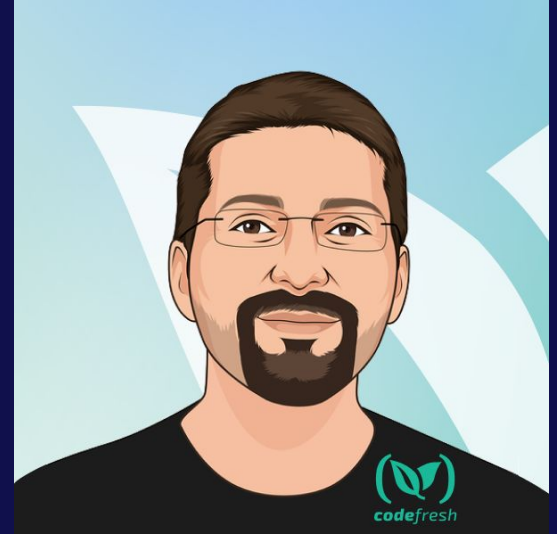


# Best Practices for GitOps Repositories

ArgoCon 22

# Kostis Kapelonis

- Developer Advocate at Codefresh and Argo Maintainer
- Codefresh is the Enterprise Platform for Argo
- Co-author of GitOps Certification with Argo -> <http://learning.codefresh.io>



# One of the Top questions for Argo CD newcomers

argoproj / argo-cd Public

Code Issues 1.6k Pull requests 143 Discussions Actions Projects 2

## Best practices for promotion between clusters #5667

Answered by kostis-codefresh musabmasood asked this question in Q&A

argoproj / argocd-example-apps Public

Code Issues 21 Pull requests 32 Actions Projects Security Insights

## [Question] Multiple environments(dev, stage, ... prod) example #57

Open emiroi opened this issue on Jul 16, 2020 · 13 comments

Thread # argo-cd

**Mykola Polonskyi** Jul 18th at 12:42 PM  
Hello community,  
Are there any information on common best practices for repository directory structure ?  
We are considering couple of them for big multi-regional and multi-environment projects and want to be aligned with community standards.

Thread # argo-cd

**Simon Chan** Mar 11th at 11:59 AM  
Going to rephrase my last question. Does anyone have any recommendations on how to layout my argocd managed manifests repo? How to manage permissions? How best to go from dev to qa to prod etc?  
Thanks (edited)

Thread # argo-cd

**procek** May 12th at 10:02 AM  
what is the flow to promote environments from for example from dev to and from test to prod? how can I achieve it ?

r/kubernetes

Posted by u/7thsvsn 2 years ago

## ArgoCD with multiple environments and clusters

I'm pretty new to k8s, and have set up ArgoCD and its app-of-apps pattern. This works fine, but now I'd like to introduce multiple environments (dev, stage, prod) with promotion between them, and prod would have also multiple clusters in different regions (using GKE). I would like each app's config repo to be as simple as possible, specifying only its image, readiness/liveness checks, resource request/limit, etc. They should not need to know about environments or clusters (but might actually have to. See below).

Thread # argo-cd

**ZW** Sep 5th at 8:58 PM  
Hi - we like to know how this is accomplished with argo-cd.  
Suppose we have 3 clusters per zone/region = dev, staging, prod.  
Suppose we have 3 zones/region.  
In gitops practice,  
Q1) how does a good build of a commit-id get promoted between environments in a region.  
Q2) how does a good build of a commit-id get promoted between zones ?  
Thank you



