

Continuous Delivery with Kubernetes for App Developers

Daniel Bryant @danielbryantuk
Edidiong Asikpo @Didicodes

www.getambassador.io @ambassadorlabs

Welcome! 🙌

- Review learning objectives
- Configure your dev env (get a remote k8s cluster, install Argo etc)
- K8s CD 101
- 10 minute coffee break
- Using Argo Rollouts for canary releasing
- Using Argo CD for GitOps
- Wrap up

Need help? Reach out on chat or join the [#developer-learning-center](#) in our Slack a8r.io/slack

Learning objectives



Learning objectives



Today, you will:

- Learn about continuous delivery
- Explore how continuous delivery has evolved on Kubernetes
- Understand about progressive delivery
- Get introduced to GitOps
- Get hands on with Argo Rollouts and Argo CD

Kubernetes Developer Learning Center

The hands-on experience for application developers learning Kubernetes.

START LEARNING JOIN THE COMMUNITY

There are a lot of ways to learn Kubernetes so how do you know where to start? The Kubernetes Developer Learning Center combines the best resources and popular open-source tools into a single learning experience-- completely for free.

Learn at your own pace with access to help from expert mentors and a community of users. From learning key architectural concepts to getting hands-on with essential tools, you'll pick up practical skills to help you adopt Kubernetes faster and with more confidence.

Course textbooks 
getambassador.io/kubernetes-learning-center

AMBASSADOR Products Developer Control Plane Developers Company Pricing Dev House GET STARTED

AMBASSADOR Products Developer Control Plane Developers Company Pricing Dev House GET STARTED

 Code
Configure your local development environment for building containers and testing K8s services faster.
GET STARTED

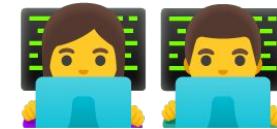
 Ship
Deploy your services without breaking production, using best practices like continuous delivery (CD) and GitOps.
GET STARTED

 Run
Observe and manage your applications across the full lifecycle of software delivery. Prepare for incidents to ensure continuous availability of your applications.
GET STARTED

What You'll Get

- Free training focused on application developers using Kubernetes
- Be part of a community of Kubernetes learners from around the world
- Hands-on experience with containers and Kubernetes
- A certificate of achievement
- Exposure to popular open source tools across the CNCF ecosystem
- UberEats vouchers when you submit any of the three modules
- Access to expert engineers who will share practical tips from using Kubernetes in production
- A chance to win exclusive swag and giveaways

Configure your dev env



Install the tooling

Visit Ambassador Labs and claim your free K8s demo cluster:

Create an account: a8r.io/workshop

Get a cluster getambassador.io/docs/telepresence/latest/quick-start/demo-node/

You will also need Docker Desktop installed:

www.docker.com/products/docker-desktop

Need help? Reach out on chat or join the [#developer-learning-center](#) on our Slack a8r.io/slack

Docs Home Kubernetes (K8s)

Edge Stack Emissary-ingress

Telepresence

Argo

Cloud

Envoy



Search documentation

Version: Latest ▾

Telepresence Quick Start

6 min • read

In this guide, we'll give you a hands-on tutorial with Telepresence. To go through this tutorial, the only thing you'll need is a computer that runs Docker Desktop >= 20.10.7. We'll give you a pre-configured remote Kubernetes cluster and a Docker container to run locally.

If you don't have Docker Desktop already installed, go to the [Docker download page](#) and install Docker.

While Telepresence works with any language, this guide uses a sample app written in Node.js and Golang. We have a version in [React](#) if you prefer.

1. Get a free remote cluster

Telepresence connects your local workstation with a remote Kubernetes cluster. In this tutorial, we'll start with a pre-configured, remote cluster.

GET A FREE REMOTE CLUSTER

Login in to Ambassador Cloud to activate your demo cluster.

2. Go to the [Service Catalog](#) to see all the services deployed on your cluster. The Service Catalog gives you a consolidated view of all your services across development, staging, and production. After exploring the Service Catalog, continue with this tutorial to test the application in your demo cluster.



Wait! The rest of this quick start requires a remote Kubernetes cluster. Before continuing, activate a [free demo cluster](#) in step 1 above.

2. Try the Emojivoto application

getambassador.io/docs/telepresence/latest/quick-start/demo-node

ON THIS PAGE

- [Get a free remote cluster](#)
- [Try the Emojivoto application](#)
- [Set up your local development](#)
- [Install Telepresence](#)
- [Core concepts](#)
- [How do I...](#)
- [Technical reference](#)
- [FAQs](#)
- [Troubleshooting](#)
- [Community](#)
- [Release Notes](#)

3. Set up your local development environment

We'll set up a development environment locally on your workstation. We'll then use Telepresence to connect this local development environment to the remote Kubernetes cluster. To save time, the development environment we'll use is pre-packaged as a Docker container.

1 Run the Docker container locally:

GNU/Linux macOS Windows

Terminal

```
$ docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET  
Connected to context telepresence-demo (https://104.198.129.180)  
emoji      : ready to intercept (traffic-agent not yet instal  
web        : ready to intercept (traffic-agent not yet instal  
voting     : ready to intercept (traffic-agent not yet instal  
web-app-778477c59c: ready to intercept (traffic-agent not yet instal
```

```
● ● ● danielbryant — com.docker.cli • docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
Last login: Thu Oct  7 16:15:58 on ttys000  
✓ ~ (:|✓) (* |docker-desktop:default) % docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJjY2Q5YmItMmQzS000DVlWE0YjUtYTIyYzBkOWI0MjUx0mp5Z2pueE9jaU50Y0RhT3NCcWdnS1dLQXFoQTlwiVFI5ZkNHcA== -v ~/Library/Application\ Support:/root/.host_config datawire/demoemojivoto  
latest: Pulling from datawire/demoemojivoto  
4e9f2cdf4387: Pull complete  
ca42e636c499: Pull complete  
754f1dfb7a29: Pull complete  
327195eeef55a: Pull complete  
Digest: sha256:d5c05ef17d2df4352878d7725edbb918e858023273a371be057c3183e3d12470  
Status: Downloaded newer image for datawire/demoemojivoto:latest  
Launching Telepresence Root Daemon  
Launching Telepresence User Daemon  
Connected to context default (https://34.123.107.52)  
emoji          : ready to intercept (traffic-agent not yet installed)  
voting         : ready to intercept (traffic-agent not yet installed)  
web            : ready to intercept (traffic-agent not yet installed)  
web-app-57bc7c4959: ready to intercept (traffic-agent not yet installed)  
/opt/emojivoto/emojivoto-web-app #
```

K8s continuous delivery 101



Velocity (& stability) is key to business success

“Continuous delivery is achieved when stability and speed can satisfy business demand.

Discontinuous delivery occurs when stability and speed are insufficient.”

- Steve Smith (@SteveSmithCD)

Velocity (& stability) is key to business success

“Continuous delivery is achieved when **stability** and **speed** can satisfy business demand.

Discontinuous delivery occurs when stability and speed are insufficient.”

- Steve Smith (@SteveSmithCD)

Velocity (& stability) is key to business success

“Continuous delivery is achieved when **stability** and **speed** can satisfy **[your]** business demand.

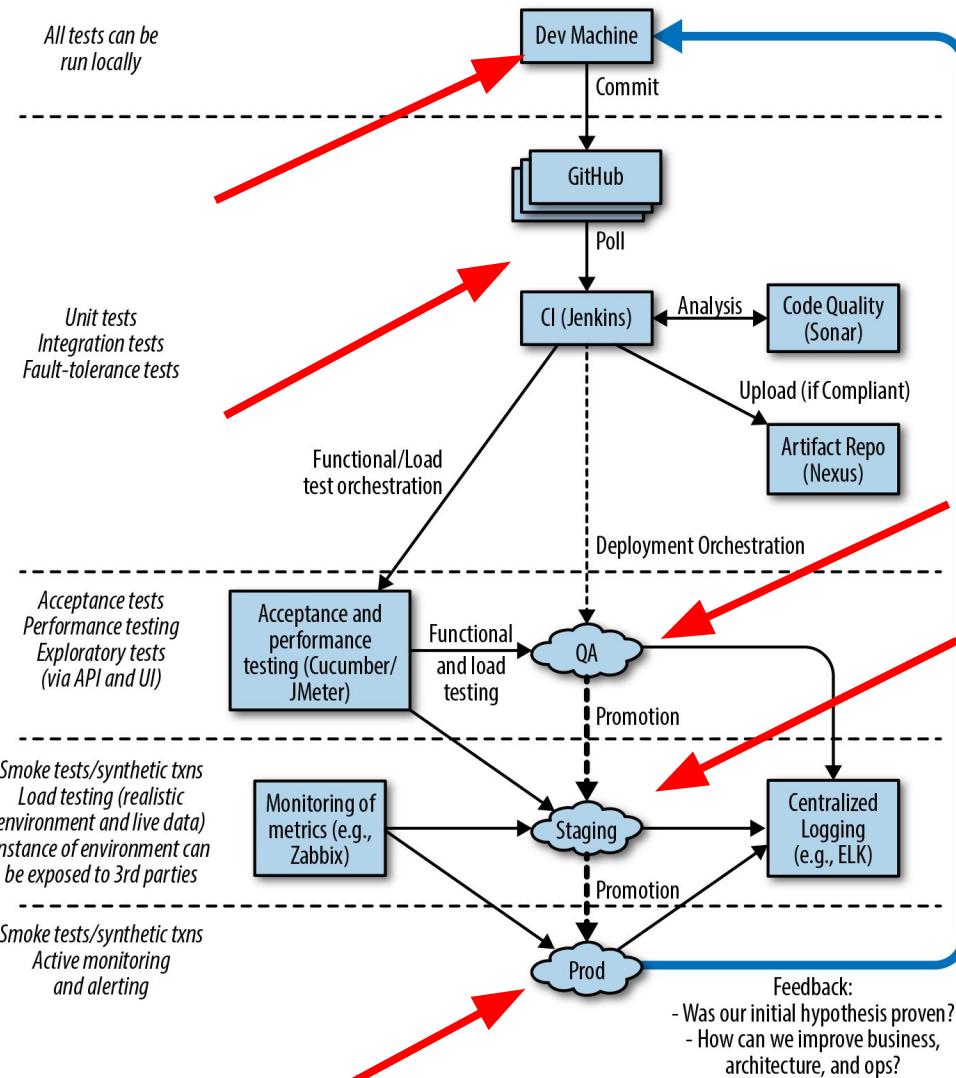
Discontinuous delivery occurs when stability and speed are insufficient.”

