



# The pains of GitOps 1.0

And the vision for GitOps 2.0

KOSTIS KAPELONIS

# Agenda

What is GitOps

Scope and lack of standards

Promotion to different environments

Observability and visibility

Mixing CI and CD

GitOps at scale

Proposed solutions

The future of GitOps

# What is GitOps

## Everything in Git

Git is single source of truth

## Declarative deployments

Deploy by matching Git state with cluster state

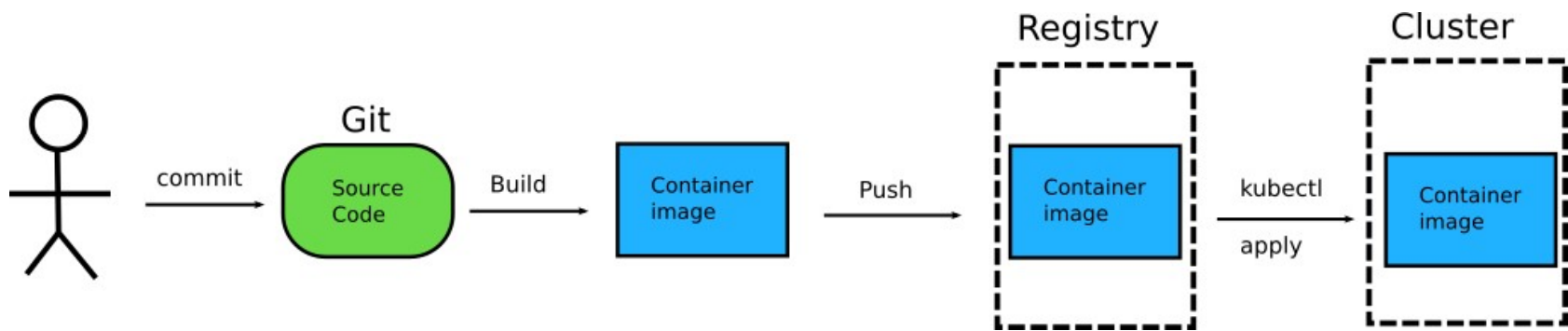
## No adhoc actions

Operations by Pull Request

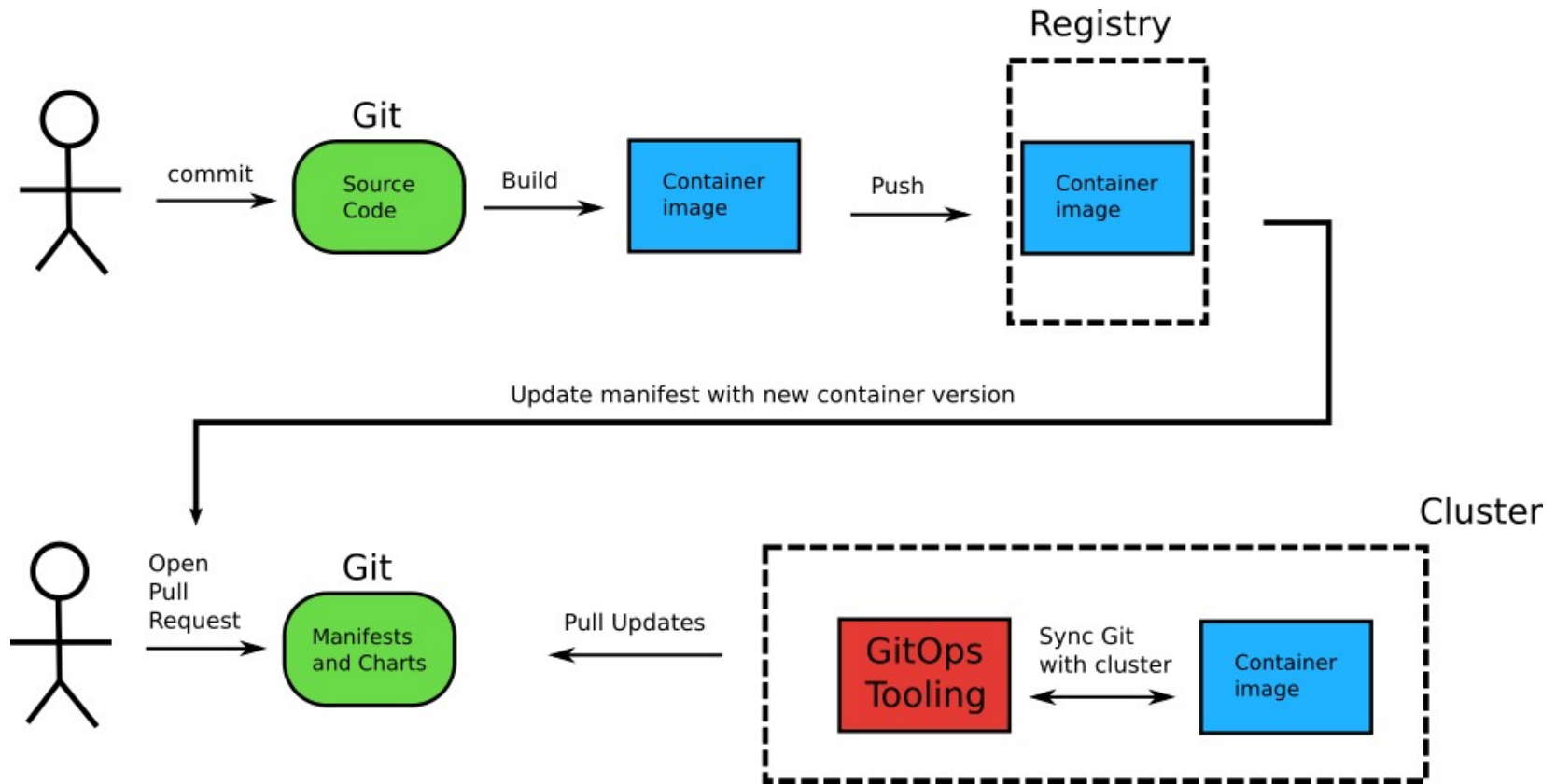
<https://www.weave.works/blog/gitops-operations-by-pull-request>

