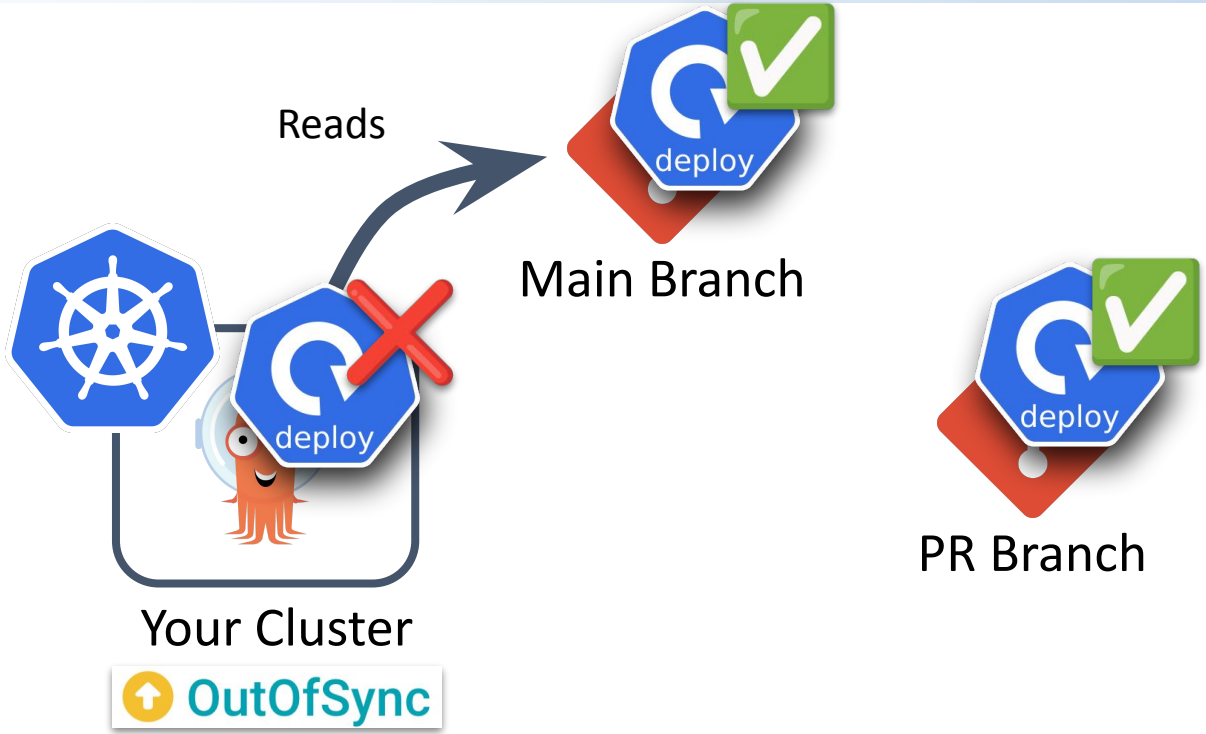
Desired state vs live state



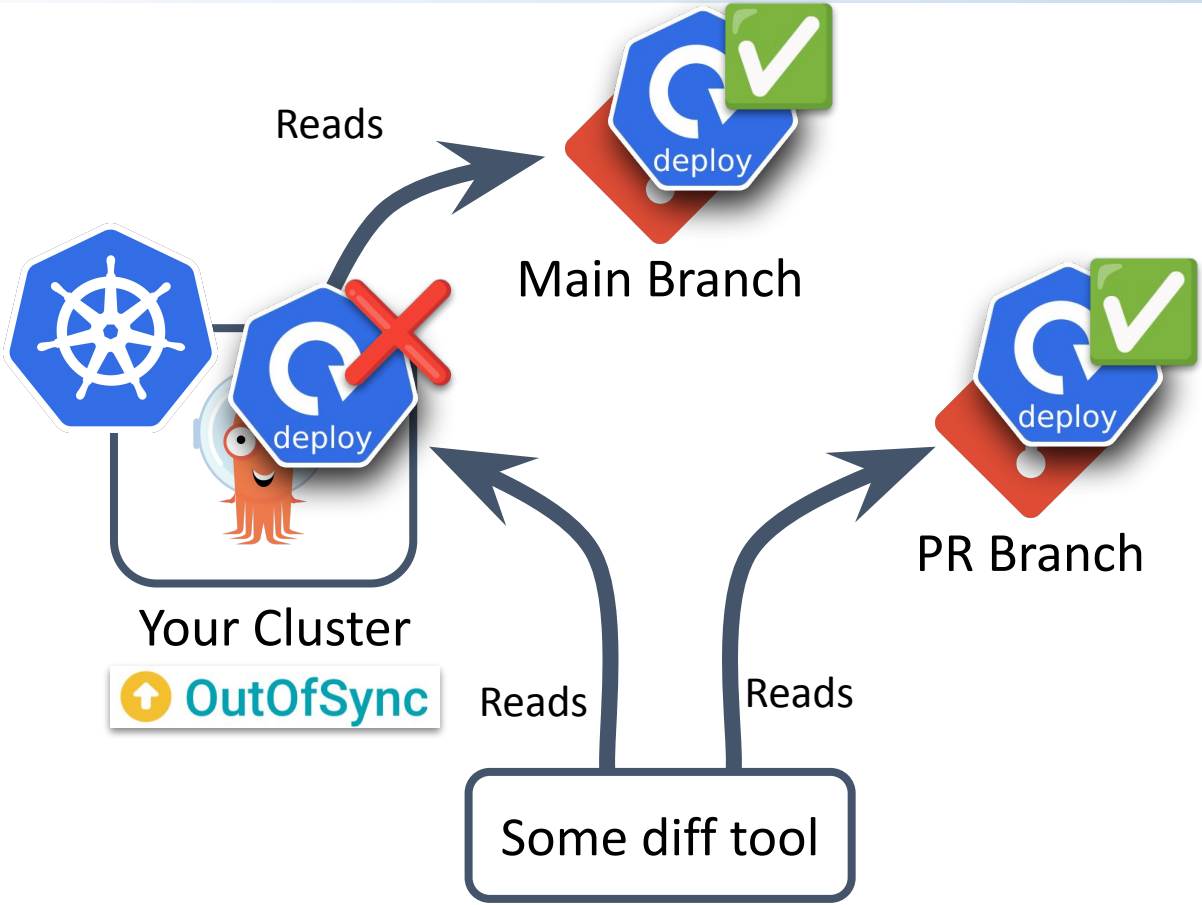
Example



Example

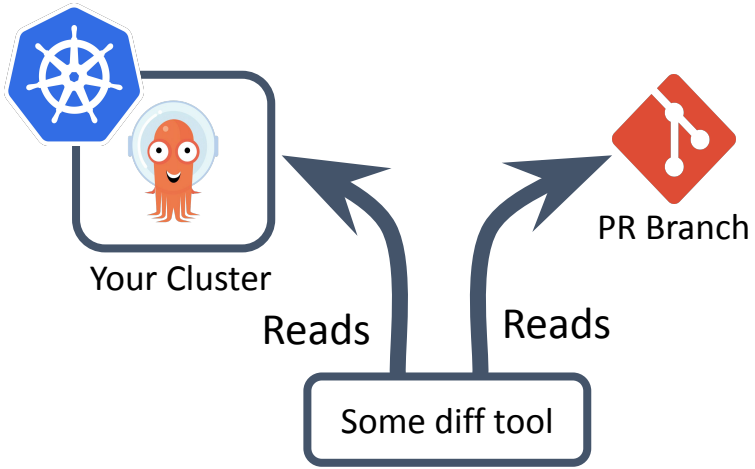

Reads
deploy
Main Branch
deploy
deploy
PR Branch
Your Cluster
OutOfSync

Example

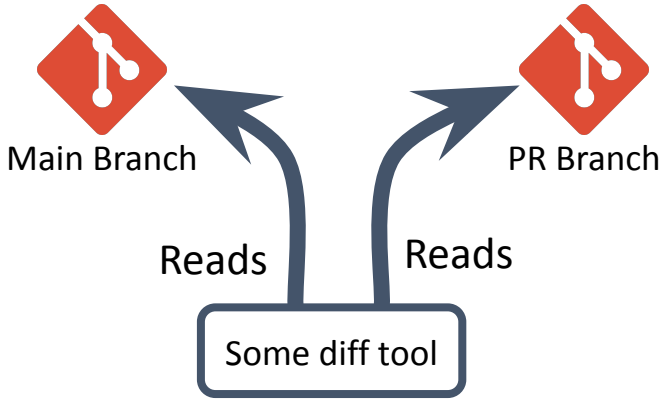

Reads
deploy
Main Branch
deploy
Your Cluster
OutOfSync
deploy
PR Branch
Reads
Reads
Some diff tool

Desired state vs live state

Less predictable



Deterministic



Existing Approaches

For previewing manifests

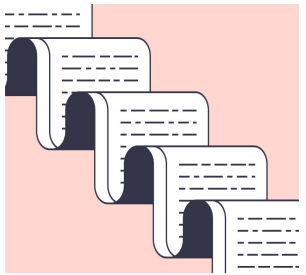
Approach: Rendered Manifests

Works with → Helm and Kustomize

Looks at → fully rendered plain manifests

Enables comparing: desired state to desired state

Link



Tool: Kubechecks