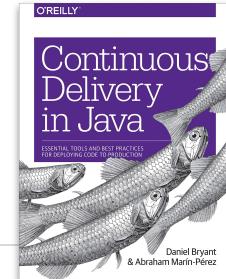
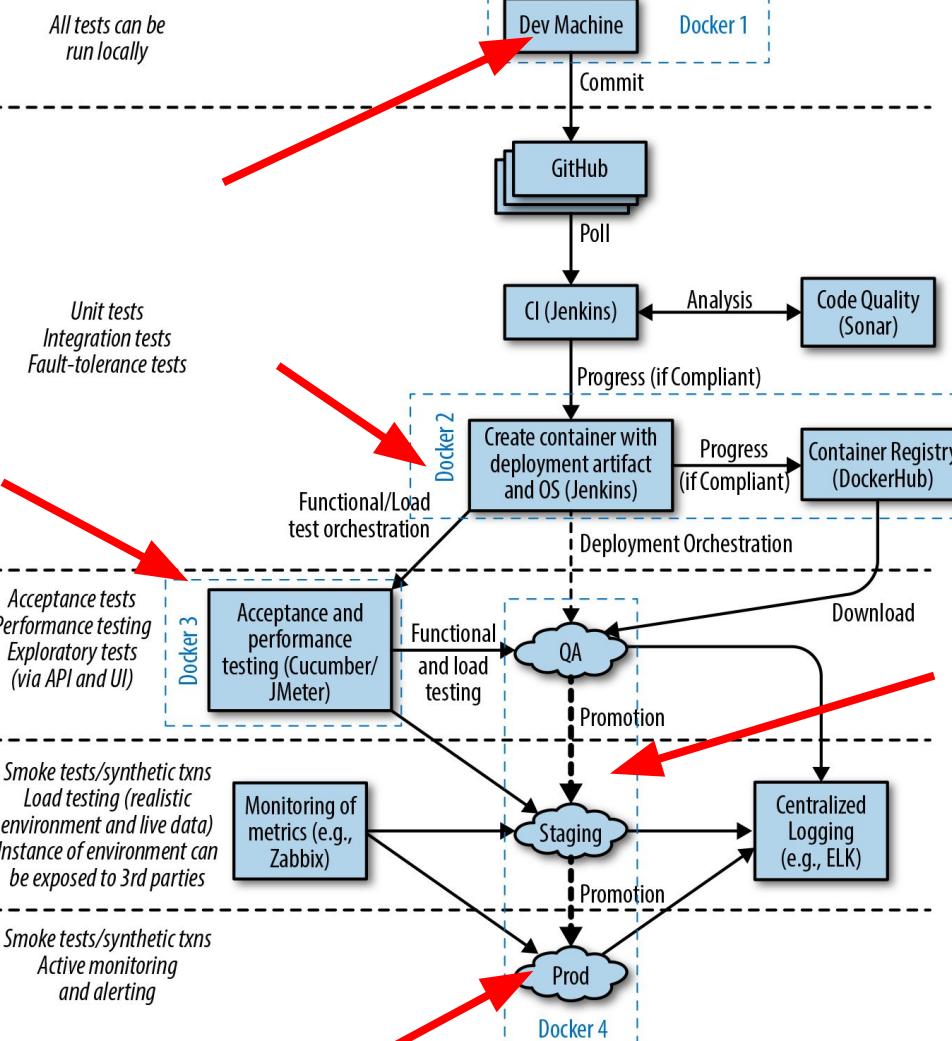
- Steve Smith (@SteveSmithCD)

CI, CD, CD... WAT?

- Continuous integration (CI) is a methodology in which developers test and build (integrate) their code changes as often as possible.
- Continuous delivery (CD) is the capability of releasing code at any time. CD assumes that your code has passed the CI pipeline and any tests that you deem necessary (such as smoke testing, QA, and load testing).
- Continuous deployment takes continuous delivery a step further by automatically releasing the application once it passes the required tests.

cloud.google.com/architecture/addressing-continuous-delivery-challenges-in-a-kubernetes-world





CI / CD: Definitions

- **Artifact**
 - The application unit e.g. JAR, ZIP file, container
 - The usual output of a CI pipeline is an artifact in an artifact storage system
- **Version**
 - An annotated instance of an artifact
 - Can be semver e.g. major.minor.path (1.4.2)
- **Environment**
 - The infrastructure or set of computing, networking, and storage resources on which you deploy your application.

CI / CD: Definitions [in K8s]

- **Artifact: Container**
 - The application unit e.g. JAR, ZIP file, container
 - The usual output of a CI pipeline is an artifact in an artifact storage system
- **Version: Container tags (and config)**
 - An annotated instance of an artifact
 - Can be semver e.g. major.minor.path (1.4.2)
- **Environment: Cluster and namespace**
 - The infrastructure or set of computing, networking, and storage resources on which you deploy your application.

CI / CD: Definitions [in K8s]

- Four main K8s objects to deploy an application on Kubernetes:
 - The [ReplicaSet](#) object lets you specify the number of [Pods](#) of your app
 - The [Deployment](#) object helps automate ReplicaSets, for example, by gradually replacing an old ReplicaSet with a new one.
 - The [StatefulSet](#)...
 - The [DaemonSet](#)...
- Configuration
 - ConfigMap
 - Secrets

Canary releasing



Progressive delivery

- Very much like continuous delivery
 - Some argue it's the same...
 - Clearly separate deploy and release
- Incrementally release features
 - Rollout changes to subset of users
 - Automated (observability) feedback
- Mechanisms
 - Blue/green
 - Canary releases
 - Dark launch

The screenshot shows a blog post titled "Towards Progressive Delivery" by James Governor. The post discusses the evolution of technology and the shift from monolithic to microservices architectures. It highlights the importance of observability and automation in modern software development. The sidebar on the right lists recent posts and provides options to subscribe via email or RSS feed.

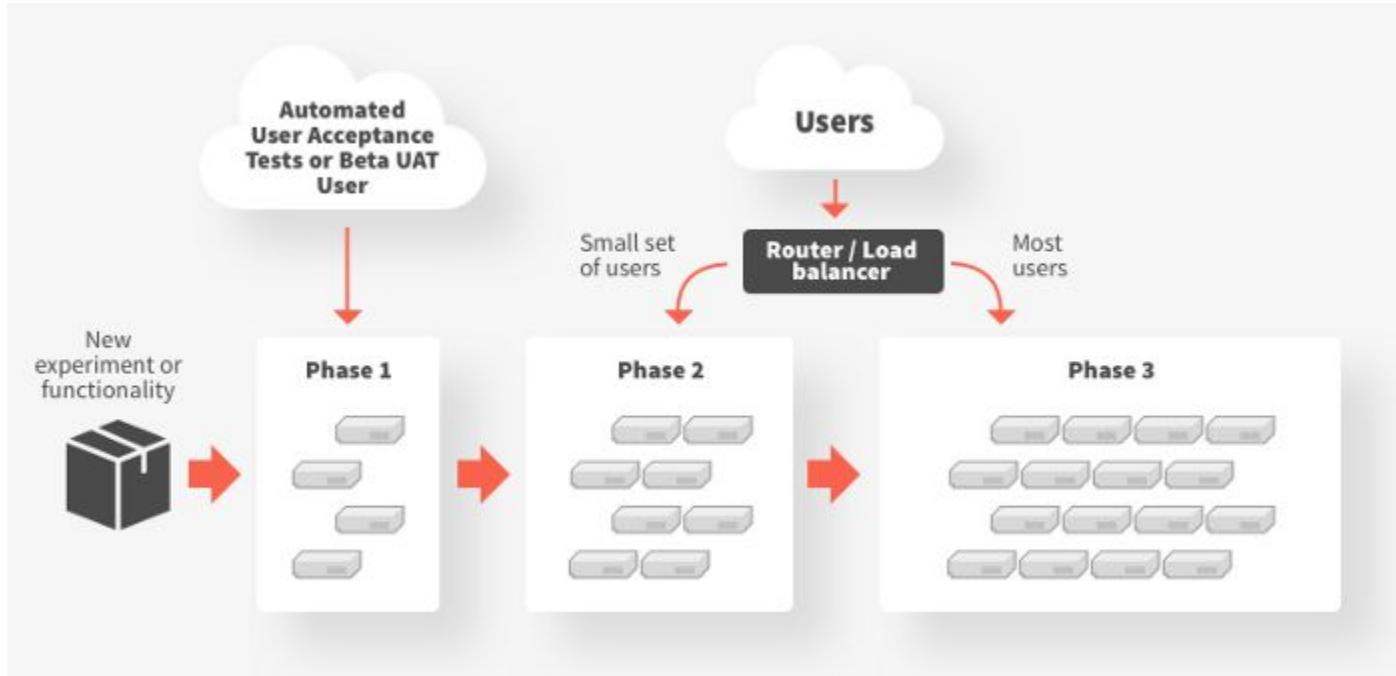
Towards Progressive Delivery
By James Governor | [About This Post](#) | August 6, 2018

At RedMonk we generally try to avoid coining up with new terms for technologies and trends – after all, while there is an available term in common use why not just adopt it? The pragmatic approach means we end up using terms that were kind of silly (api, micro, service, cloud, etc.) but it also means we avoid coining up with carry little numbers like “high performance application platform as a service” (hpaps).

The balance of money has changed a lot since we launched the firm, as the shape of the industry has. Tech today is a lot more playful than it was when it was driven by enterprise technology vendors. The internet has changed a lot, and it has certainly changed how names for things grow and spread. Developers are the new longhorns. As one company complains wistfully that a new term is born, another is adopting and propagating it with glee.

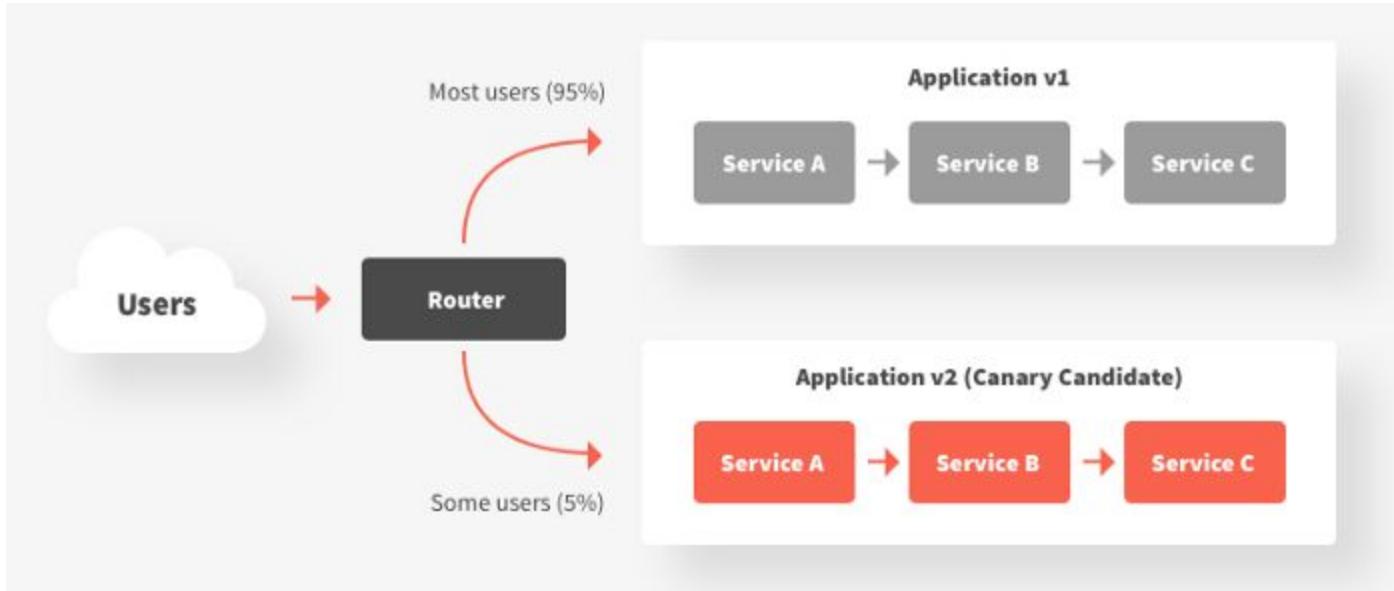
redmonk.com/jgovernor/2018/08/06/towards-progressive-delivery/

Canary releasing



<https://blog.getambassador.io/cloud-native-patterns-canary-release-1cb8f82d371a>

Canary releasing



<https://blog.getambassador.io/cloud-native-patterns-canary-release-1cb8f82d371a>

GitOps

What is GitOps?