## Traditional deployment

---



## GitOps deployment



# GitOps advantages

**Git state = cluster state**

Git is single source of truth

**Git history = deployment  
history**

Deploy by matching Git state with cluster state

**Deploy/rollback with Git tools**

Operations by Pull Request

## GitOps tools

# ArgoCD

<https://argoproj.github.io/argo-cd/>

Applications / **guestbook**

APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

**Healthy** **Synced** **Sync OK**

To HEAD (6bed858)  
Authored by Alex Collins <alexco...  
Updates examples to better reflec...

To 6bed858  
Succeeded 7 days ago (Mon Aug 17 2020 19:27:35 GMT+0300)  
Authored by Alex Collins <alexco...  
Updates examples to better reflect hoo... (#41)

guestbook (application) → guestbook-ui (svc) → guestbook-ui (deploy) → guestbook-ui-65b878495d-8rcbt (pod)

# Flux V2

<https://toolkit.fluxcd.io/>

my-azure-demo Workloads Deploy Explore Monitor Kostas

Resources History

You can configure git webhooks to speed up syncing

Select source repo/cluster: Image repository search Filters Update all More actions

Workload	Image	Source Latest image	Target my-azure-demo	Status
codefresh/daemonset/dind-ly-monitor-runner	codefresh/dind-volume-utils	v15 2mo	v5 3y	Read-only
codefresh/deploy.../dind-volume-provisioner-runner	codefresh/dind-volume-provisioner	v25 15d	v28 8mo	Read-only
codefresh/deployment/monitor	codefresh/agent	25245388aa77842adca1... 2mo	stable 2mo	Read-only
codefresh/deployment/runner	codefresh/venona	1.5.5-d07f344d30a6f25... 1d	1.3.5 5mo	Read-only
default/deployment/simple-deployment	kostiscodefresh/simple-web-app	c88df35 2mo	3d9b390 2mo	Updatable
flux/deployment/flux	fluxcd/flux	1.21.0 1mo	1.21.0 1mo	Read-only
flux/deployment/memcached	memcached	1-alpha 7d	1.5.28 10mo	Read-only
weave.daemonset/prom-node-exporter	prom/node-exporter	master 9d	v0.15.2 3y	Read-only
weave.daemonset/weave-scope-agent	weaveworks/scope	3810-cordon-control-c... 4d	1.13.1 6mo	Read-only
weave/deployment/kube-state-metrics	coreos/kube-state-metrics	v2.0.0-alpha.2 8d	v1.6.0 1y	Read-only
weave/deployment/prometheus	prom/prometheus	v2.22.1 32m	v2.22.0 2td	Read-only
	weaveworks/watch	master-8944d57-wip 23d	master-85fd1d 3y	

## Applied GitOps

# Solve configuration drift

Know what changed in the cluster

# Use Git for everything

...and not just for source code

# Deployment is pull request

Merging a pull request starts the deployment

The screenshot shows a web interface for comparing two Kubernetes manifests. The interface has tabs for SUMMARY, PARAMETERS, MANIFEST, DIFF (selected), and EVENTS. Below the tabs, there are two checkboxes:  Compact diff and  Inline Diff. The main content area displays a diff for the manifest `/Service/default/guestbook-ui`. The diff is presented in two columns, with line numbers on the left of each column. The left column shows the original manifest, and the right column shows the modified manifest. The only change is in the `ports` section, where the `targetPort` has been updated from `8080` to `80`. This change is highlighted with a green background in the original image.