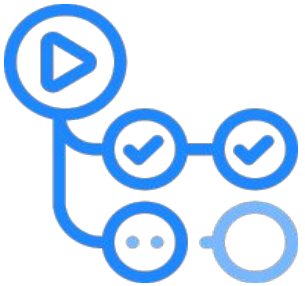
- K8s Deployment
- Clones the repository
- Runs checks
- Comments results on the PR
- Compares: desired state to live state

Link

Tool: ArgoCD Diff Action

- GitHub action
- Runs the `argocd diff` command
- Compares: desired state to live state
- Isn’t actively maintained lately

Link



Tool (incubating): ArgoCD Hydrator

Link

Rendered manifests pattern → Integrated

Renders and pushes manifests →

Dag Bjerre Andersen

From Denmark 🇩🇰

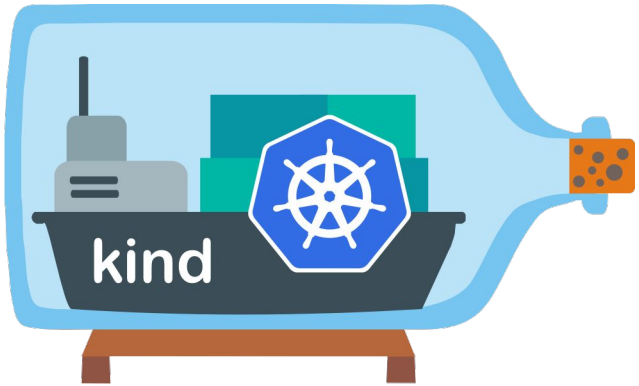
Infrastructure Engineer @ Doubble

Doubble is an app for double dating and meeting new people in groups

Ephemeral Clusters

How can we use ephemeral Kubernetes clusters to rendering your manifests changes directly on your pull requests?