1. An operating model for Kubernetes and other cloud native technologies, providing a set of best practices that unify Git deployment, management and monitoring for containerized clusters and applications.
2. A path towards a developer experience for managing applications; where end-to-end CI/CD pipelines and Git workflows are applied to both operations, and development.

[Weaveworks Guide to GitOps](#)

GitOps 101

“Describe the desired state of the whole system using a declarative specification for each environment.

- A git repo is the single source of truth for the desired state of the whole system.
- All changes to the desired state are Git commits.
- All specified properties of the cluster are also observable in the cluster, so that we can detect if the desired and observed states are the same (converged) or different (diverged). ”

www.weave.works/blog/what-is-gitops-really

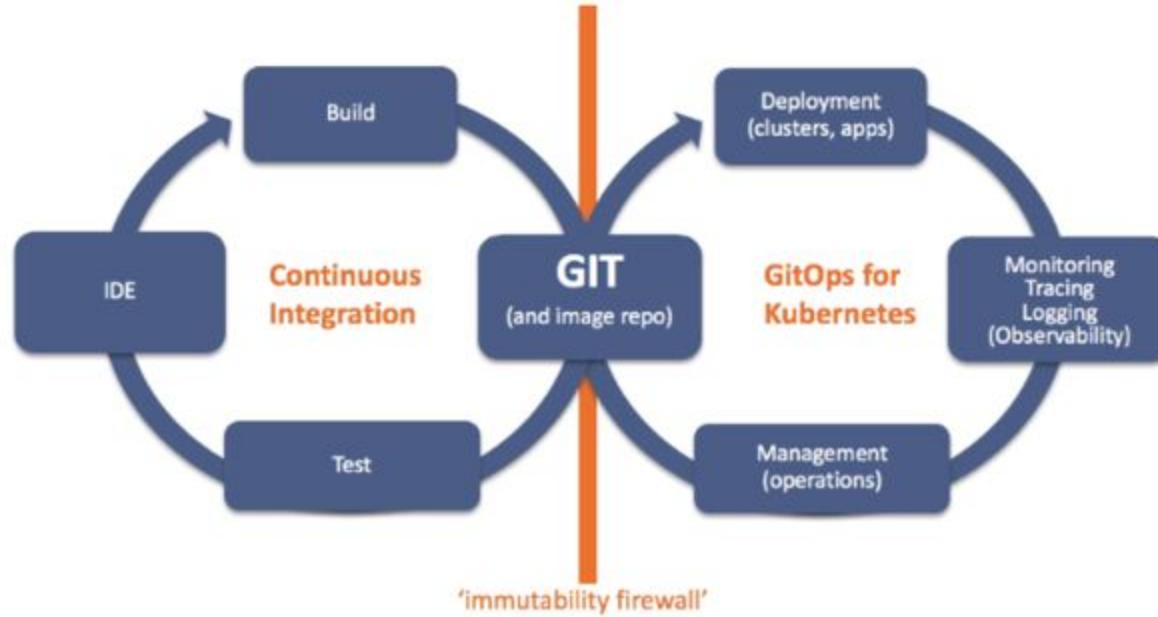
GitOps 101

“When the desired and observed states are not the same then:

- There is a convergence mechanism to bring the desired and observed states in sync both eventually, and autonomically. Within the cluster, this is Kubernetes.
- This is triggered immediately with a “change committed” alert.
- After a configurable interval, an alert “diff” may also be sent if the states are divergent.”

www.weave.works/blog/what-is-gitops-really

GitOps 101



www.weave.works/blog/what-is-gitops-really

If manual GitOps was a thing (it's not!)

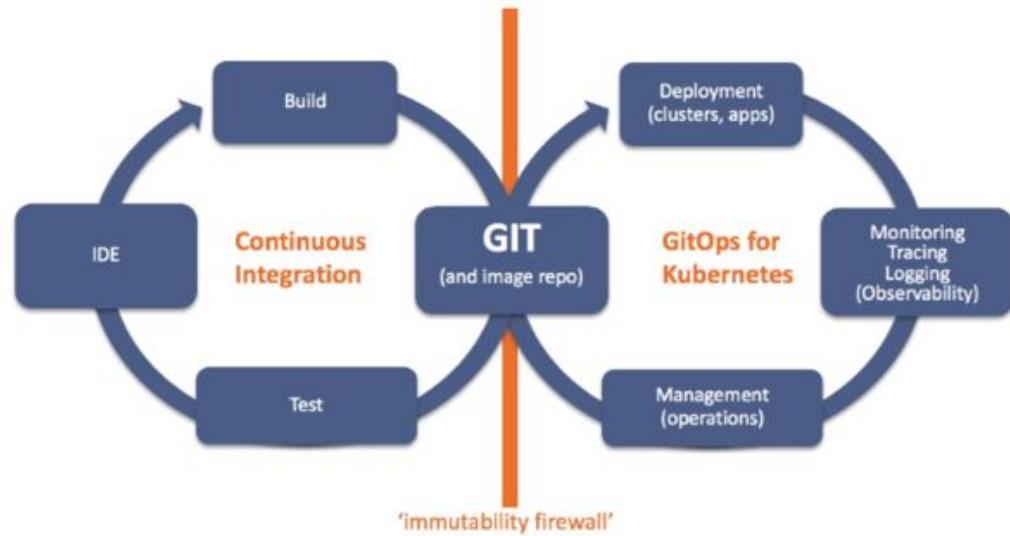
```
kubectl diff -f .
```

```
# depending on results from diff
```

```
kubectl apply -f .
```

```
# or
```

```
kubectl delete -f .
```

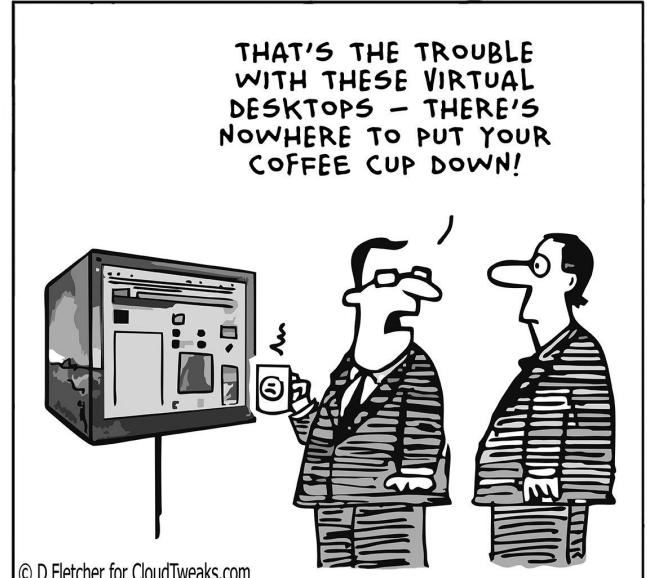


10 minute (coffee) break



But don't go away! After this we're hands on with

- Argo Rollouts for canary releasing
- Argo CD for GitOps



© D.Fletcher for CloudTweaks.com

Using Argo Rollouts for canarying

argoproj.github.io/argo-rollouts/

Introduces Rollout CRD

- Think Deployment++

The screenshot shows a web browser displaying the official Argo Rollouts documentation at argoproj.github.io/argo-rollouts/. The page has a green header with the title "Argo Rollouts - Kubernetes Progressive Delivery Controller". On the left, there is a sidebar with a navigation menu for the "Argo Rollouts - Kubernetes Progressive Delivery Controller" documentation, including sections like Overview, Installation, Concepts, Architecture, Getting Started, Dashboard, Rollout, Traffic Management, Analysis, Experiments, Notifications, Kubectl Plugin, Best Practices, Migrating, FAQ, Security, Roadmap, Contributing, Releases, Roadmap, and Blog. To the right of the sidebar, the main content area starts with a section titled "What is Argo Rollouts?". It describes Argo Rollouts as a "Kubernetes controller" and set of "CRDs" that provide advanced deployment capabilities such as blue-green, canary, canary analysis, experimentation, and progressive delivery features to Kubernetes. Below this, there is a section about integrating with ingress controllers and service meshes, and another about providers verifying KPIs. At the bottom of the content area, there is a link to a demonstration video and a "Why Argo Rollouts?" section featuring a purple background with the Argo Rollouts logo (an orange octopus) and the text "Argo Rollouts".

Using Argo Rollouts for canary releasing

```
# Assuming you have your remote cluster and local container up and running:  
# www.getambassador.io/docs/telepresence/latest/quick-start/demo-node/  
  
# Save the keyboards!  
alias k=kubectl  
  
  
# Look around at argoctl and argorollouts namespaces  
  
kubectl get ns  
  
kubectl get svc -n argo-rollouts  
  
kubectl get svc -n argoctl
```

```
danielbryant — com.docker.cli - docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
web-app-57bc7c4959: ready to intercept (traffic-agent not yet installed)  
/opt/emojivoto/emojivoto-web-app # alias k=kubectl  
/opt/emojivoto/emojivoto-web-app # k get ns  
NAME STATUS AGE  
default Active 166m  
kube-system Active 166m  
kube-public Active 166m  
kube-node-lease Active 166m  
ambassador Active 84m  
argo-rollouts Active 84m  
argocd Active 84m  
emojivoto Active 84m  
/opt/emojivoto/emojivoto-web-app # k get svc -n argo-rollouts  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
argo-rollouts-metrics ClusterIP 10.43.41.161 <none> 8090/TCP 85m  
/opt/emojivoto/emojivoto-web-app # k get svc -n argocd  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
argocd-dex-server ClusterIP 10.43.93.104 <none> 5556/TCP,5557/TCP,5558/TCP 85m  
argocd-metrics ClusterIP 10.43.212.171 <none> 8082/TCP 85m  
argocd-redis ClusterIP 10.43.189.93 <none> 6379/TCP 85m  
argocd-repo-server ClusterIP 10.43.232.196 <none> 8081/TCP,8084/TCP 85m  
argocd-server ClusterIP 10.43.61.68 <none> 80/TCP,443/TCP 85m  
argocd-server-metrics ClusterIP 10.43.106.201 <none> 8083/TCP 85m  
/opt/emojivoto/emojivoto-web-app #
```