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4 labels:
5   app.kubernetes.io/instance: guestbook
6 name: guestbook-ui
7 spec:
8   ports:
9     - port: 8080
10       targetPort: 8080
11 selector:
12   app: guestbook-ui

1 apiVersion: v1
2 kind: Service
3 metadata:
4 labels:
5   app.kubernetes.io/instance: guestbook
6 name: guestbook-ui
7 spec:
8   ports:
9     - port: 80
10       targetPort: 80
11   - port: 8080
12       targetPort: 80
13 selector:
14   app: guestbook-ui
```



# GitOps challenges

What lies beneath the surface

# GitOps gaps

## Scope

GitOps covers only deployments

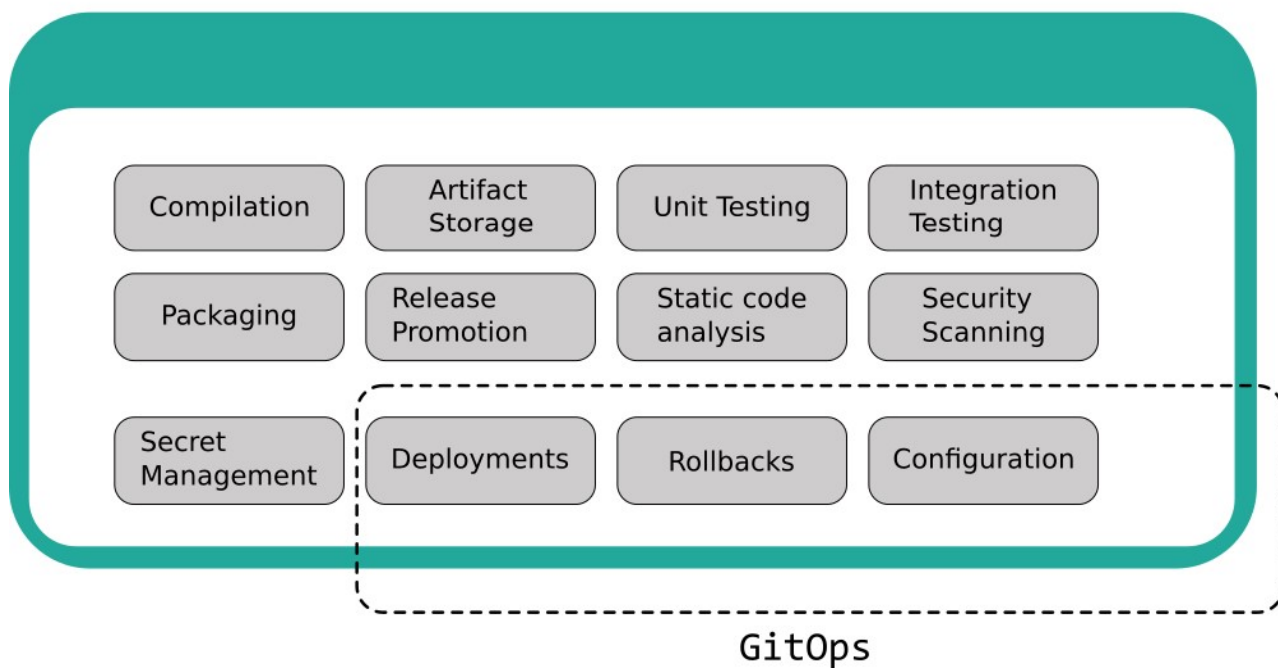
## How to do Rollbacks

No best practice

## How to handle secrets

No best practice

## GitOps only covers deployments



No documented best practices

---

## How to handle secrets

- Store in Git?
- Fetch from Hashicorp Vault?
- Use Bitnami Sealed secrets
- Use Mozilla Sops?

## How to do rollbacks?

- Manually git revert?
- Manually git reset?
- Automatically go to previous hash?
- Automatically go to previous container image?

## Rolling back

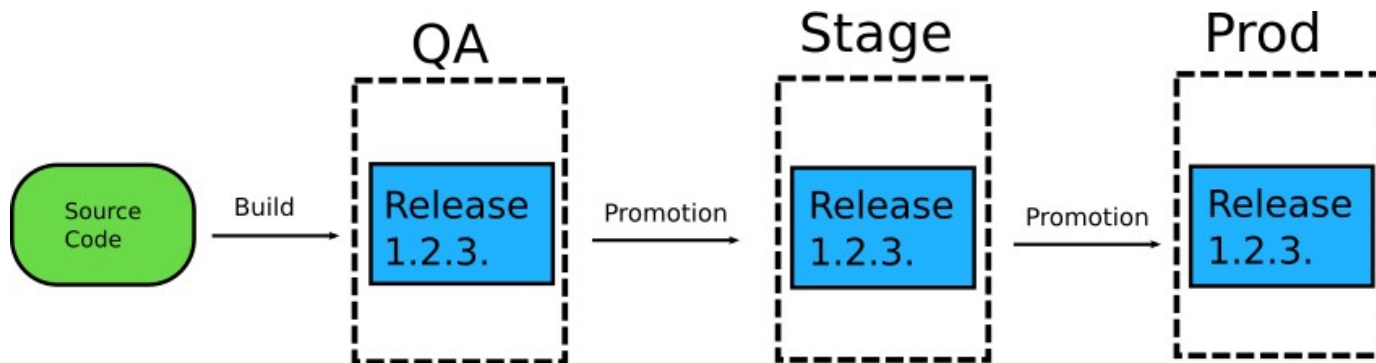
# Flux rollback

The screenshot shows the Flux UI for a deployment named 'default:deployment/simple-deployment'. At the top, there are buttons for 'Lock', 'Automate', and 'Update'. Below this, the 'Images' section shows 'webserver-simple' with the Docker image 'docker.io/kostiscodefresh/simple-web-app:3d9b390' and a note that it is '2 months ago' and '2 versions behind latest'. A 'Filter tags' dropdown is present. Below the images, there is a list of tags: 'c88df35' (2 months ago), 'latest' (2 months ago), 'master' (2 months ago), and '3d9b390' (2 months ago). The '3d9b390' tag is highlighted as 'Current'. To the right of the tags are 'Release' buttons. The 'Status' section shows a green dot and the word 'Ready'. The 'History' section shows a 'Manual deploy' and a 'Sync to cluster' event, both occurring 'a few seconds ago'. A 'Commit' by Kostis Kapelonis is also listed as occurring 'a minute ago'.

# ArgoCD rollback

The screenshot shows the ArgoCD UI for a deployment. It displays two identical panels for revision '6bed858'. The top panel shows 'Deployed At: 7 days ago (Mon Aug 17 2020 19:27:35 GMT+0300)' and 'Active for: 6 days 17:52 hours'. The 'Revision' section shows 'Revision: 6bed858' and 'Authored by: Alex Collins <alexec@users.noreply.github.com>' (10 months ago). A 'Redeploy' button is visible. Below this, there are sections for 'DIRECTORY', 'DIRECTORY RECURSE' (false), 'TOP-LEVEL ARGUMENTS', and 'EXTERNAL VARIABLES'. The bottom panel shows 'Deployed At: 7 days ago (Mon Aug 17 2020 18:31:49 GMT+0300)' and 'Active for: 55:46 min'.

# Environment promotion



# Environment promotion

↑ **GassiestFunInTheWest** 1 point · 1 year ago

↓ Do you run multiple environments? How do you orchestrate releases through and across environments?

[Give Award](#) [Share](#) [Report](#) [Save](#)

↑ **Tacticus** 11 points · 1 year ago

↓ How do you handle promotion through different environments or rollback in the event of a bad service deploy?