# Possible Options

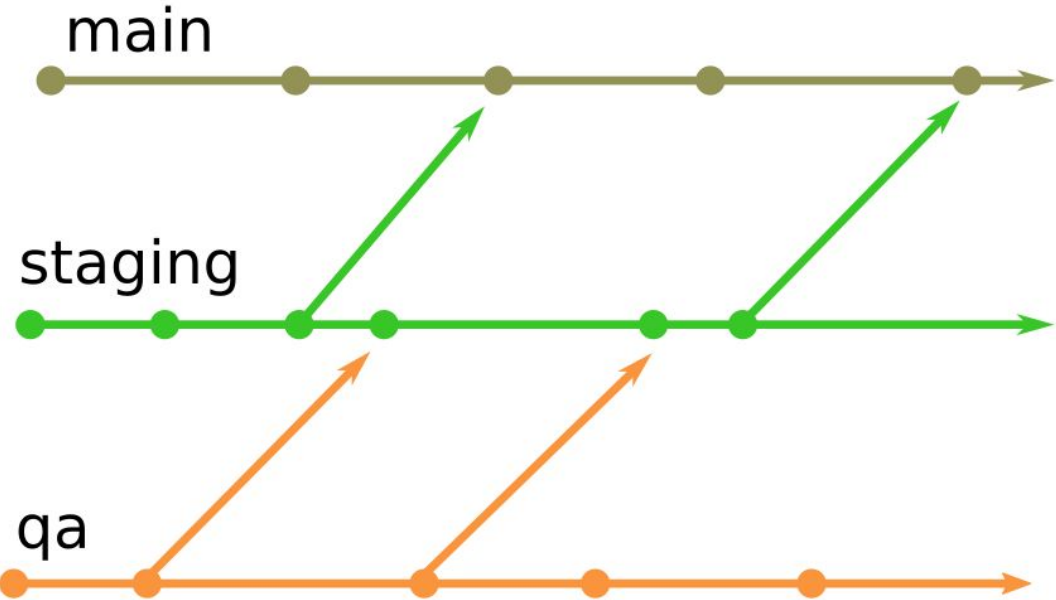
- Branch per environment
- Folder per environment
- Git Repo for each environment
- A combination of the above



