



SYDNEY



5 & 6 SEPTEMBER

Mastering Kubernetes Deployments and workflows

With the Argo Project Suite





Kostis Kapelonis

- Developer Advocate (Octopus Deploy/Codefresh)
- Argo Maintainer (Argo CD, Argo Rollouts)
- Co-author GitOps Certification
learning.codefresh.io



Agenda

- Project Introduction
- Argo Workflows
- Argo CD
- Argo Rollouts
- Argo Events
- Use Cases



SYDNEY



5 & 6 SEPTEMBER

Introduction

Get More Done with **Kubernetes**

Open source tools for Kubernetes to run workflows, manage clusters, and do GitOps right.

[View on GitHub](#)

Trusted by



Google

Red Hat

WORDPRESS

ticketmaster

<https://argoproj.github.io/>



Argo CD

🐱 ★ 16997

Declarative continuous delivery with a fully-loaded UI.

[Learn More](#)



Argo Rollouts

🐱 ★ 2619

Advanced Kubernetes deployment strategies such as Canary and Blue-Green made easy.

[Learn More](#)



Argo Workflows

🐱 ★ 14685

Kubernetes-native workflow engine supporting DAG and step-based workflows.

[Learn More](#)

Argo Events

🐱 ★ 2299

Event based dependency management for Kubernetes.

[Learn More](#)



What the Argo Projects do

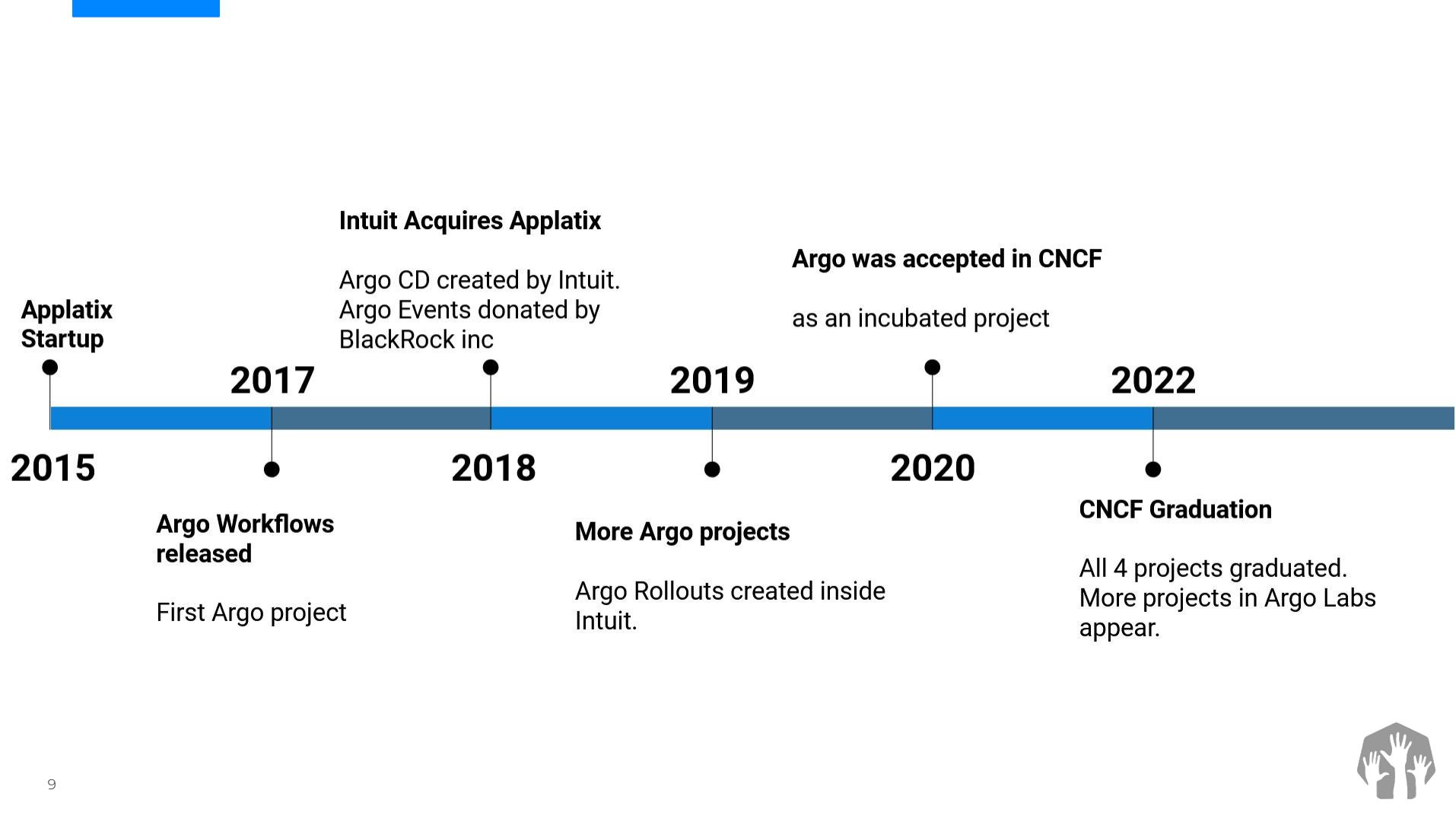
- Argo CD → Deploy your App with GitOps
- Argo Workflows → Execute a job/process
- Argo Events → Monitor/Create Events
- Argo Rollouts → Avoid deployment downtime