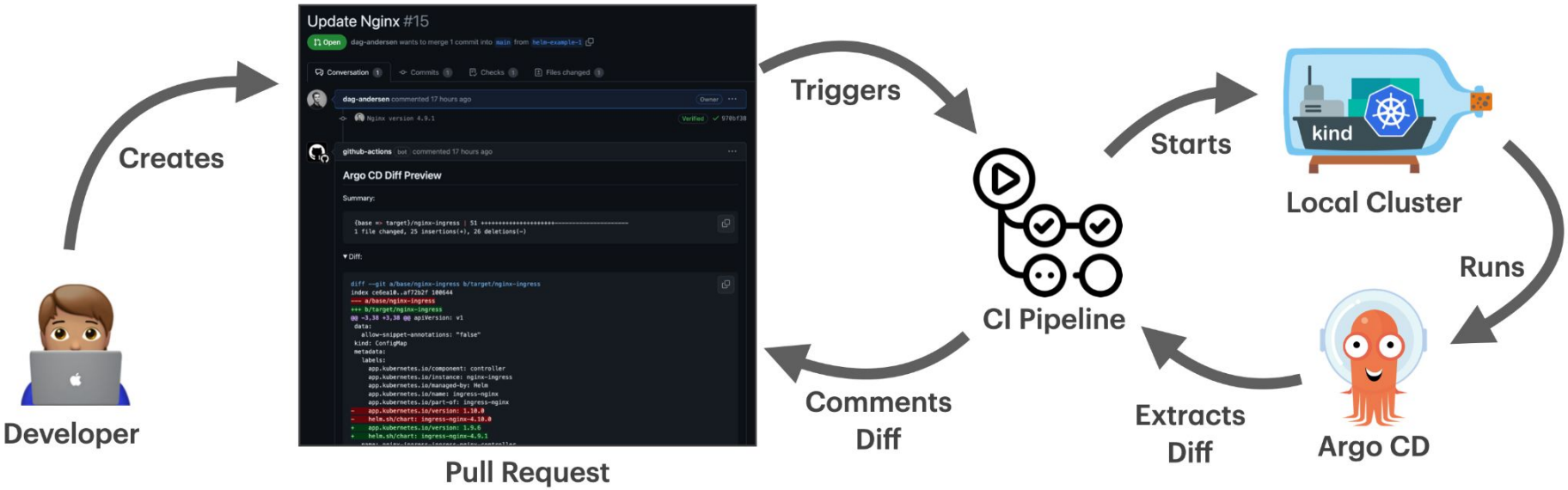
Ephemeral Clusters



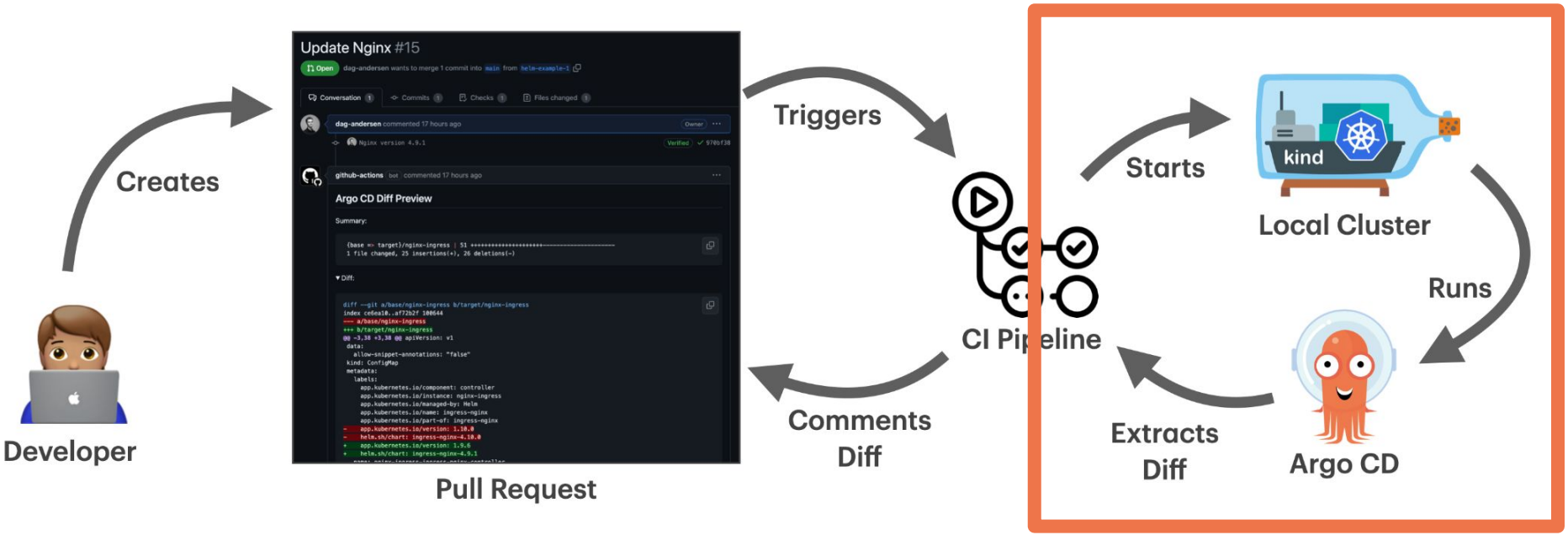
kind

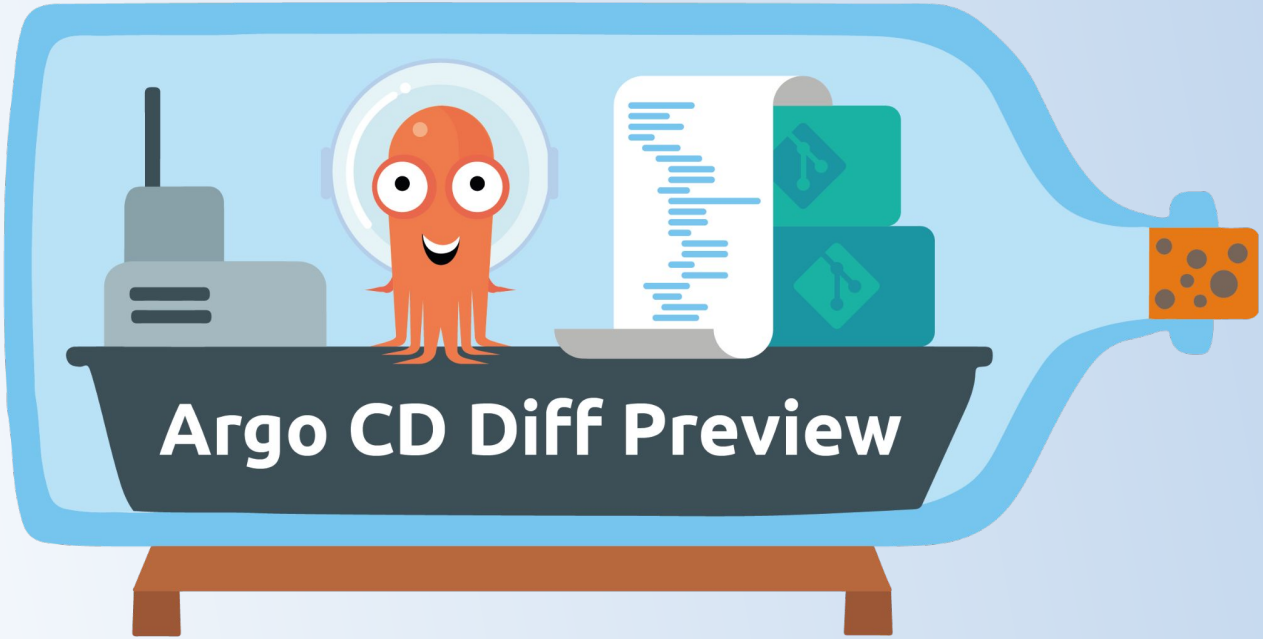
minikube

The Flow


Creates
Developer
Update Nginx #15
Open
dag-andersen wants to merge 1 commit into main from helm-example-1
Conversation
Commits
Checks
Files changed
dag-andersen commented 17 hours ago
Owner
Nginx version 4.9.1
Verified
github-actions bot commented 17 hours ago
Argo CD Diff Preview
Summary:
1 file changed, 25 insertions(+), 26 deletions(-)
Diff:
Pull Request
Triggers
CI Pipeline
Starts
kind
Local Cluster
Runs
Argo CD
Extracts Diff
Comments Diff

The Flow


Creates
Developer
Update Nginx #15
Open
dag-andersen wants to merge 1 commit into main from helm-example-1
Conversation
Commits
Checks
Files changed
dag-andersen commented 17 hours ago
Owner
Nginx version 4.9.1
Verified
github-actions bot commented 17 hours ago
Argo CD Diff Preview
Summary:
1 file changed, 25 insertions(+), 26 deletions(-)
Diff:
Pull Request
Triggers
Comments
Diff
CI Pipeline
Starts
kind
Local Cluster
Runs
Argo CD
Extracts
Diff