Don't use branches for  
GitOps environments

Don't use long-running  
branches for GitOps  
environments

# Git branches



# Environments

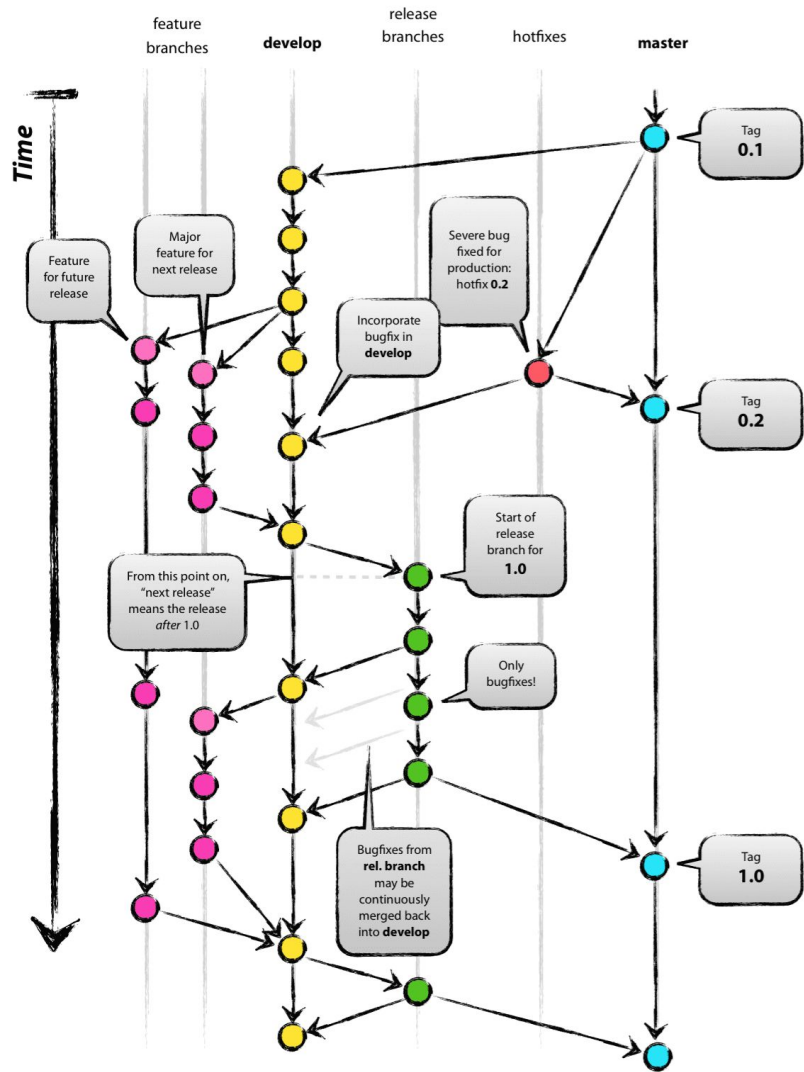
Production

Staging

QA

 Don't





# A successful Git branching model



By Vincent Driessen  
on Tuesday, January 05, 2010

## Note of reflection (March 5, 2020)

This model was conceived in 2010, now more than 10 years ago, and not very long after Git itself came into being. In those 10 years, git-flow (the branching model laid out in this article) has become hugely popular in many a software team to the point where people have started treating it like a standard of sorts — but unfortunately also as a dogma or panacea.

During those 10 years, Git itself has taken the world by a storm, and the most popular type of software that is being developed with Git is shifting more towards web apps — at least in my filter bubble. Web apps are typically continuously delivered, not rolled back, and you don't have to support multiple versions of the software running in the wild.

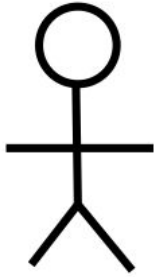
This is not the class of software that I had in mind when I wrote the blog post 10 years ago. If your team is doing continuous delivery of software, I would suggest to adopt a much simpler workflow (like GitHub flow) instead of trying to shoehorn git-flow into your team.

If, however, you are building software that is explicitly versioned, or if you need to support multiple versions of your software in the wild, then git-flow may still be as good of a fit to your team as it has been to people in the last 10 years. In that case, please read on.

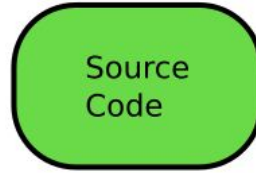
To conclude, always remember that panaceas don't exist. Consider your own context. Don't be hating. Decide for yourself.



Dev



Commit  
and  
push



Trunk-based  
Development

Git Repository



Commit  
and  
push



Trunk-based Development  
or any other methodology

Ops

Other Git  
repository



# Environments

Staging

Production

- Replica number decrease
- Hotfix for 1.24
- Upgrade app to 1.24
- Rate limit change
- Ingress config change

Git Commits in "staging" branch

There is no Git Merge operation that can bring needed changes

Upgrade app to 1.23

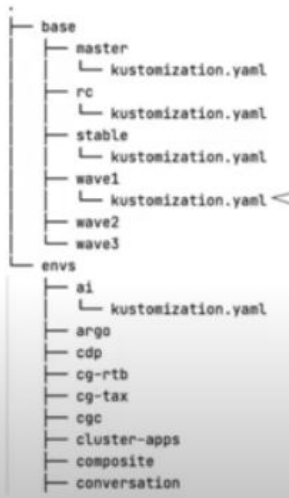
Main branch

**Do** use folders for GitOps  
environments

- > base
- > integration-gpu
  - > integration-non-gpu
  - > load-gpu
  - > load-non-gpu
  - > prod-asia
  - > prod-eu
  - > prod-us
  - > qa
  - > staging-asia
  - > staging-eu
  - > staging-us

- > variants
  - > asia
  - > eu
  - > non-prod
  - > prod
  - > us

## Manifests Structure



```

apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
  # rc, stable or master
  - ../rc
  
```

# Promote by copying files

- Promote/demote anything
- Order of commits does not matter
- Take only what you want
- Scales to any number of environments
- Most flexible for all corner cases



# Use the correct tools

- Kustomize -> overlays/folders
- Helm -> value files
- Argo CD can work with folders/subfolders
- ApplicationSets can work with folders



# Q/A and Resources

- <https://codefresh.io/blog/stop-using-branches-deploying-different-gitops-environments/>
- <https://codefresh.io/blog/how-to-model-your-gitops-environments-and-promote-releases-between-them/>
- <https://github.com/argoproj/argo-cd/discussions/5667>
- GitOps Certification with Live demo <https://learning.codefresh.io/>
- Argo CD best practices in Intuit (<https://www.youtube.com/watch?v=ESQLqjbM8h0>)