Install Argo Rollouts CLI tool

```
# Install Argo Rollouts CLI tool (in the container)

curl -LO
https://github.com/argoproj/argo-rollouts/releases/latest/download/kubectl-argo-rollouts
-linux-amd64

chmod +x ./kubectl-argo-rollouts-linux-amd64

mv ./kubectl-argo-rollouts-linux-amd64 /usr/local/bin/kubectl-argo-rollouts

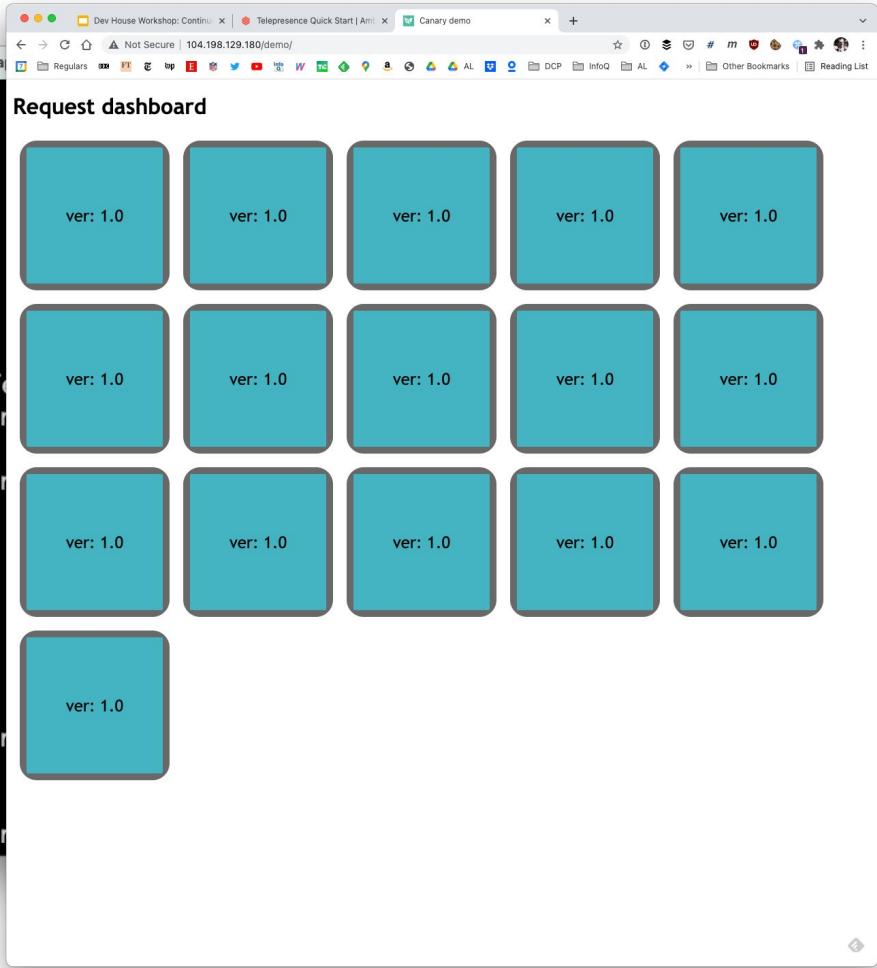
kubectl argo rollouts version
```

```
danielbryant — com.docker.cli • docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
argocd-dex-server      ClusterIP  10.43.93.104    <none>        5556/TCP,5557/TCP,5558/TCP  85m  
argocd-metrics         ClusterIP  10.43.212.171   <none>        8082/TCP                  85m  
argocd-redis           ClusterIP  10.43.189.93   <none>        6379/TCP                  85m  
argocd-repo-server     ClusterIP  10.43.232.196   <none>        8081/TCP,8084/TCP          85m  
argocd-server          ClusterIP  10.43.61.68    <none>        80/TCP,443/TCP            85m  
argocd-server-metrics  ClusterIP  10.43.106.201   <none>        8083/TCP                  85m  
/opt/emojivoto/emojivoto-web-app # curl -L0 https://github.com/argoproj/argo-rollouts/releases/latest/download/kubectl-argo-rollouts-linux-amd64  
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current  
                                         Dload  Upload   Total  Spent  Left  Speed  
100  166  100  166    0     0    61      0  0:00:02  0:00:02  ---:---   61  
100  641  100  641    0     0   221      0  0:00:02  0:00:02  ---:---   221  
100 61.3M  100 61.3M   0     0  3448k      0  0:00:18  0:00:18  ---:--- 4171k  
/opt/emojivoto/emojivoto-web-app # chmod +x ./kubectl-argo-rollouts-linux-amd64  
/opt/emojivoto/emojivoto-web-app # mv ./kubectl-argo-rollouts-linux-amd64 /usr/local/bin/kubectl-argo-rollouts  
/opt/emojivoto/emojivoto-web-app # kubectl argo rollouts version  
kubectl-argo-rollouts: v1.0.7+1d8052e  
  BuildDate: 2021-09-23T23:17:03Z  
  GitCommit: 1d8052ec0dc4358b8b29831aff2eb13f78f647c1  
  GitTreeState: clean  
  GoVersion: go1.16.3  
  Compiler: gc  
  Platform: linux/amd64  
/opt/emojivoto/emojivoto-web-app #
```

Apply the demo code in your cluster

```
wget https://github.com/kostis-codefresh/summer-of-k8s-app-manifests/archive/refs/heads/main.zip  
unzip main.zip  
  
cd summer-of-k8s-app-manifests-main/  
  
kubectl create namespace demo  
kubectl apply -f . -n demo  
  
kubectl get svc ambassador -n ambassador  
  
# view via your web browser http://<Ambassador Edge Stack External IP>/demo/
```

```
'main.zip' saved
/opt/emojivoto/emojivoto-web-app # unzip main.zip
Archive: main.zip
  creating: summer-of-k8s-app-manifests-main/
  inflating: summer-of-k8s-app-manifests-main/README.md
  inflating: summer-of-k8s-app-manifests-main/mapping.yaml
  inflating: summer-of-k8s-app-manifests-main/resolver.yaml
  inflating: summer-of-k8s-app-manifests-main/rollout.yaml
  inflating: summer-of-k8s-app-manifests-main/service.yaml
/opt/emojivoto/emojivoto-web-app # cd summer-of-k8s-app-manifests-main
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main
namespaces/demo created
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main
mapping.getambassador.io/summer-k8s-mapping created
mapping.getambassador.io/summer-k8s-mapping-stable created
mapping.getambassador.io/summer-k8s-mapping-unstable created
kubernetesendpointresolver.getambassador.io/endpoint created
rollout.argoproj.io/summer-k8s-rollout created
service/summer-k8s-service-stable created
service/summer-k8s-service-canary created
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP
ambassador  LoadBalancer  10.43.104.240  104.198.129.180
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main
```



Update rollout.yaml

```
vi rollout.yaml

# change container image v1 to v2

containers:

- name: webserver-simple

  image: kostiscodefresh/summer-of-k8s-app:v2

  imagePullPolicy: Always

ports:

- containerPort: 8080

# use your cursor to move around in vi, shift+i to edit, and ESC + ":wq" to save and quit
```

```
● ○ ● com.docker.cli - docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
---  
apiVersion: argoproj.io/v1alpha1  
kind: Rollout  
metadata:  
  name: summer-k8s-rollout  
spec:  
  revisionHistoryLimit: 1  
  replicas: 10  
  selector:  
    matchLabels:  
      app: summer-k8s-app  
  template:  
    metadata:  
      labels:  
        app: summer-k8s-app  
    spec:  
      containers:  
      - name: webserver-simple  
        image: kostiscodefresh/summer-of-k8s-app:v2  
        imagePullPolicy: Always  
        ports:  
        - containerPort: 8080  
  strategy:  
I rollout.yaml [Modified] 19/36 52%
```

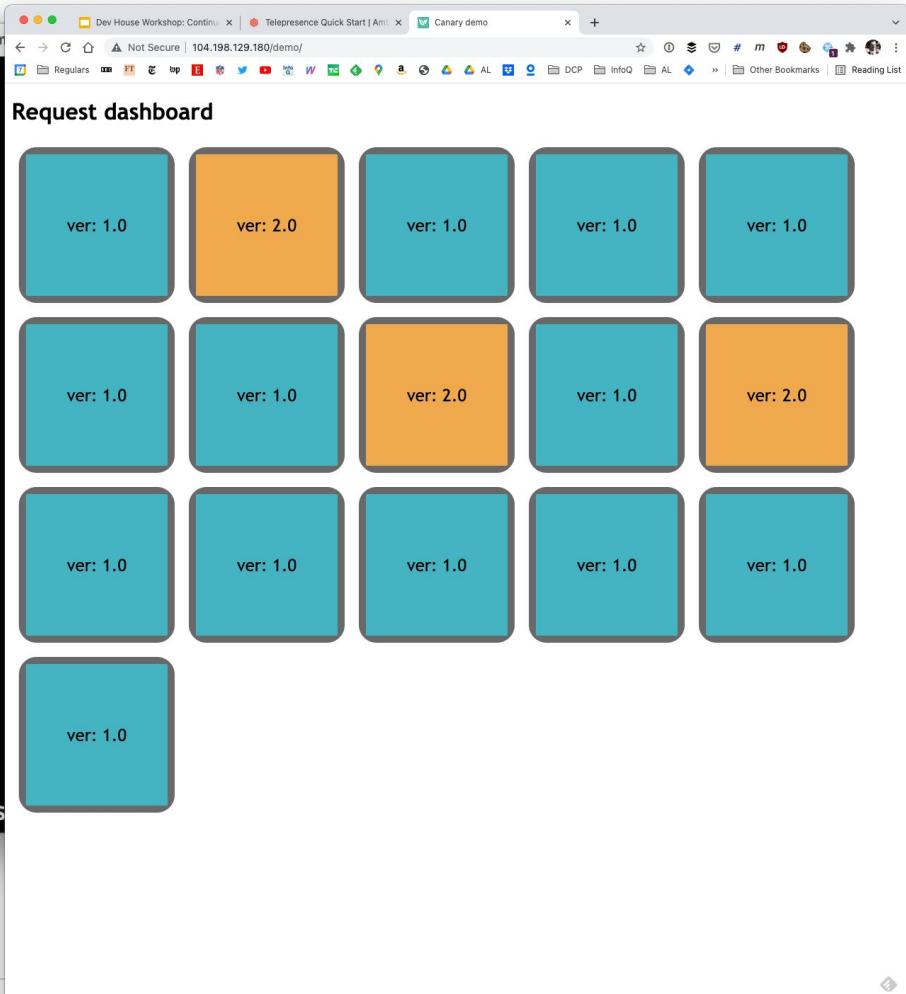
Apply changes and observe

```
kubectl apply -f rollout.yaml -n demo
```

```
kubectl argo rollouts get rollout summer-k8s-rollout -n demo
```

```
Updated:      3
Ready:       13
Available:   13

NAME
└─ summer-k8s-rollout
    ├─ # revision:2
    │   └─ summer-k8s-rollout-8874dc9
    │       ├─ summer-k8s-rollout-8874dc9-j2l16
    │       ├─ summer-k8s-rollout-8874dc9-nrhm6
    │       └─ summer-k8s-rollout-8874dc9-nwlkp
    └─ # revision:1
        └─ summer-k8s-rollout-7b46df69d
            ├─ summer-k8s-rollout-7b46df69d-6dkv4
            ├─ summer-k8s-rollout-7b46df69d-cglkz
            ├─ summer-k8s-rollout-7b46df69d-pdczb
            ├─ summer-k8s-rollout-7b46df69d-6w9s7
            ├─ summer-k8s-rollout-7b46df69d-f57xx
            ├─ summer-k8s-rollout-7b46df69d-f87cg
            ├─ summer-k8s-rollout-7b46df69d-fzjl5
            ├─ summer-k8s-rollout-7b46df69d-jvr7g
            ├─ summer-k8s-rollout-7b46df69d-qg5r9
            └─ summer-k8s-rollout-7b46df69d-z4xhf
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifes
```