Argo CD Diff Preview


Argo CD Diff Preview

Demo

https://github.com/dag-andersen/argocon-us-2024-demo

Demo (in case wifi does not work) 1/3

ArgoCon NORTH AMERICA

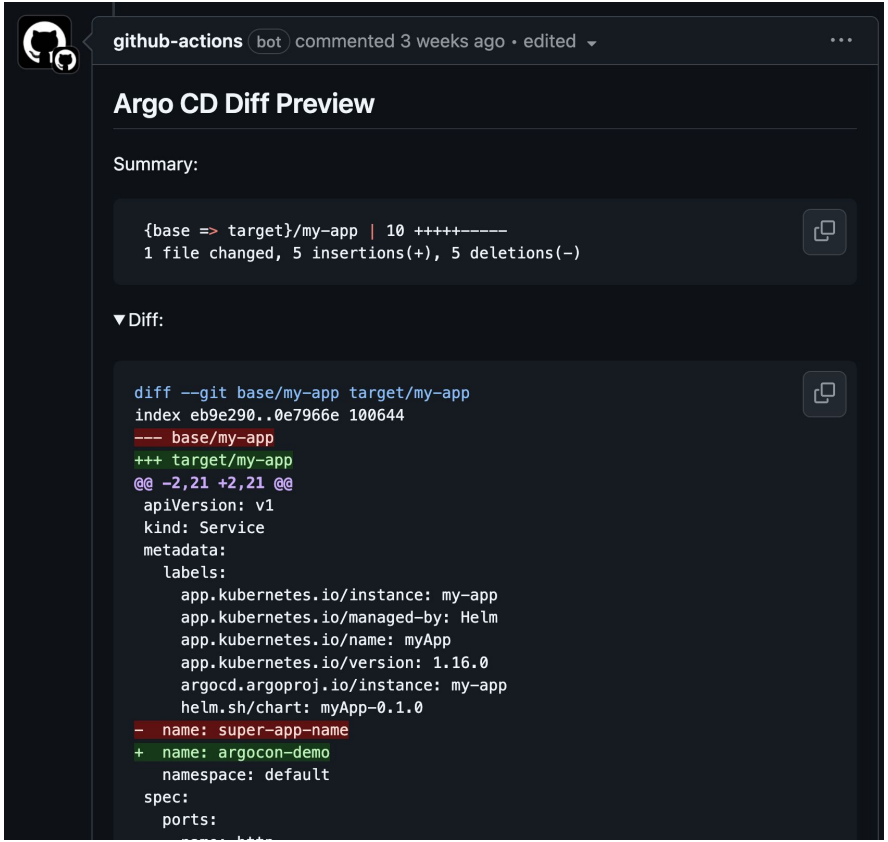
2 applications/my-app.yaml — Viewed

```
@@ -1,18 +1,20 @@
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: my-app
  namespace: argocd
spec:
  project: default
  destination:
    name: in-cluster
    namespace: default
  sources:
    - repoURL: https://github.com/dag-andersen/argocon-demo
      ref: local-files
    - path: charts/myApp
      repoURL: https://github.com/dag-andersen/argocon-demo
      helm:
        valueFiles:
          - $local-files/values/values.yaml
+       valuesObject:
+         replicaCount: 5
```

2 values/values.yaml — Viewed

```
@@ -1 +1 @@
- fullnameOverride: super-app-name
+ fullnameOverride: argocon-demo
```

Demo (in case wifi does not work) 2/3



Demo (in case wifi does not work) 3/3

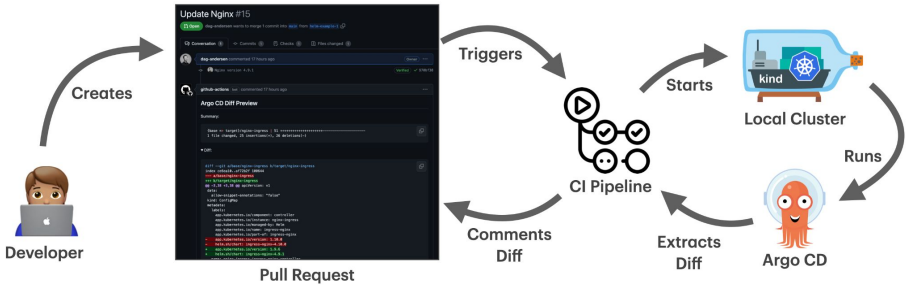
ArgoCon
NORTH AMERICA

.github/workflows/generate-diff.yml

```yaml
name: Argo CD Diff Preview

on:
  pull_request:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      pull-requests: write

    steps:
      - uses: actions/checkout@v4
        with:
          path: pull-request

      - uses: actions/checkout@v4
        with:
          ref: main
          path: main

      - name: Generate Diff
        run: |
          docker run \
            --network=host \
            -v /var/run/docker.sock:/var/run/docker.sock \
            -v $(pwd)/main:/base-branch \
            -v $(pwd)/pull-request:/target-branch \
            -v $(pwd)/output:/output \
            -e TARGET_BRANCH=${{ github.head_ref }} \
            -e REPO=${{ github.repository }} \
            dagandersen/argocd-diff-preview:v0.0.23

      - name: Post diff as comment
        run: |
          gh pr comment ${{ github.event.number }} \
            --repo ${{ github.repository }} \
            --body-file output/diff.md
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

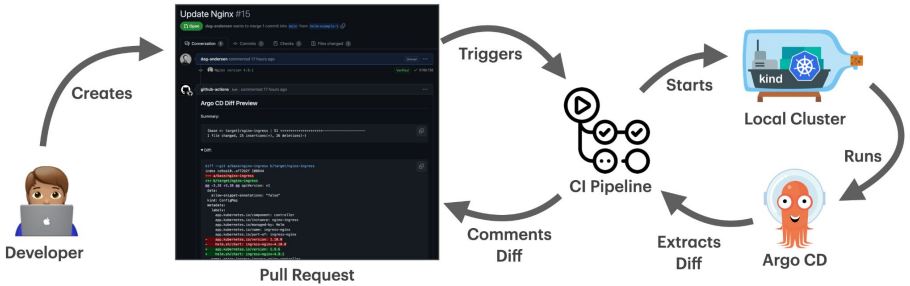
How does it work? (1/2)


```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  project: default
  syncPolicy:
    automated:
      prune: true
      selfHeal: true
    syncOptions:
      - ApplyOutOfSyncOnly=true
  destination:
    name: https://whatever.com
    namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      ref: local-files
      targetRevision: HEAD
    - chart: my-chart
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```