[Give Award](#) [Share](#) [Report](#) [Save](#)

↑ **coderranger** 3 points · 1 year ago

↓ Branches, different envs are configured to watch different branches of the gitops repos. Promotion is opening a PR to merge from one to the next. Rollback is undoing that merge.

[Give Award](#) [Share](#) [Report](#) [Save](#)

↑ **Tacticus** 1 point · 1 year ago

↓ So you don't have any automated progression through environments. manual branch merges are it?

[Give Award](#) [Share](#) [Report](#) [Save](#)

↑ **Tacticus** 7 points · 1 year ago

↓ Gitops tools really need to find a better way of doing multi environment promotion.

[Give Award](#) [Share](#) [Report](#) [Save](#)

↑ **GassiestFunInTheWest** 3 points · 1 year ago

↓ Agreed. Without this, GitOps is strictly worse than CD pipelines. From what I can see though, it's not something that the GitOps movement is trying to address.

As such, I'm going to stick with push-based CD pipelines.

[Give Award](#) [Share](#) [Report](#) [Save](#)

First environment



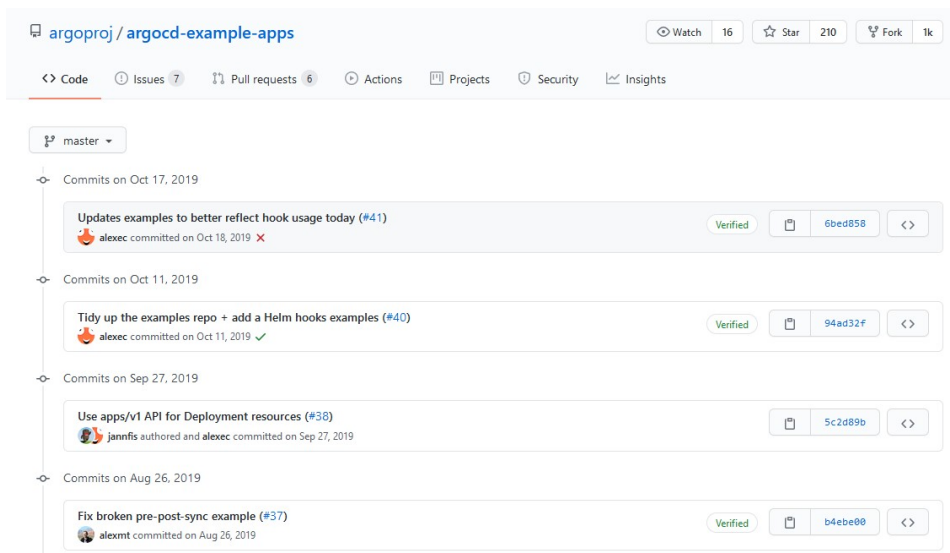
**Commit to App repo**  
**Create container**  
**Push to registry**  
**Commit to manifest repo**  
**Open Pull Request**  
**Merge Pull Request**  
**Sync Cluster**

Second environment





# Observability



## No info on features

GitOps covers only deployments

## Git history is problematic

No info on unit tests, or testing results

## How to see multiple environments

And what features are deployed

## Typical questions in a team

---

### Cluster questions

- Is the application healthy?
- How many replicas?
- Is traffic routing correct?
- When was the last deployment?
- Is Git state same as cluster state?

### Application questions

- Does production have feature X?
- Is bug Y present in staging?
- What features were deployed yesterday?
- What was the last feature deployed 2 hours ago?
- How long did feature Z stay in the staging environment before going to production?

## Answering cluster questions

# ArgoCD

The screenshot shows the ArgoCD interface for an application named 'guestbook'. At the top, there are navigation tabs: APP DETAILS, APP DIFF, SYNC, SYNC STATUS, HISTORY AND ROLLBACK, DELETE, and REFRESH. Below these, the application is in a 'Healthy' state. There are two 'Synced' status indicators: one for the HEAD (6bed858) and another for Sync OK (6bed558). The Sync OK indicator shows it succeeded 7 days ago. A diagram below shows the application structure: 'guestbook' (application) branches into 'guestbook-ui' (service) and 'show 2 hidden resources' (filter). The 'guestbook-ui' service is linked to a 'deployment' resource, which is linked to a 'pod' resource. The 'pod' resource is currently in a 'running' state with 1/1 pods.

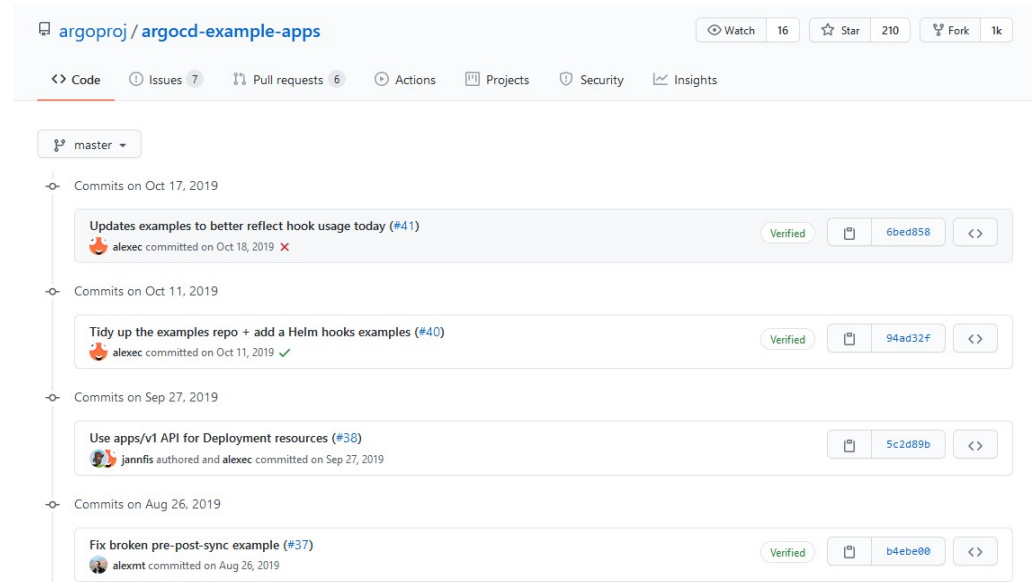