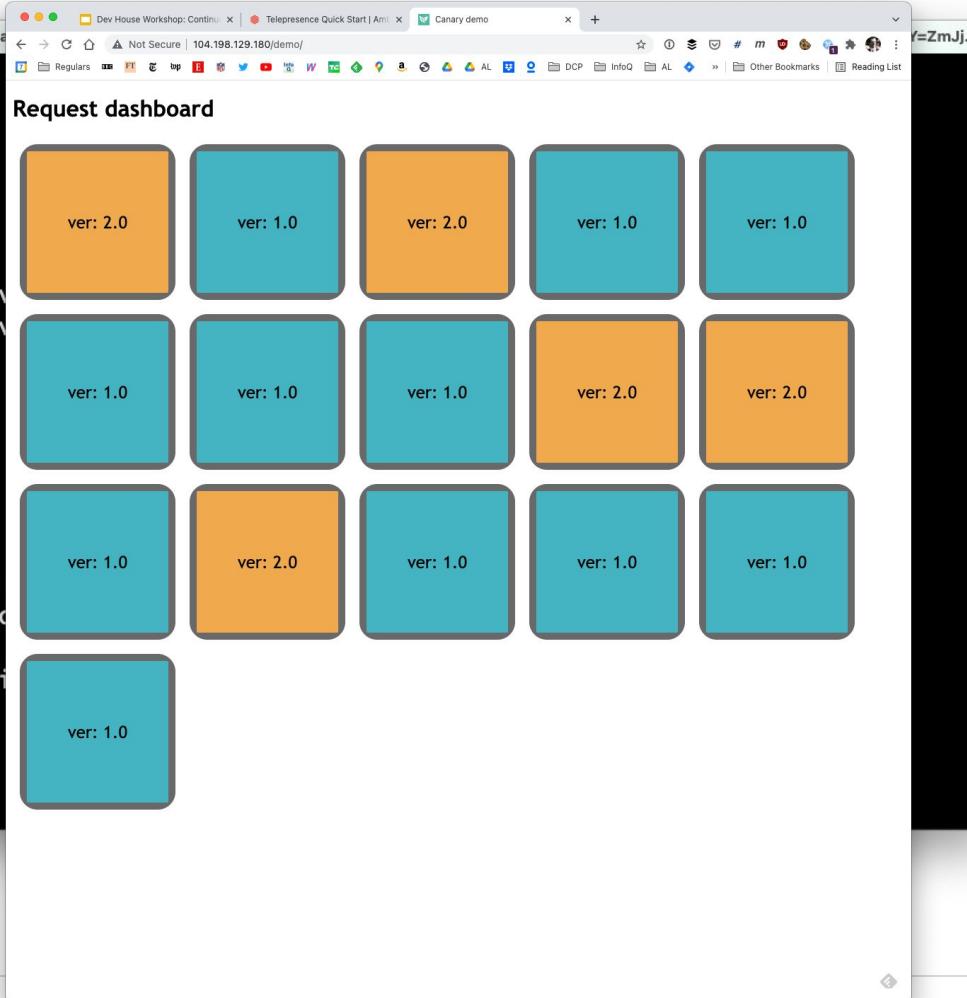
Promote your canary

```
kubectl argo rollouts promote summer-k8s-rollout -n demo
```

```
kubectl argo rollouts get rollout summer-k8s-rollout -n demo -w
```

```
# Can also create a new terminal to watch $ docker exec -it <container_id> /bin/sh
```

```
danielbryant — com.docker.cli • docker run -p8083:8083 -p8080:8080 --name ambassador -v /var/run/docker.sock:/var/run/docker.sock ambassador:latest  
Namespace: demo  
Status: || Paused  
Message: CanaryPauseStep  
Strategy: Canary  
Step: 3/6  
SetWeight: 60  
ActualWeight: 60  
Images: kostiscodefresh/summer-of-k8s-app:v2.0  
kostiscodefresh/summer-of-k8s-app:v1.0  
Replicas:  
Desired: 10  
Current: 16  
Updated: 6  
Ready: 16  
Available: 16  
  
NAME  
C summer-k8s-rollout  
└─# revision:2  
| └─ summer-k8s-rollout-8874dc9  
|   ├─ summer-k8s-rollout-8874dc9-j2ll6  
|   ├─ summer-k8s-rollout-8874dc9-nrhm6  
|   ├─ summer-k8s-rollout-8874dc9-nwlkp  
|   └─ summer-k8s-rollout-8874dc9-5vd4m
```



Reset the demo app

```
kubectl delete -f . -n demo
```

Using Argo CD for GitOps

argo-cd.readthedocs.io/en/stable/

The screenshot shows a web browser displaying the Argo CD documentation at argo-cd.readthedocs.io/en/stable/. The page title is "Argo CD - Declarative GitOps CD for Kubernetes". The left sidebar contains a navigation menu with links to Overview, Understand The Basics, Core Concepts, Getting Started, Operator Manual, User Guide, Developer Guide, FAQ, Security Considerations, Support, Roadmap, Releases, and Blog. The main content area features a large heading "Overview" and a sub-section "What Is Argo CD?". It includes a screenshot of the Argo CD UI showing a "Syncing" dialog over a list of application components like "application-controller", "argocd-server", and "argocd-repo-server". To the right of the main content is a "Table of contents" sidebar with links to What Is Argo CD?, Why Argo CD?, Getting Started, Quick Start, How it works, Architecture, Features, Development Status, and Adoption.

Argo CD - Declarative GitOps CD for Kubernetes

Overview

Understand The Basics

Core Concepts

Getting Started

Operator Manual

User Guide

Developer Guide

FAQ

Security Considerations

Support

Roadmap

Releases

Blog

What Is Argo CD?

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.

APPLICATION DETAILS

Argo CD - Declarative GitOps CD for Kubernetes

Synchronizing application manifests from https://github.com/argoproj/deployments.git

Revision: 1

Syncing

CANCEL

What Is Argo CD?

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.

Argo CD synchronizes application manifests from a Git repository using Kubernetes Deployments. It provides a user interface for managing applications, monitoring their status, and triggering syncs.

Why Argo CD?

Application definitions, configurations, and environments should be declarative and version controlled. Application deployment and lifecycle management should be automated, auditable, and easy to understand.

Getting Started

Quick Start

Table of contents

- What Is Argo CD?
- Why Argo CD?
- Getting Started
- Quick Start
- How it works
- Architecture
- Features
- Development Status
- Adoption

Using Argo CD for GitOps

```
kubectl edit deploy argocd-server -n argocd
```

```
# Add directly below:
```

```
- command:
```

```
  - argocd-server
```

```
# the following (with correct spacing at the beginning of the line!)
```

```
  - --insecure
```

```
  - --rootpath=/argo-cd
```

```
# ":wq" to save the changes and quit
```

```
danielbryant — com.docker.cli - docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
  matchLabels:  
    app.kubernetes.io/name: argocd-server  
  topologyKey: kubernetes.io/hostname  
  weight: 100  
- podAffinityTerm:  
  labelSelector:  
    matchLabels:  
      app.kubernetes.io/part-of: argocd  
    topologyKey: kubernetes.io/hostname  
    weight: 5  
containers:  
command:  
- argocd-server  
- --insecure  
- --rootpath=/argo-cd  
- --staticassets  
- /shared/app  
image: quay.io/argoproj/argocd:v2.0.4  
imagePullPolicy: Always  
livenessProbe:  
  failureThreshold: 3  
  httpGet:  
    path: /healthz?full=true  
- /tmp/kubectl-edit-256237874.yaml 56/153 36%
```

Configuring Edge Stack for access

```
vi argocd-mapping.yaml

# Add contents to file

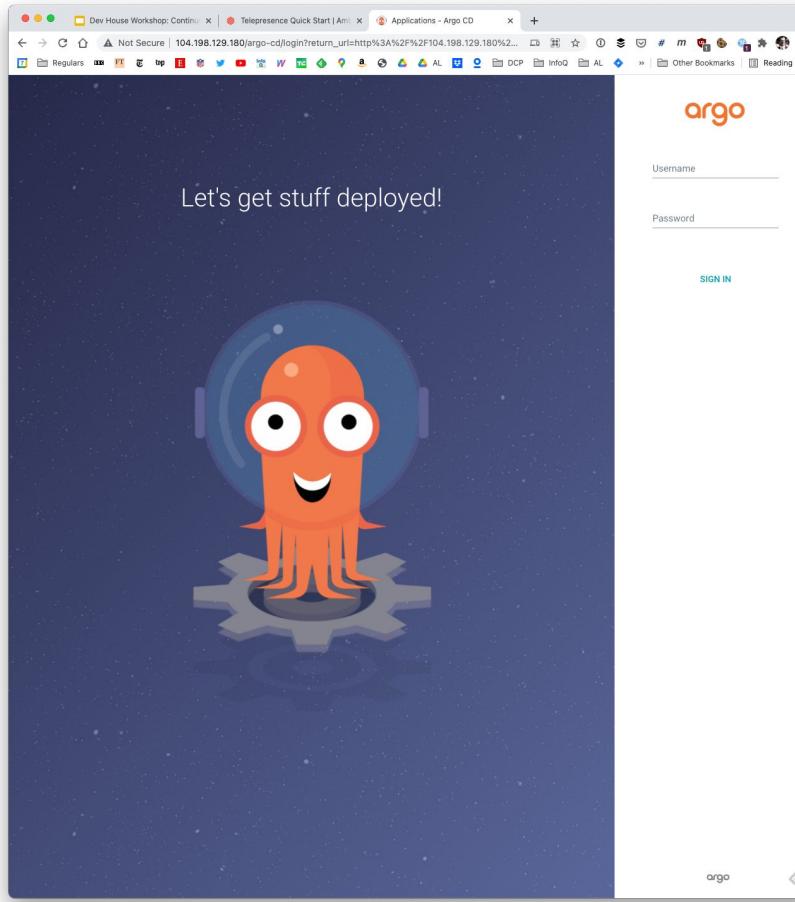
apiVersion: getambassador.io/v2
kind: Mapping
metadata:
  name: argocd-server
  namespace: argocd
spec:
  prefix: /argo-cd
  rewrite: /argo-cd
  service: argocd-server.argocd:443

# And apply the file
kubectl apply -f argocd-mapping.yaml
```

```
danielbryant — com.docker.cli - docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJj...  
|   |   summer-k8s-rollout-8874dc9-8h86k   Pod      ✓ Running  6s    ready:1/1  
|   |   summer-k8s-rollout-8874dc9-n9zcw   Pod      ✓ Running  6s    ready:1/1  
└ # revision:1  
  └─ summer-k8s-rollout-7b46df69d   ReplicaSet ✓ Healthy  3m    stable  
    |   summer-k8s-rollout-7b46df69d-6dkv4 Pod      ✓ Running  3m    ready:1/1  
    |   summer-k8s-rollout-7b46df69d-cglkz Pod      ✓ Running  3m    ready:1/1  
    |   summer-k8s-rollout-7b46df69d-pdczb Pod      ✓ Running  3m    ready:1/1  
    |   summer-k8s-rollout-7b46df69d-6w9s7 Pod      ✓ Running  2m59s  ready:1/1  
    |   summer-k8s-rollout-7b46df69d-f57xx Pod      ✓ Running  2m59s  ready:1/1  
    |   summer-k8s-rollout-7b46df69d-f87cg Pod      ✓ Running  2m59s  ready:1/1  
    |   summer-k8s-rollout-7b46df69d-fzjl5 Pod      ✓ Running  2m59s  ready:1/1  
    |   summer-k8s-rollout-7b46df69d-jvr7g Pod      ✓ Running  2m59s  ready:1/1  
    |   summer-k8s-rollout-7b46df69d-qg5r9 Pod      ✓ Running  2m59s  ready:1/1  
    └   summer-k8s-rollout-7b46df69d-z4xhf Pod      ✓ Running  2m59s  ready:1/1  
  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main # kubectl delete ns demo  
namespace "demo" deleted  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main # kubectl create ns demo  
namespace/demo created  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main # kubectl edit deploy argocd-server -n argocd  
deployment.apps/argocd-server edited  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main # vi argocd-mapping.yaml  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main # kubectl apply -f argocd-mapping.yaml  
mapping.getambassador.io/argocd-server created  
/opt/emojivoto/emojivoto-web-app/summer-of-k8s-app-manifests-main #
```