How does it work? (1/2)


```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  syncPolicy:
    automated:
      prune: true
      selfHeal: true
    syncOptions:
      - ApplyOutOfSyncOnly=true
  destination:
    name: https://whatever.com
    namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      ref: local-files
      targetRevision: HEAD
    - chart: my-chart
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```

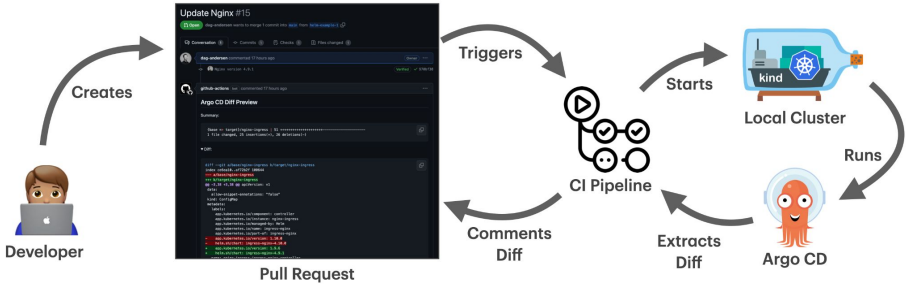


How does it work? (1/2)

ArgoCon NORTH AMERICA

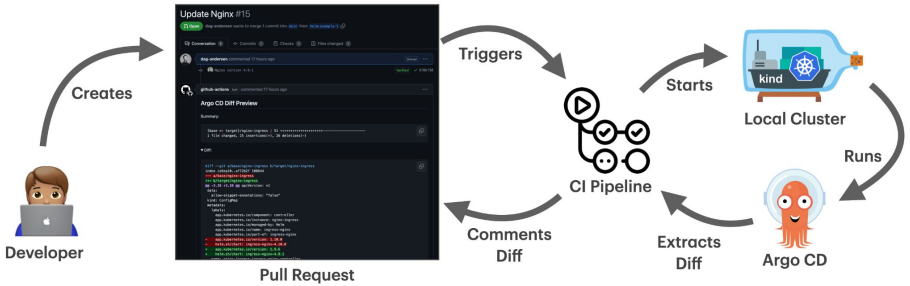
```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  project: default
  syncPolicy:
```

```yaml
  destination:
    name: https://whatever.com
    namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      ref: local-files
      targetRevision: HEAD
    - chart: my-chart
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```



How does it work? (1/2)

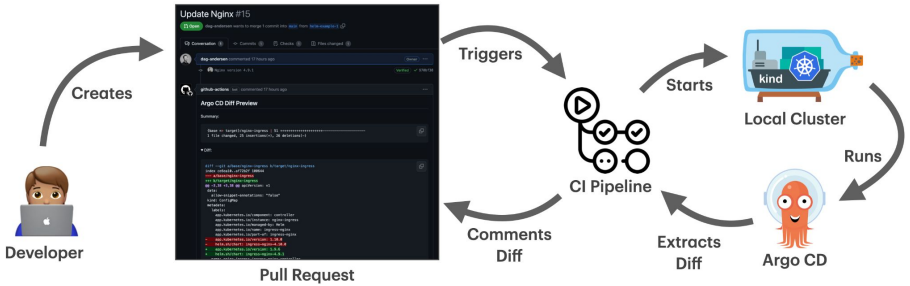

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  project: default
  syncPolicy:
    
  destination:
    name: https://whatever.com
    namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      ref: local-files
      targetRevision: HEAD
    - chart: my-chart
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```



How does it work? (1/2)

ArgoCon NORTH AMERICA

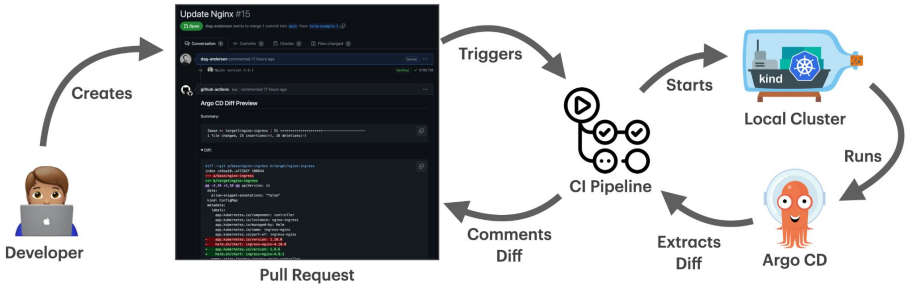
```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  project: default
  syncPolicy:
    destination:
      name: XXXXXX → "in-cluster" 🔄
      namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      ref: local-files
      targetRevision: HEAD
    - chart: my-chart
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```



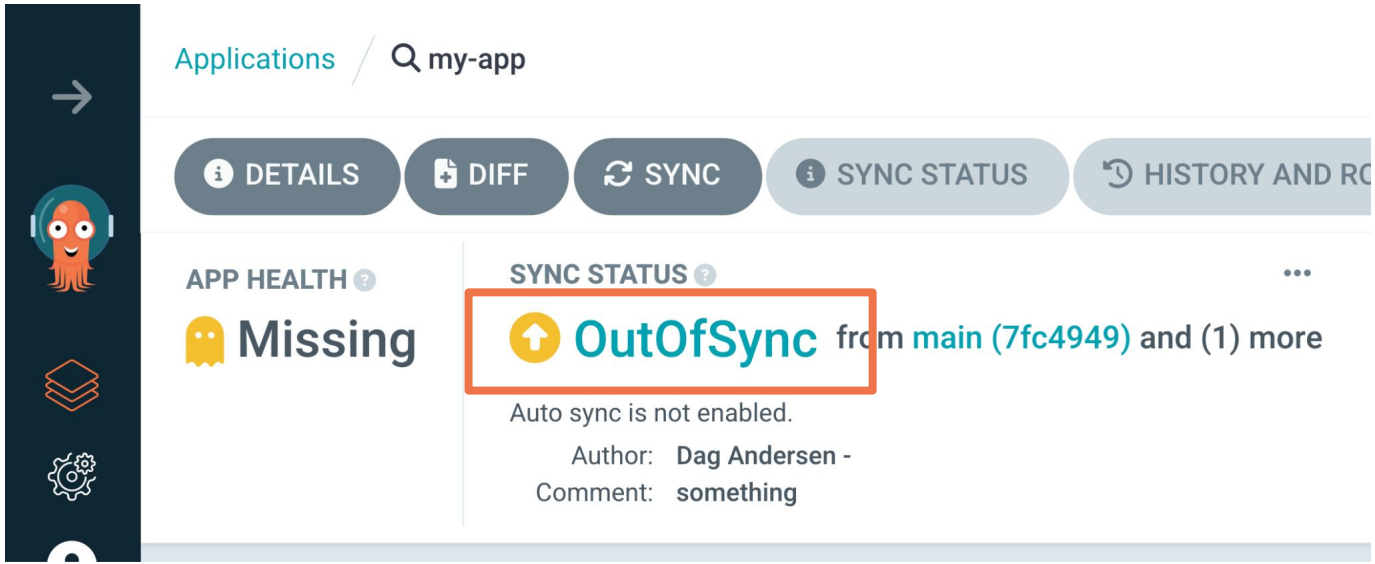
How does it work? (1/2)

ArgoCon NORTH AMERICA

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: something
spec:
  project: default
  syncPolicy:
    ❌
  destination:
    name: ❌❌❌❌❌❌ → "in-cluster" 🔄
    namespace: something
  sources:
    - repoURL: https://github.com/dag-andersen/repo.git
      targetRevision: ❌❌❌ → "<branch-name>" 🔄
      repoURL: helm-chart-registry.com
      targetRevision: x.x.x
      helm:
        releaseName: something
        valueFiles:
          - $local-files/whatever/values.yaml
```