# Flux

The screenshot shows the Flux interface for a deployment. The top navigation bar includes Workloads, Deploy, Explore, Monitor, and Settings. The main content area is divided into several sections: Resources, History, Deploy progress, Summary, and Monitor. The 'Deploy progress' section shows a 'Sync' status with a green checkmark, indicating the deployment is up-to-date. Below this, the 'Apply' section shows a table with columns for Workload, Status, Old Pods, New Pods, and Elapsed time. The table shows a workload named 'default:deployment/sample-deployment' with a status of 'Deploying...', 11 Old Pods, 11 New Pods, and an elapsed time of 00:04. The 'Monitor' section shows a search bar for the workload and a 'View workload' button. The 'Resources' section shows a graph for CPU usage. At the bottom, there are buttons for 'Back to resources' and 'Updating 1 image'.

# Observability

## Git history != observability

- Hard to search
- Hard to reason
- Hard to correlate hashes with features
- No information on testing, security scanning, static analysis



The screenshot displays the GitHub interface for the repository `argoproj/argocd-example-apps`. The repository has 16 watchers, 210 stars, and 1k forks. The commit history is filtered to the `master` branch and shows four commits:

- Commits on Oct 17, 2019:** "Updates examples to better reflect hook usage today (#41)" by alexec, committed on Oct 18, 2019. Hash: `6bed858`.
- Commits on Oct 11, 2019:** "Tidy up the examples repo + add a Helm hooks examples (#40)" by alexec, committed on Oct 11, 2019. Hash: `94ad32f`.
- Commits on Sep 27, 2019:** "Use apps/v1 API for Deployment resources (#38)" by jannfis and alexec, committed on Sep 27, 2019. Hash: `5c2d89b`.
- Commits on Aug 26, 2019:** "Fix broken pre-post-sync example (#37)" by alexmt, committed on Aug 26, 2019. Hash: `b4ebe90`.



# Mixing CI and CD

## How to do smoke tests

How to rollback depending on test result?

## How to do parallel deployments

How to deploy to US, EU, Asia at the same time?

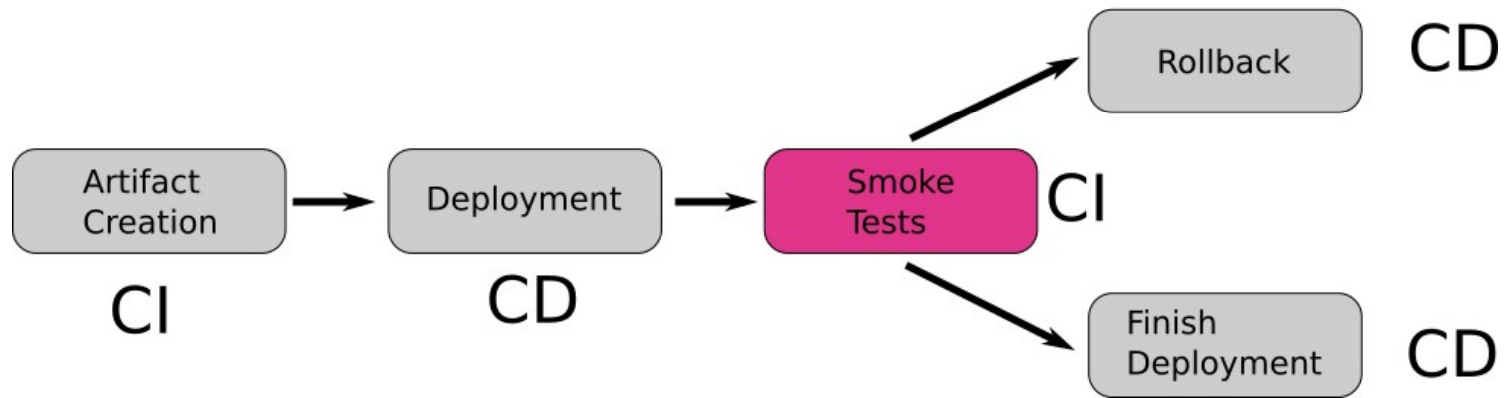
## Pre and Post deployment steps

How to send a slack message after deployment?

## Rollback on smoke tests

---

### Deployment phase



Other scenarios currently not handled by GitOps tools

---

## Pre-deployment

- Check network access
- Human Approval
- Mark deployment in metrics
- Validate container image
- Disable alerts

## Post-deployment

- Send slack message
- Mark deployment in metrics
- Trigger another deployment
- Cleanup resources



# GitOps at scale

## **Multiple environments**

Understand correlation of multiple deployments

## **Dynamic resources break**

Deploy by matching Git state with cluster state

## **No adhoc actions**

Operations by Pull Request only

## GitOps at scale

---

### For 10 environments

- 10 distinct Git repositories?
- Multiple parent-child repositories?
- 1 Git repository with 10 folders?
- 1 Git repository with 10 branches?

### Other problems

- How to find common settings?
- How to promote releases?
- How to co-ordinate deployments
- How to handle auditing?

## GitOps and dynamic resources

SUMMARY    PARAMETERS    MANIFEST    **DIFF**    EVENTS

Compact diff     Inline Diff

**apps/Deployment/simple/simple-deployment**

1	apiVersion: apps/v1	1	apiVersion: apps/v1