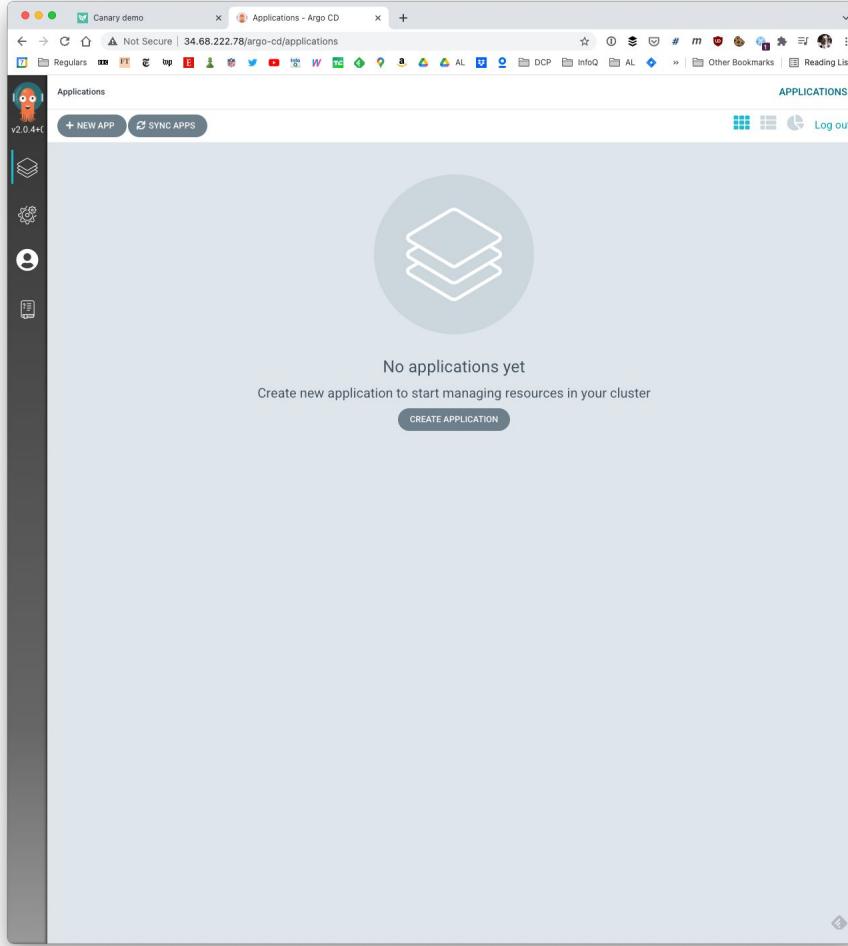


```
# http://<Ambassador Edge Stack IP>/argo-cd/
```



Login to Argo CD

```
# Username: admin  
  
# Password  
  
kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath=".data.password"  
| base64 -d ; echo " "  
  
# Watch for your browser ad blocker causing problems
```



Configure your app in Argo CD

Fork Kostis' repo into your GitHub account (to make changes)

<https://github.com/kostis-codefresh/summer-of-k8s-app-manifests>

(my copy:

<https://github.com/danielbryantuk/summer-of-k8s-app-manifests>)

Add this to Argo CD via the UI

Not Secure | 104.198.129.180/argo-cd/applications?new=%7B"apiVersion"%3A"argoproj.io%2Fv1alpha1... | Week 3 | Ship | Summer of K8s | danielbryantuk/summer-of-k8s-app-manifests | +

Regulars bp AL DCP InfoQ AL Reading List

Applications

CREATE CANCEL

GENERAL

Application Name
dev-house

Project
default

SYNC POLICY

Manual

SYNC OPTIONS

SKIP SCHEMA VALIDATION AUTO-CREATE NAMESPACE
 PRUNE LAST APPLY OUT OF SYNC ONLY
 REPLACE ⚠

PRUNE PROPAGATION POLICY: foreground

SOURCE

Repository URL
<https://github.com/danielbryantuk/summer-of-k8s-app-manifests> GIT

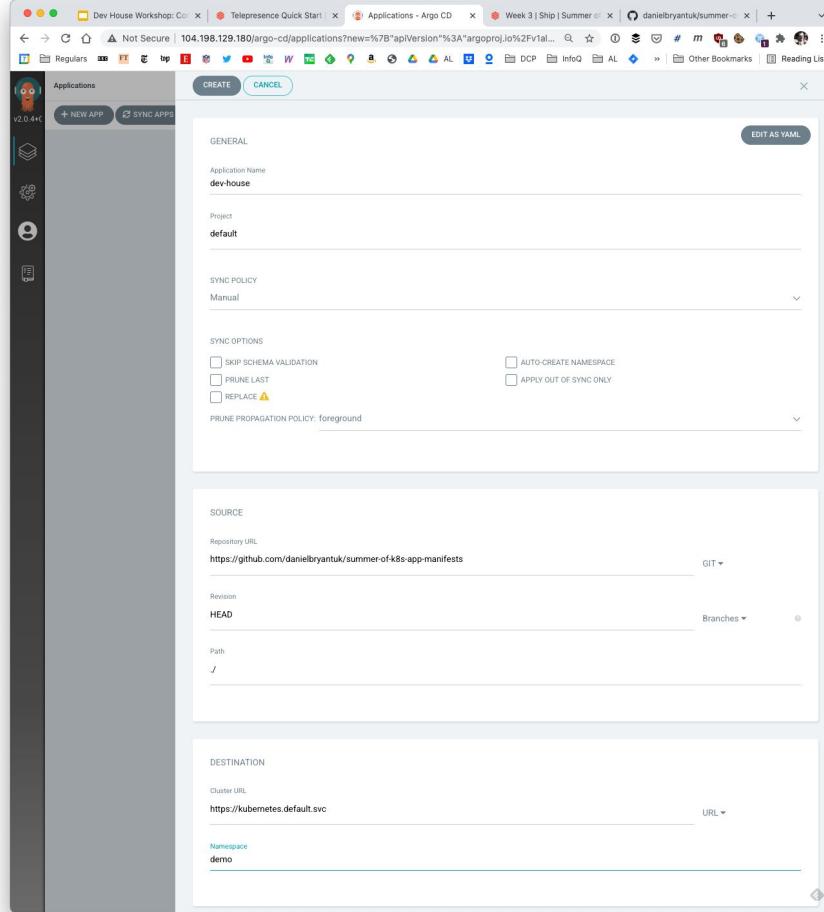
Revision
HEAD Branches

Path
/

DESTINATION

Cluster URL
<https://kubernetes.default.svc> URL

Namespace
demo

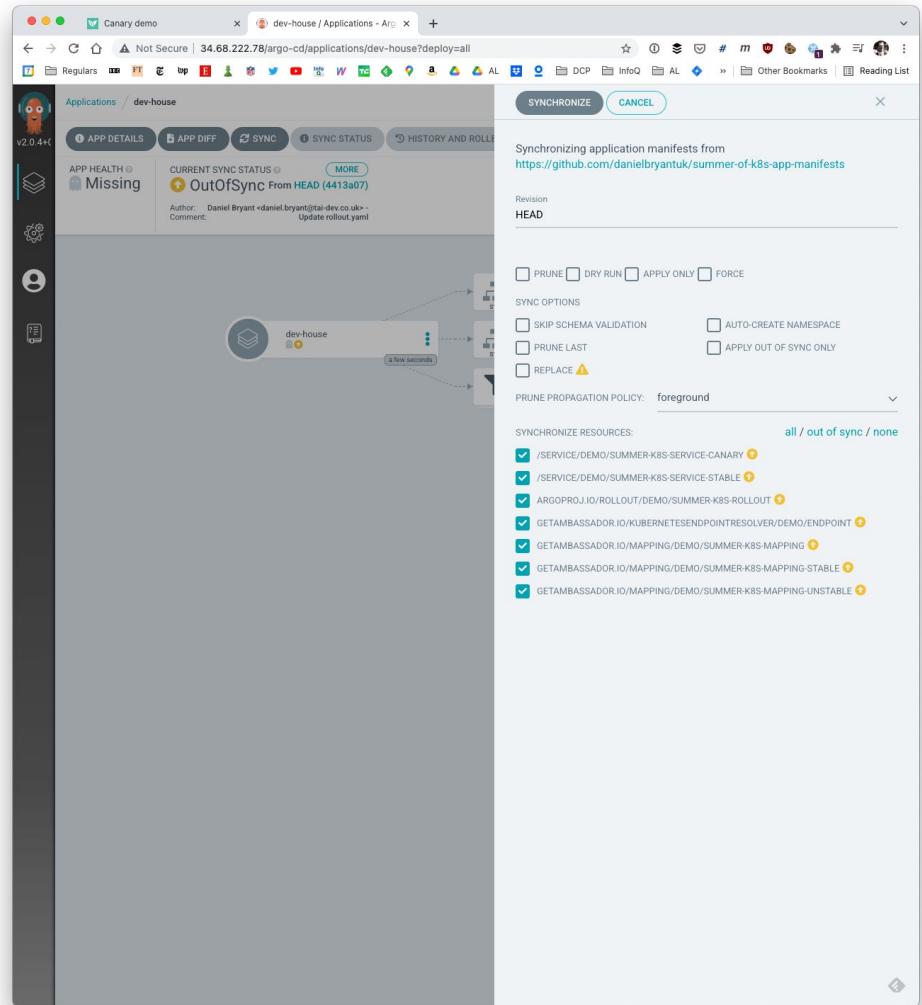


Sync(hronize)!