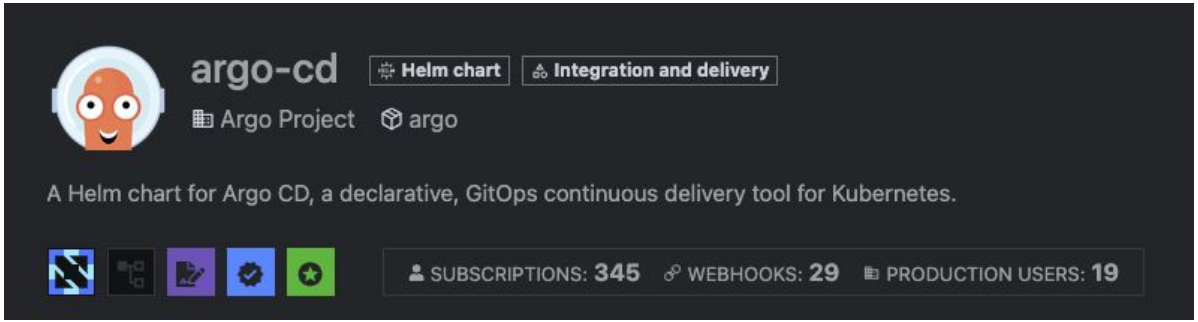
How does it work? (2/2)




```
$ argocd app manifests <app> >> extracted.yaml
```

Custom Argo CD Installation and CMPs

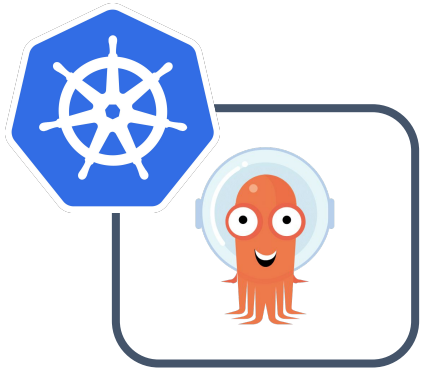
- Argo CD is installed through a Helm Chart
- You can edit any value you want
  - Match your live cluster
- You can install any Config Management Plugin that is installable from the Helm Chart



Desired state vs live state

When using ephemeral clusters

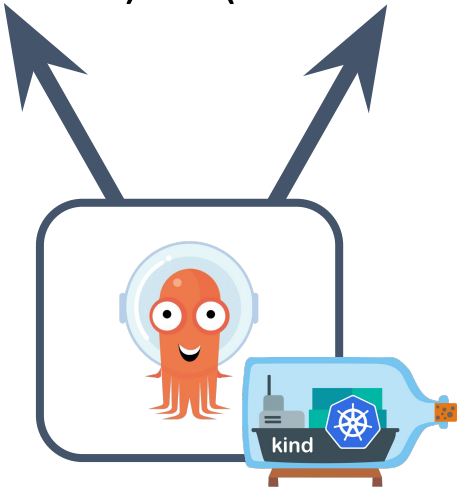
Comparing desired states only



Prod Cluster
(*Live State*)

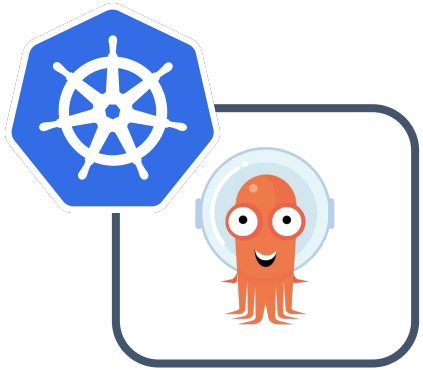
Main Branch
(*Desired State*)

PR Branch
(*Desired State*)



Ephemeral Cluster

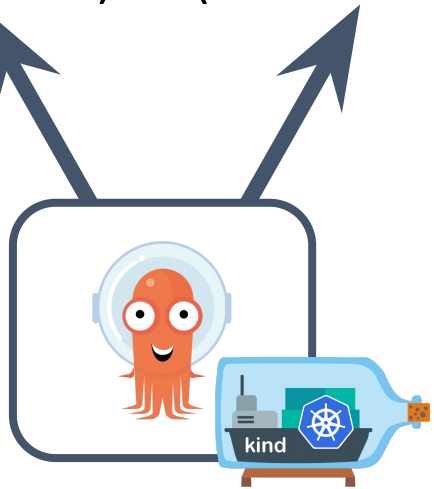
Comparing desired states only



Prod Cluster
(*Live State*)

Main Branch
(*Desired State*)

PR Branch
(*Desired State*)