2	kind: Deployment	2	kind: Deployment
3	metadata:	3	metadata:
4	labels:	4	labels:
5	app.kubernetes.io/instance: sample-deployment	5	app.kubernetes.io/instance: sample-deployment
6	name: simple-deployment	6	name: simple-deployment
7	spec:	7	spec:
8	replicas: 3	8	replicas: 1
9	selector:	9	selector:
10	matchLabels:	10	matchLabels:
11	app: trivial-go-web-app	11	app: trivial-go-web-app
12	template:	12	template:
13	metadata:	13	metadata:
14	labels:	14	labels:
15	app: trivial-go-web-app	15	app: trivial-go-web-app
16	spec:	16	spec:
17	containers:	17	containers:
18	- image: 'docker.io/kostiscodefresh/simple-web-app:c88df35'	18	- image: 'docker.io/kostiscodefresh/simple-web-app:c88df35'
19	name: webservers-simple	19	name: webservers-simple
20	ports:	20	ports:
21	- containerPort: 8080	21	- containerPort: 8080

# A vision for GitOps 2.0

Solving the challenges

GitOps > guestbook

HEALTHY SYNCED URL: codefresh-demos SOURCE: https://github.com/argoproj/argocd-example-apps/guestbook DESTINATION: in-cluster/ync-waves LAST UPDATED: July 5, 2020 10:30

DASHBOARD ROLL OUTS SERVICES MAP

+ ADD FILTER This Week This Month Last 6 Months Jul 1, 2020 - Sep 24, 2020 Search Rollouts

Recent Rollouts

ROLLING OUT Updates examples to better reflect hook usage today | 1 code@ | CURRENT VERSION | 1 minute ago

Pull Requests #1234 #5132 Issues SAAS-1543 SAAS-1543 SAAS-1543 +3 more Committers klan Arbel (arbel) Ohg Vloverhol Pipelines a2k et blabtext +3 more

Updated Services

Service	Current Version	Target Version	Status
guestbook-ui	codefresh.io/ycu/build-a34d23	gcr.io/heptio-images/ks-guestbook-ui-0.1	1/4
guestbook-api	codefresh.io/pipeline-name/build-a34d23	gcr.io/heptio-images/ks-guestbook-api-0.3	3/4
guestbook-ui	codefresh.io/ycu/build-a34d23	gcr.io/heptio-images/ks-guestbook-ui-0.2	1/4
guestbook-api	codefresh.io/pipeline-name/build-a34d23	gcr.io/heptio-images/ks-guestbook-api-0.4	1/4

- Tidy up the examples repo + add a Helm hooks examples | 94ad32f | 1 hour ago
- Use apps/v1 API for Deployment resources | dbe0585 | ROLLBACK | July 26, 2020
- Tidy up the examples repo + add a Helm hooks examples | 94ad32f | July 25, 2020
- feat: use one configmap to read triggers and templates patterns | 7ee0b75 | July 25, 2020
- feat: update cm structure , support high level template and triggers | 11bb0bf | July 23, 2020
- chore: use one annotations for yaml and json | 61c89ff | July 23, 2020
- feat: move builtin triggers and templates into cm | 9428196 | July 21, 2020
- Run kubernetees deployment as non-root (#104) | 571aa08 | July 21, 2020

Load more

# Codefresh GitOps

Based on ArgoCD

# GitOps dashboard

**GitOps > guestbook**

HEALTHY SYNCED URL: codefresh-demo.com SOURCE: https://github.com/argoproj/argocd-example-apps/guestbook DESTINATION: in-cluster/sync-waves LAST UPDATED: July 1, 2020 | 20:30

DASHBOARD ROLL OUTS SERVICES MAP

+ ADD FILTER This Week This Month Last 6 Months Jul 1, 2020 → Sep 24, 2020 Search Rollouts

### Recent Rollouts

Timeline: 20 Jul, 21 Jul, 25 Jul, 26 Jul