And view the app via browser

Want to see the cluster changes?

```
watch kubectl get all -n demo
```



Not Secure | 104.198.129.180/argo-cd/applications/dev-house

Regulars btp bp AL DCP InfoQ AL Other Bookmarks Reading List

Applications / dev-house

APPLICATION DETAILS

v2.0.4-k Missing

APP HEALTH: APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH Log out

CURRENT SYNC STATUS: OutOfSync From HEAD (bae416f)

Author: Daniel Bryant <daniel.bryant@tai-dev.co.uk> Comment: Update rollout.yaml

LAST SYNC RESULT: Sync OK To HEAD (bae416f)

Author: Daniel Bryant <daniel.bryant@tai-dev.co.uk> Comment: Update rollout.yaml

26 minutes

clear filters to show 5 additional resources

Diagram illustrating the application structure:

```
graph TD; dev-house --- summer_k8s_service_canary[summer-k8s-service-canary]; dev-house --- summer_k8s_service_stable[summer-k8s-service-stable];
```

Not Secure | 104.198.129.180/argo-cd/applications/dev-house

Regulars btp bp AL DCP InfoQ AL Other Bookmarks Reading List

Applications / dev-house

APPLICATION DETAILS

v2.0.4-k

APP HEALTH: APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH Log out

CURRENT SYNC STATUS: Synced To HEAD (bae416f)

Author: Daniel Bryant <daniel.bryant@tai-dev.co.uk> Comment: Update rollout.yaml

LAST SYNC RESULT: Sync OK To bae11cf

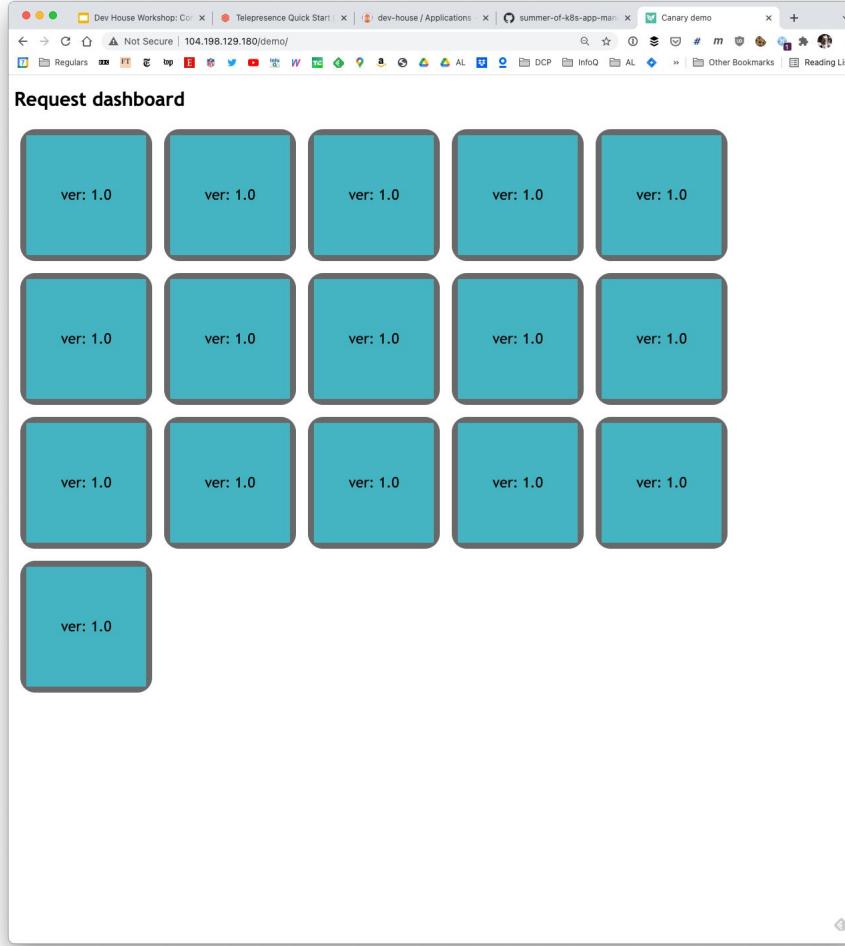
Author: Daniel Bryant <daniel.bryant@tai-dev.co.uk> Comment: Update rollout.yaml

26 minutes

clear filters to show 10 additional resources

Diagram illustrating the application structure with many more components:

```
graph TD; dev-house --- svc1[summer-k8s-service-canary]; dev-house --- svc2[summer-k8s-service-canary]; dev-house --- rollout1[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout2[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout3[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout4[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout5[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout6[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout7[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout8[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout9[summer-k8s-rollout-7b46df69d...]; dev-house --- rollout10[summer-k8s-rollout-7b46df69d...]
```



Update GitHub and synchronize

Change the container image in your fork to v2

And synchronize (note, no `kubectl` in sight!)

Dev House Workshop: Con | Telepresence Quick Start | dev-house / Applications | Editing summer-of-k8s-app-manifests | Canary demo

github.com/danielbryantuk/summer-of-k8s-app-manifests/edit/main/rollout.yaml

Regulars F DCP AL InfoQ AL DCP AL Other Bookmarks Reading List

Search or jump to... Pull requests Issues Marketplace Explore

danielbryantuk / summer-of-k8s-app-manifests Public forked from kostiscodefresh/summer-of-k8s-app-manifests

Code Pull requests Actions Projects Wiki Security Insights Settings

summer-of-k8s-app-manifests / rollout.yaml in main Cancel changes

Edit file Preview changes Spaces 2 No wrap

```
1 ---
2   apiVersion: argoproj.io/v1alpha1
3   kind: Rollout
4   metadata:
5     name: summer-k8s-rollout
6   spec:
7     revisionHistoryLimit: 1
8     replicas: 10
9     selector:
10       matchLabels:
11         app: summer-k8s-app
12     template:
13       metadata:
14         labels:
15           app: summer-k8s-app
16     spec:
17       containers:
18         - name: webserver-simple
19           image: kostiscodefresh/summer-of-k8s-app:v2
20           imagePullPolicy: Always
21         ports:
22           - containerPort: 8000
23     strategy:
24       canary:
25         stableService: summer-k8s-service-stable
26         canaryService: summer-k8s-service-canary
27         trafficRouting:
28           ambassador:
29             mappings:
30               - summer-k8s-mapping
31         steps:
32           - setWeight: 30
33           - pause: {}
34           - setWeight: 60
35           - pause: {}
```

Commit changes

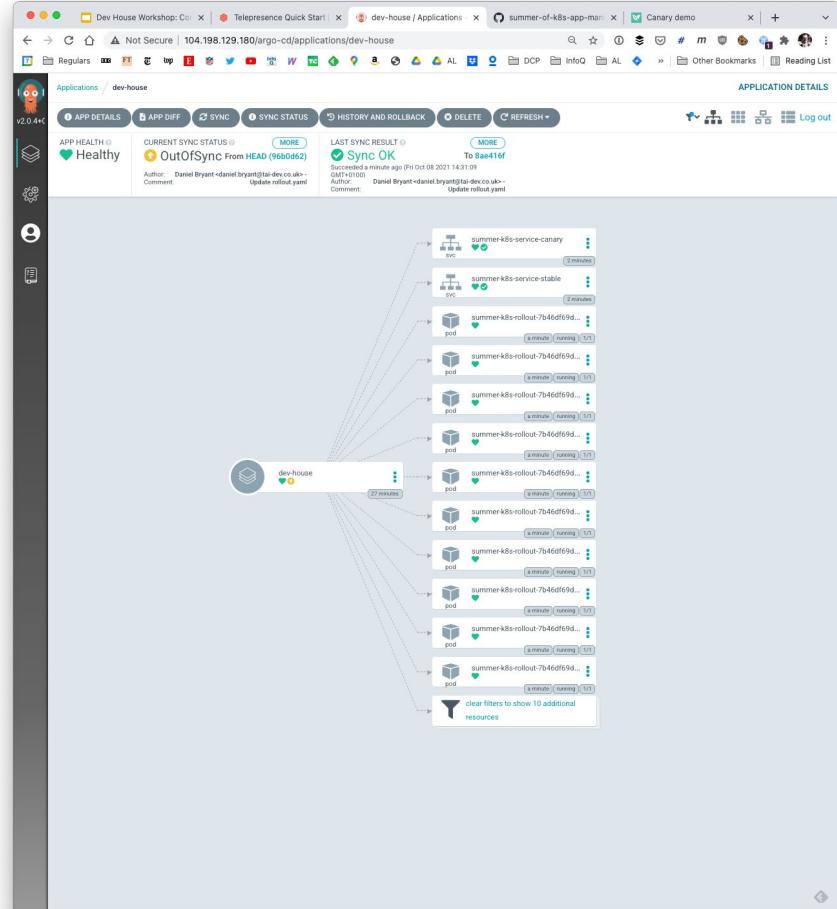
Update rollout.yaml

Add an optional extended description...

daniel.bryant@tai-dev.co.uk

Choose which email address to associate with this commit

Commit directly to the main branch.



The screenshot shows the Dev House Workshop interface for the 'dev-house' application. The top navigation bar includes tabs for 'Applications', 'dev-house', and 'Canary demo'. Below the navigation is a toolbar with buttons for 'APP DETAILS', 'APP DIFF', 'SYNC', 'SYNC STATUS', 'HISTORY AND ROLLBACK', 'DELETE', and 'REFRESH'. A sidebar on the left displays the version 'v2.0.4+e' and icons for 'Regulars', 'CloudWatch Metrics', 'AWS Lambda', 'Amazon CloudWatch Logs', 'Amazon CloudWatch Metrics Insights', 'Amazon CloudWatch Metrics Insights Metrics', 'Amazon CloudWatch Metrics Insights Metrics Insights', and 'Amazon CloudWatch Metrics Insights Metrics Insights Metrics'. The main content area shows the 'APPLICATION DETAILS' for 'dev-house'. It displays the current sync status as 'Synced' with a green checkmark, and the last sync result as 'Sync OK' with a green checkmark. A note indicates a successful sync a few seconds ago. The deployment history shows multiple 'pod' icons representing different rollout stages: 'summer-k8s-rollout-7b46df69d' (2 minutes, running 1/1), 'summer-k8s-rollout-8874dc9-7-9' (a few seconds, running 1/1), and 'summer-k8s-rollout-8874dc9-7-9' (a few seconds, running 1/1). A note at the bottom indicates 'clear filters to show 11 additional resources'.

The figure displays a 'Request dashboard' consisting of a 5x5 grid of cards. Each card contains a version identifier ('ver: 1.0' or 'ver: 2.0') and is colored either orange or teal. The distribution of versions is as follows:

- Row 1: ver: 2.0 (orange), ver: 1.0 (teal), ver: 1.0 (teal), ver: 2.0 (orange), ver: 1.0 (teal)
- Row 2: ver: 1.0 (teal), ver: 2.0 (orange), ver: 1.0 (teal), ver: 1.0 (teal), ver: 1.0 (teal)
- Row 3: ver: 1.0 (teal), ver: 2.0 (orange), ver: 1.0 (teal), ver: 1.0 (teal), ver: 1.0 (teal)
- Row 4: ver: 1.0 (teal)

Let's change some more things

Change the container image in your fork to v3

And config the promote pauses to be time-based: duration: 30s

Run in your terminal

```
kubectl argo rollouts get rollout summer-k8s-rollout -n demo -w
```

And synchronize

[Edit main/rollout.yaml](https://github.com/danielbryantuk/summer-of-k8s-app-manifests/edit/main/rollout.yaml)

```
1 ---
2   apiVersion: argoproj.io/v1alpha1
3   kind: Rollout
4   metadata:
5     name: summer-k8s-rollout
6   spec:
7     revisionHistoryLimit: 1
8     replicas: 10
9     selector:
10       matchLabels:
11         app: summer-k8s-app
12     template:
13       metadata:
14         labels:
15           app: summer-k8s-app
16       spec:
17         containers:
18           - name: webserver-simple
19             image: kostiscodefresh/summer-of-k8s-app:v3
20             imagePullPolicy: Always
21             ports:
22               - containerPort: 8080
23         strategy:
24           canary:
25             stableService: summer-k8s-service-stable
26             canaryService: summer-k8s-service-canary
27             trafficRouting:
28               ambassador:
29                 mappings:
30                   - summer-k8s-mapping
31             steps:
32               - setWeight: 30
33               - pause: {duration: 30s}
34               - setWeight: 60
35               - pause: {duration: 30s}
```

Commit changes

Update rollout.yaml

Add an optional extended description...