Ephemeral Cluster

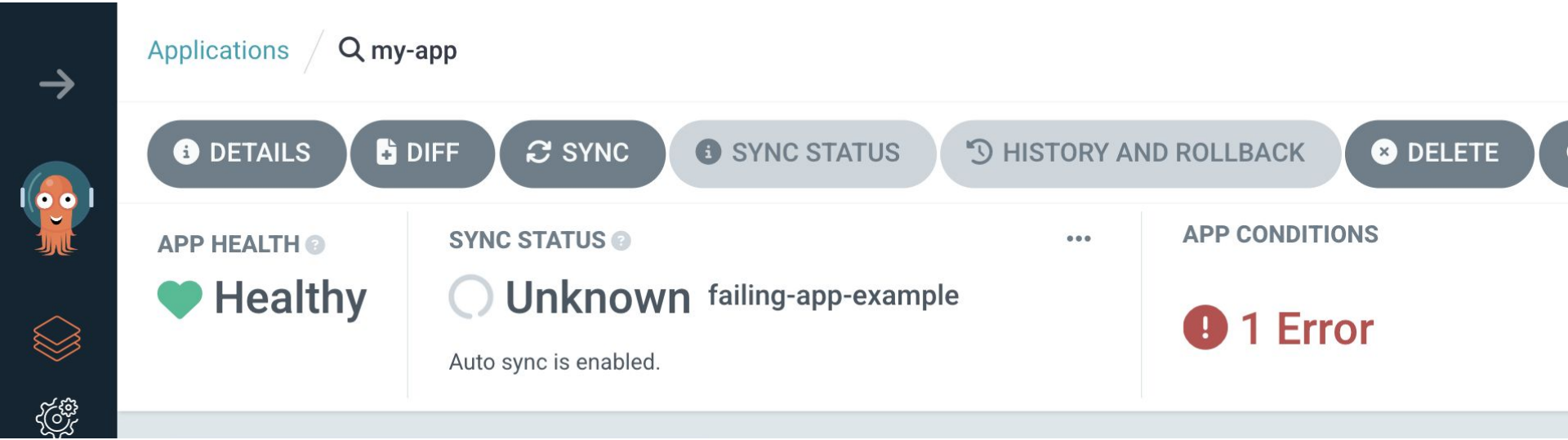
Faster feedback loop

We want to find issues before we merge

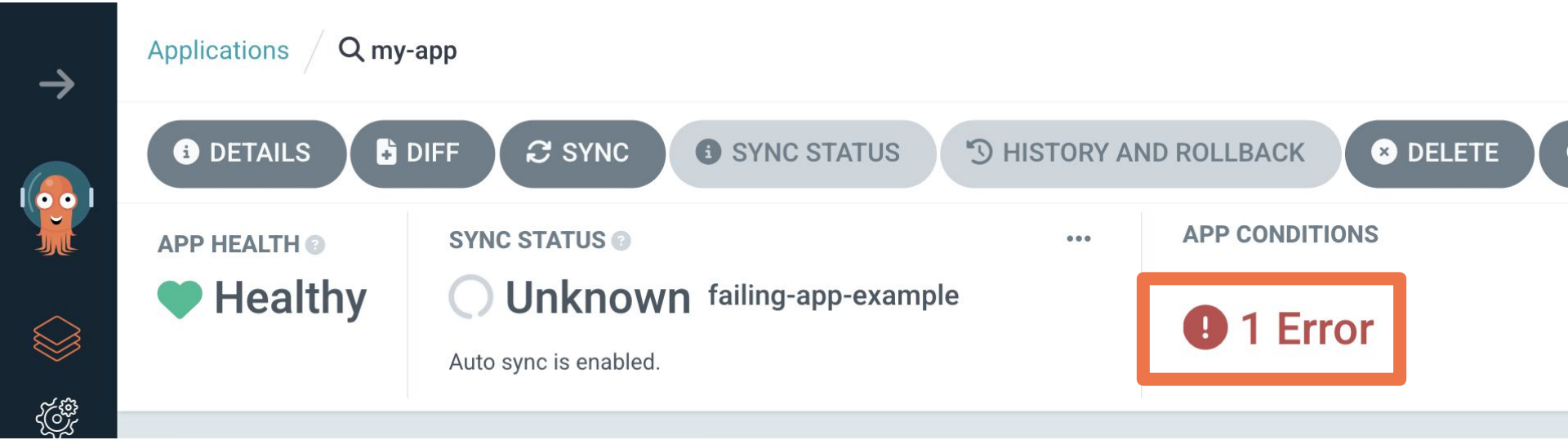
Faster feedback loop

creates a

Faster feedback loop



Faster feedback loop

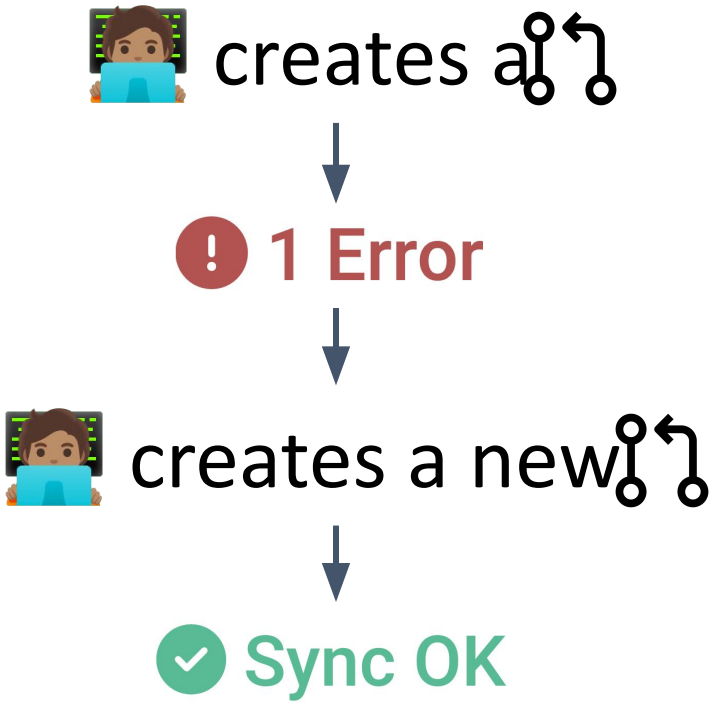


Faster feedback loop

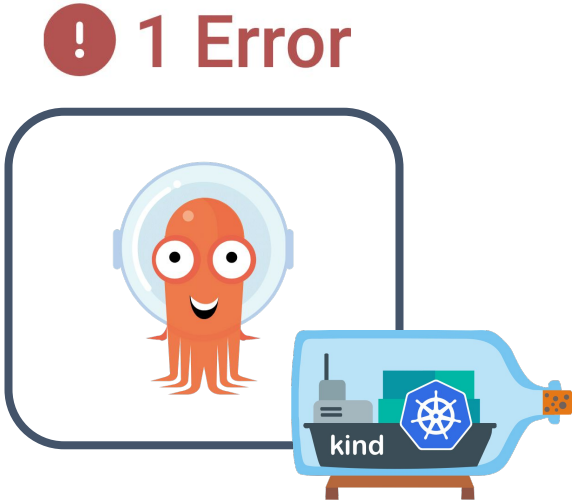
ComparisonError 

Failed to load target state: failed to generate manifest for source 2 of 2: rpc error: code = Unknown desc = `helm template . --name-template my-app --namespace default --kube-version 1.31 --values <path to cached source>/examples/some/wrong/path/values.yaml <api versions removed> --include-crds` failed exit status 1: Error: open <path to cached source>/examples/some/wrong/path/values.yaml: no such file or directory

Faster feedback loop



Faster feedback loop



Ephemeral Cluster

Faster feedback loop


```
Installing Argo CD Helm Chart version: 'latest'
Waiting for Argo CD to start...
Argo CD is now available
Logging in to Argo CD through CLI...
Argo CD installed successfully
Getting resources from base-branch
Waiting for 5 out of 5 applications to become 'OutOfSync'. Retrying in 5 seconds. Timeout in 180 seconds...
Got all resources from 5 applications for base
Removing applications
Removed applications successfully
Getting resources from target-branch
Waiting for 5 out of 5 applications to become 'OutOfSync'. Retrying in 5 seconds. Timeout in 180 seconds...
Failed to process application: my-app with error:
Failed to load target state: failed to generate manifest for source 2 of 2: rpc error: code = Unknown desc = `helm template .
--name-template my-app --namespace default --kube-version 1.31 --values <path to cached source>/examples/some/wrong/path/values.yaml
<api versions removed> --include-crds` failed exit status 1: Error: open <path to cached source>/examples/some/wrong/path/values.
yaml: no such file or directory
Error: "Failed to process applications"
```