**ROLLING OUT** Updates examples to better reflect hook usage today | 1ccdee0 **CURRENT VERSION** 1 minute ago

Pull Requests: #1234 #5132 Issues: SAAS-1563 SAAS-1563 SAAS-1563 +3 more Committers: Idan Arbel (arbel) Oleg VVerchol Pipelines: e2e cd bloateszt +3 more

#### Updated Services

Service	Old Version	Progress	New Version	Progress
guestbook-ui codefresh.io/e2e/build-a34d23	gcr.io/heptio-images/ks-guestbook-ui:0.1	3/4	gcr.io/heptio-images/ks-guestbook-ui:0.2	1/4
guestbook-api codefresh.io/pipeline-name/build-a34d23	gcr.io/heptio-images/ks-guestbook-api:0.3	0/4	gcr.io/heptio-images/ks-guestbook-api:0.4	4/4

COMPLETED

- Tidy up the examples repo + add a Helm hooks examples | 94ad32f | 1 hour ago
- Use apps/v1 API for Deployment resources | 6bed585 | ROLLBACK | July 26, 2020
- Tidy up the examples repo + add a Helm hooks examples | 94ad32f | July 25, 2020
- feat: use one configmap to read triggers and templates patterns | 7ee6b75 | July 25, 2020
- feat: update cm structure, support high level template and triggers. | 1f0bbfb | July 21, 2020
- chore: use one annotations for yaml and json | 81c8f9f | July 21, 2020
- feat: move builtin triggers and templates into cm | 9428196 | July 21, 2020
- Run kubernetes deployment as non-root (#104) | 571aba8 | July 21, 2020

Load more

## Easily correlate features and deployments

The screenshot displays the Codefresh GitOps dashboard for a project named 'guestbook'. The interface includes a top navigation bar with 'HEALTHY' status, 'SYNCED' indicator, and various settings like 'URL', 'SOURCE', 'DESTINATION', and 'LAST UPDATED'. Below this is a 'DASHBOARD' section with tabs for 'ROLL OUTS', 'SERVICES', and 'MAP'. A search bar for rollouts is present, showing filters for 'This Week', 'This Month', and 'Last 6 Months'. The main content area is titled 'Recent Rollouts' and features a timeline visualization. A specific rollout is expanded, showing a 'ROLLING OUT' status and a description: 'Updates examples to better reflect hook usage today'. This rollout is linked to several 'Pull Requests' (e.g., #1234, #5132), 'Issues' (e.g., SAAS-1563), 'Committers' (Idan Arbel, Oleg Vjverchov), and 'Pipelines' (e2e, cd, bloatesst). Below the rollout details, a list of 'Updated Services' is shown, including 'guestbook-ui' and 'guestbook-api', each with its previous and current versions and a 'ROLLBACK' button. A list of recent commit messages is also visible, such as 'Tidy up the examples repo + add a Helm hooks examples' and 'feat: use one configmap to read triggers and templates patterns'. A 'Load more' button is located at the bottom of the list.

# Codefresh GitOps

- See which features are in deployment
- See which Pull Requests affected a deployment
- Inspect result of unit tests and security scans
- Understand affected Kubernetes services

# Multi-environment dashboard

Search Kubernetes 🔍

Environment	Status	URL	Recent Activity	SERVICES	DEPLOYMENTS	PODS
accounts	ERROR	https://accounts.example.com	<ul style="list-style-type: none"><li>1/24/20, 8:16 PM Tagged 2.1.3 ❌</li><li>1/24/20, 8:15 PM Configmap change ✅</li><li>1/24/20, 8:13 PM Tagged 2.1.3 ✅</li></ul>	3 0 0 0	3 0 0 0	3 0 0 0