daniel.bryant@tai-dev.co.uk

Choose which email address to associate with this commit

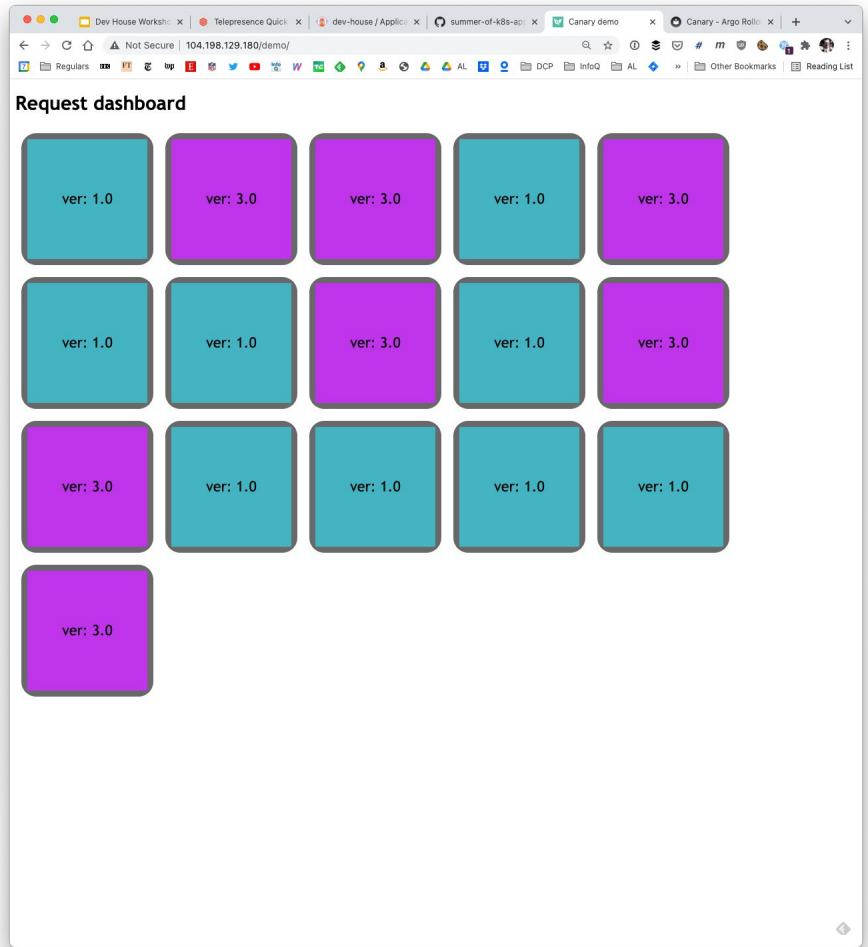
Commit directly to the [main](#) branch.

```

danielbryant - com.docker.cli - docker run -p8083:8083 -p8080:8080 --name ambassador-demo --cap-add=NET_ADMIN --device /dev/net/tun:/dev/net/tun --pull always --rm -it -e AMBASSADOR_API_KEY=ZmJu...
Strategy: Canary
Step: 1/6
SetWeight: 30
ActualWeight: 30
Images: kostiscodefresh/summer-of-k8s-app:v1 (stable)
         kostiscodefresh/summer-of-k8s-app:v3 (canary)
Replicas:
Desired: 10
Current: 13
Updated: 3
Ready: 13
Available: 13

NAME                           KIND    STATUS   AGE     INFO
○ summer-k8s-rollout          Rollout  Paused  6m1s
└─# revision:3
| └─@ summer-k8s-rollout-55b999ff8b             ReplicaSet  Healthy  9s   canary
|   ├─□ summer-k8s-rollout-55b999ff8b-gwwa7   Pod      Running  9s   ready:1/1
|   ├─□ summer-k8s-rollout-55b999ff8b-jxmok   Pod      Running  9s   ready:1/1
|   └─□ summer-k8s-rollout-55b999ff8b-vtldz   Pod      Running  9s   ready:1/1
└─# revision:2
| └─@ summer-k8s-rollout-8874dc9               ReplicaSet  ScaledDown 3m51s
|   ├─□ summer-k8s-rollout-8874dc9-76wbl   Pod      Terminating 3m51s ready:0/1
|   ├─□ summer-k8s-rollout-8874dc9-h6lwg   Pod      Terminating 3m51s ready:0/1
|   └─□ summer-k8s-rollout-8874dc9-r5265   Pod      Terminating 3m51s ready:0/1
└─# revision:1
└─@ summer-k8s-rollout-7b46df69d              ReplicaSet  Healthy  6m1s  stable
  ├─□ summer-k8s-rollout-7b46df69d-796kj   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-b5zw6   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-lcwgs   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-q5zns   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-qcxbw   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-x7qrw   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-c2fnt   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-c8kxx   Pod      Running  6m1s  ready:1/1
  ├─□ summer-k8s-rollout-7b46df69d-fnzb8   Pod      Running  6m1s  ready:1/1
  └─□ summer-k8s-rollout-7b46df69d-wnrjj   Pod      Running  6m1s  ready:1/1

```

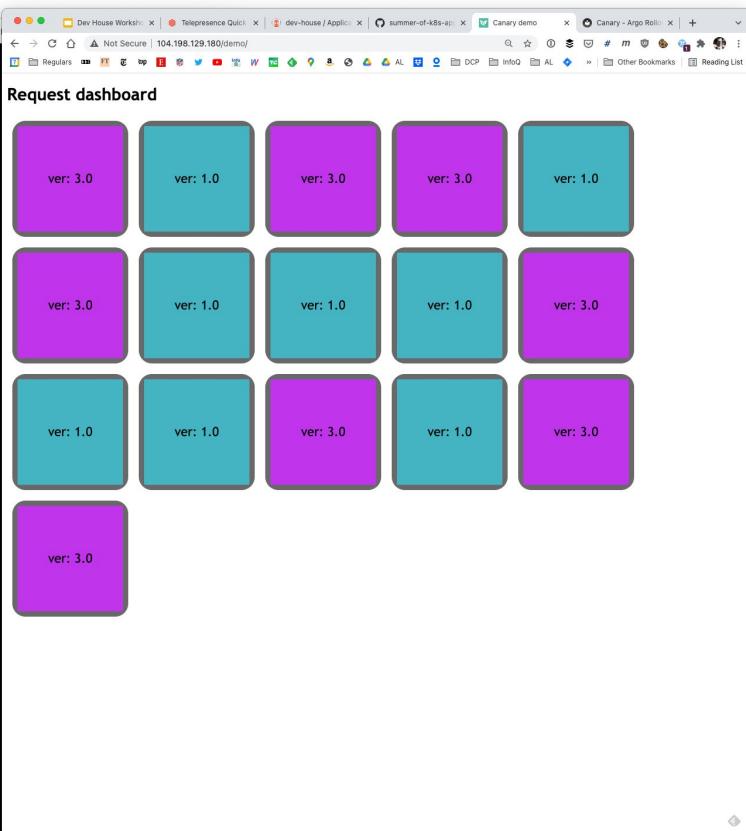


```

Strategy: Canary
Step: 1/6
SetWeight: 30
ActualWeight: 30
Images: kostiscodefresh/summer-of-k8s-app:v1 (stable)
         kostiscodefresh/summer-of-k8s-app:v3 (canary)
Replicas:
Desired: 10
Current: 13
Updated: 3
Ready: 13
Available: 13

NAME                           KIND    STATUS      AGE     INFO
summer-k8s-rollout            Rollout  Paused   6m1s
# revision:3
└─ summer-k8s-rollout-55b999ffb8   ReplicaSet  ✓ Healthy  9s   canary
    └─ summer-k8s-rollout-55b999ffb8-gwwq7 Pod    ✓ Running  9s   ready:1/1
    └─ summer-k8s-rollout-55b999ffb8-jxmqk Pod    ✓ Running  9s   ready:1/1
    └─ summer-k8s-rollout-55b999ffb8-vtldz Pod    ✓ Running  9s   ready:1/1
# revision:2
└─ summer-k8s-rollout-8874dc9   ReplicaSet  ● ScaledDown 3m51s
    └─ summer-k8s-rollout-8874dc9-76wbl Pod    ○ Terminating 3m51s ready:0/1
    └─ summer-k8s-rollout-8874dc9-h6lwg Pod    ○ Terminating 3m51s ready:0/1
    └─ summer-k8s-rollout-8874dc9-r5265 Pod    ○ Terminating 3m51s ready:0/1
# revision:1
└─ summer-k8s-rollout-7b46df69d  ReplicaSet  ✓ Healthy  6m1s  stable
    └─ summer-k8s-rollout-7b46df69d-796kj Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-b5zw6 Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-lcwgs Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-q5zns Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-qcxbw Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-x7qrw Pod    ✓ Running  6m1s  ready:1/1
    └─ summer-k8s-rollout-7b46df69d-c2fnt Pod    ✓ Running  6m   ready:1/1
    └─ summer-k8s-rollout-7b46df69d-c8kkx Pod    ✓ Running  6m   ready:1/1
    └─ summer-k8s-rollout-7b46df69d-fnzb8 Pod    ✓ Running  6m   ready:1/1
    └─ summer-k8s-rollout-7b46df69d-wnrjj Pod    ✓ Running  6m   ready:1/1

```



What's next?

The screenshot shows a browser window with multiple tabs open, including "Dev House Workshop", "Telepresence Quick", "dev-house / Applications", "summer-of-k8s-apps", "Canary demo", "DataDog - Argo Rollouts", and the current tab, "Argo Rollouts - Kubernetes Progressive Delivery Controller". The main content is the "Datadog Metrics" section of the Argo Rollouts documentation.

Argo Rollouts - Kubernetes Progressive Delivery Controller

Datadog Metrics

Important
Available since v0.10.0

A Datadog query can be used to obtain measurements for analysis.

```
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: log-error-rate
spec:
  args:
    - name: service-name
  metrics:
    - type: error-rate
      interval: 5m
      successCondition: result <= 0.01
      failureLimit: 3
      provider:
        datadog:
          interval: 5m
          query: |
            sum:requests.error.count{service:{{args.service-name}}} /
            sum:requests.request.count{service:{{args.service-name}}}
```

Datadog api and app tokens can be configured in a kubernetes secret in argo-rollouts namespace.

```
apiVersion: v1
kind: Secret
metadata:
  name: datadog
type: Opaque
data:
  address: https://api.datadoghq.com
  api-key: <datadog-api-key>
  app-key: <datadog-app-key>
```

Previous **Prometheus** **Next** **NewRelic**

Made with Material for MkDocs

argoproj.github.io/argo-rollouts/features/analysis/

Wrapping up! 📦

Thanks for joining us: It's been fun!

Today, you have:

- Learned about continuous delivery
- Explored how continuous delivery has evolved on Kubernetes
- Learned progressive delivery
- Introduced GitOps
- Got hands on with Argo Rollouts and Argo CD

Continue learning at the [Kubernetes Developer Learning Center!](#)

Take our Ambassador Cloud tour: [a8r.io/workshop](#)

Join Ankur Soman and team for office hours Wed, 2nd Feb! [a8r.io/cd-office-hours](#)

Bonus slides

Other K8s concepts for CD / PD

- [PodDisruptionBudgets](#)
- [Liveness Probes,](#)
- [Readiness Probes,](#)
- [startupProbe,](#)
- [Readiness gates](#)

Good examples:

loft.sh/blog/kubernetes-readiness-probes-examples-common-pitfalls/