ArgoCon
NORTH AMERICA

Limitations

Things to keep in mind…

False sense of security

ArgoCon
NORTH AMERICA

Some issues happen at run time (or at resource creation time)

False sense of security

Some issues happen at run time (or at resource creation time)

Examples of things that can break:

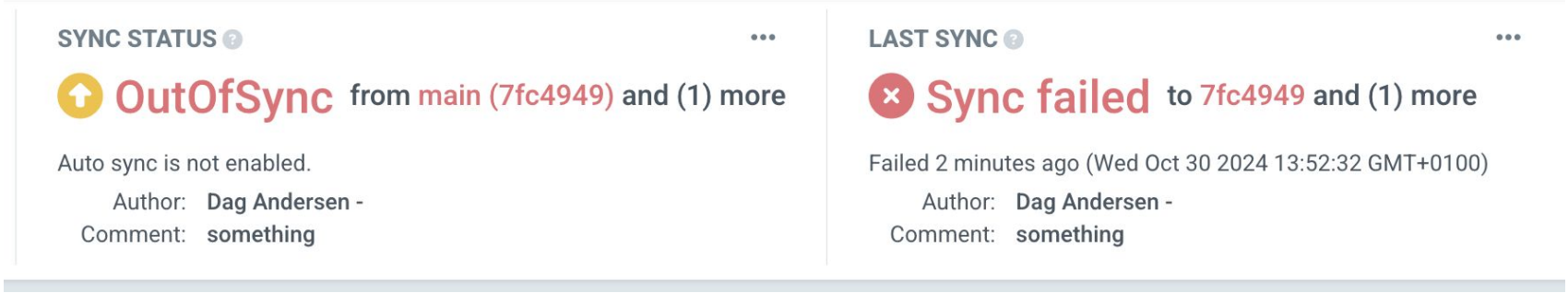
- Pods being rejected because of: `securityContext.runAsNonRoot: true`

False sense of security

Some issues happen at run time (or at resource creation time)

Examples of things that can break:

- Pods being rejected because of: `securityContext.runAsNonRoot: true`
- Missing `CustomResourceDefinitions` in your destination cluster



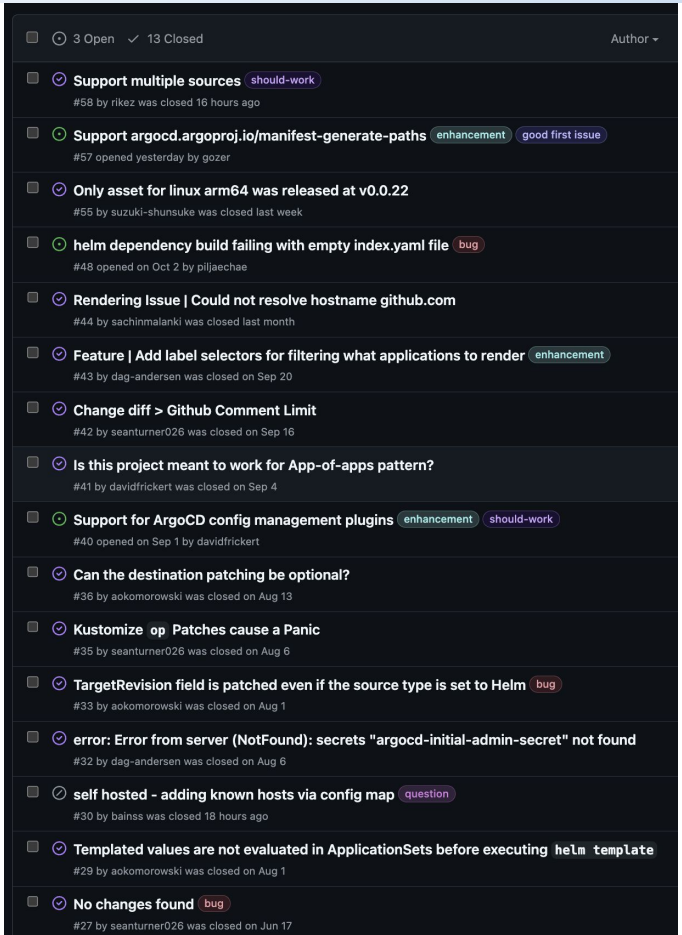
Speed

- Spinning up a local Kubernetes cluster and installing Argo CD on it every single run takes time
  - It takes around 70 seconds

- The full PR pipeline may take 2+ minutes
  - Pulling code x2
  - Pulling the docker image
  - Running the tool

- Speed depends on how many applications you are rendering

GitHub Issues



Adoption

Who uses `argocd-diff-preview`?

Companies using `argocd-diff-preview`

Summarize main points

What did we learn?

Summary

Main benefits of using *ephemeral clusters*

- Runs in complete isolation
  - Great for security
- Runs in parallel
  - No queues, No locks, No webhooks
- Enables comparing desired states only
  - Out-of-sync apps are not a problem
  - Can even be run locally

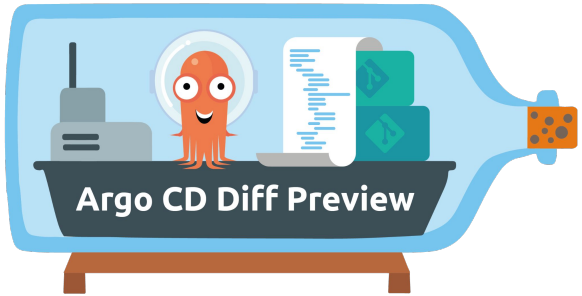
Main limitations

- Spinning up a local cluster and installing Argo CD is slow
- Can give a false sense of security

Summary

Main benefits of using *specifically* `argocd-diff-preview`

- Easy to set up
- Not opinionated
  - It only compares branches



Main limitations

- Does not work well when the Argo CD configuration is spread across multiple repositories
  - We are currently working on improving this

Thank you!

ArgoCon
NORTH AMERICA

If you have any questions, please feel free to reach out to us after the talk, on LinkedIn, or on CNCF Slack

Dag Bjerre Andersen
Linkedin
https://www.linkedin.com/in/dagbjerreandersen/

argocd-diff-preview
GitHub
https://github.com/dag-andersen/argocd-diff-preview

Regina Voloshin
Linkedin
https://www.linkedin.com/in/regina-voloshin-09b9841/