billing	DEPLOYED	https://billing.example.com	<ul style="list-style-type: none"><li>1/24/20, 8:13 PM Handled empty field ✅</li><li>1/24/20, 8:11 PM fixed apply button ✅</li><li>1/24/20, 8:09 PM rollback ✅</li></ul>	7 0 0 0	0 5 0 0	11 0 0 0
orders	DEPLOYED	https://orders.example.com	<ul style="list-style-type: none"><li>1/24/20, 8:13 PM bump to 0.1.2 ✅</li><li>1/24/20, 8:11 PM bump to 0.1.2 ✅</li><li>1/24/20, 8:09 PM bump to 0.1.1 ✅</li></ul>	7 0 0 0	0 5 0 0	11 0 0 0
prod	DEPLOYED	https://example.com	<ul style="list-style-type: none"><li>1/23/20, 6:55 PM Pipeline with version reading ✅</li><li>1/23/20, 6:53 PM Pipeline with version reading ✅</li></ul>	2 0 0 0	1 0 0 0	1 0 0 0
qa	DEPLOYED	https://qa.example.com	<ul style="list-style-type: none"><li>1/24/20, 4:24 PM Tagged 1.2.4 ✅</li><li>1/24/20, 4:22 PM Tagged 1.2.3 ✅</li></ul>	7 0 0 0	0 5 0 0	11 0 0 0
staging	IN PROGRESS	https://staging.example.com	<ul style="list-style-type: none"><li>1/24/20, 4:22 PM Config map change ⚙️</li></ul>	3 0 0 0	3 0 0 0	3 0 0 0



## Pre/Post deployment steps

The screenshot displays an ArgoCD pipeline run for the 'production' environment. The pipeline is titled 'cd' and is in a 'COMPLETED' state. The run duration is 11 min 16 s, and it was executed 2 hours ago. The pipeline consists of three stages: PREPARE, DEPLOY, and POST-DEPLOY.

**PREPARE →**

- Prepare environment variables (Step type: freestyle, 7 s)
- Cloning main repository... (Step type: git-clone, 4 s)
- Saving old requirements (Step type: freestyle, 7 s)

**DEPLOY**

- Renew SSL certs on dev environment (Step type: freestyle, 22 s)
- Update ArgoCD Application spec (Step type: deploy, 16 s)
- Sync ArgoCD app and wait (Step type: argocd-sync, 9 min 15 s)**
- deploy\_to\_saas\_environment (Step type: freestyle)
- Creating deployment mark for new relic (Step type: freestyle, 13 s)

**POST-DEPLOY**

- run\_e2e\_pipeline (Step type: codefresh-inc/codefresh-run)
- Preparing slack message (Step type: freestyle, 8 s)
- Sending message to slack (Step type: freestyle, 7 s)
- Sending error report to Slack (Step type: freestyle)

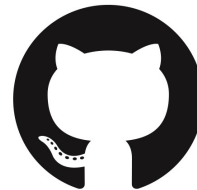
The 'Sync ArgoCD app and wait' step is highlighted with a red circle, indicating it is the focus of the 'Pre/Post deployment steps' discussion.

# The GitOps working group

A CNCF community project

## The GitOps working group

---



## An open working group

The founders of the GitOps Working Group are creating a neutral working group to clearly define a principle-led meaning of GitOps to better enable interoperability between tools. With a clear definition, GitOps Certification Programs for individuals will also be possible.

<https://github.com/fluxcd/gitops-working-group/>

## GitOps challenges and a vision for GitOps 2.0

---



<https://codefresh.io/devops/pains-gitops-1-0/>

<https://codefresh.io/devops/vision-gitops-2-0/>

<https://codefresh.io/devops/entering-gitops-2-0/>

<https://codefresh.io/devops/launching-future-devops-gitops-2-0/>

<https://codefresh.io/devops/announcing-gitops-working-group/>