All 4 projects are self-contained

- There are NO dependencies between the 4 projects
- You can use each project on its own
- There are several common integrations
- Some shared code parts (e.g. notifications, SSO)
- You get extra value by combining them
- It is possible to use all 4 of them (explained later in Use cases)







argoproj-labs

README.md

argoproj-labs

This org is managed by the Argo project maintainers and not part of the CNCF Argo umbrella projects. New repos in this org need to be sponsored and created by one of the Argo project maintainers. The goal is to have a place to collaborate with the community to quickly run experiments, POCs and possibly new features to be later incorporated in one of the Argo projects.

Pinned

[argocd-image-updater](#) Public

Automatic container image update for Argo CD

Go 1.2k 249

[argocd-operator](#) Public

A Kubernetes operator for managing Argo CD clusters.

Go 612 660

[community](#) Public

Community documents for argoproj-labs

12 6

[argocd-autopilot](#) Public

Argo-CD Autopilot

Go 873 119

<https://github.com/argoproj-labs>



Created by

INTUIT

Maintained with ❤️ by:

Akuity



BlackRock



Codefresh



Intuit



Pipekit



Red Hat



[Contact us](#) to learn more about corporate maintainers.

Codefresh was acquired by Octopus Deploy in 2024



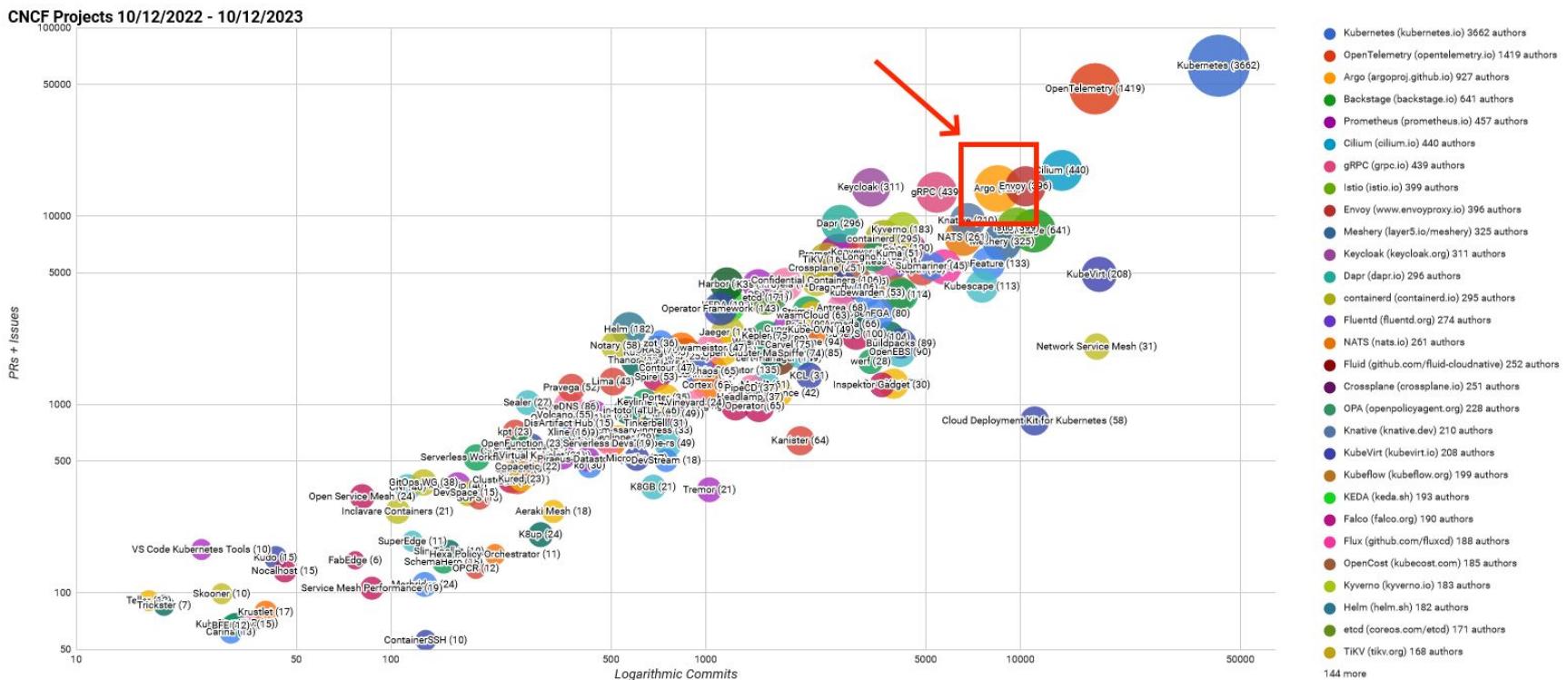
SYDNEY



5 & 6 SEPTEMBER

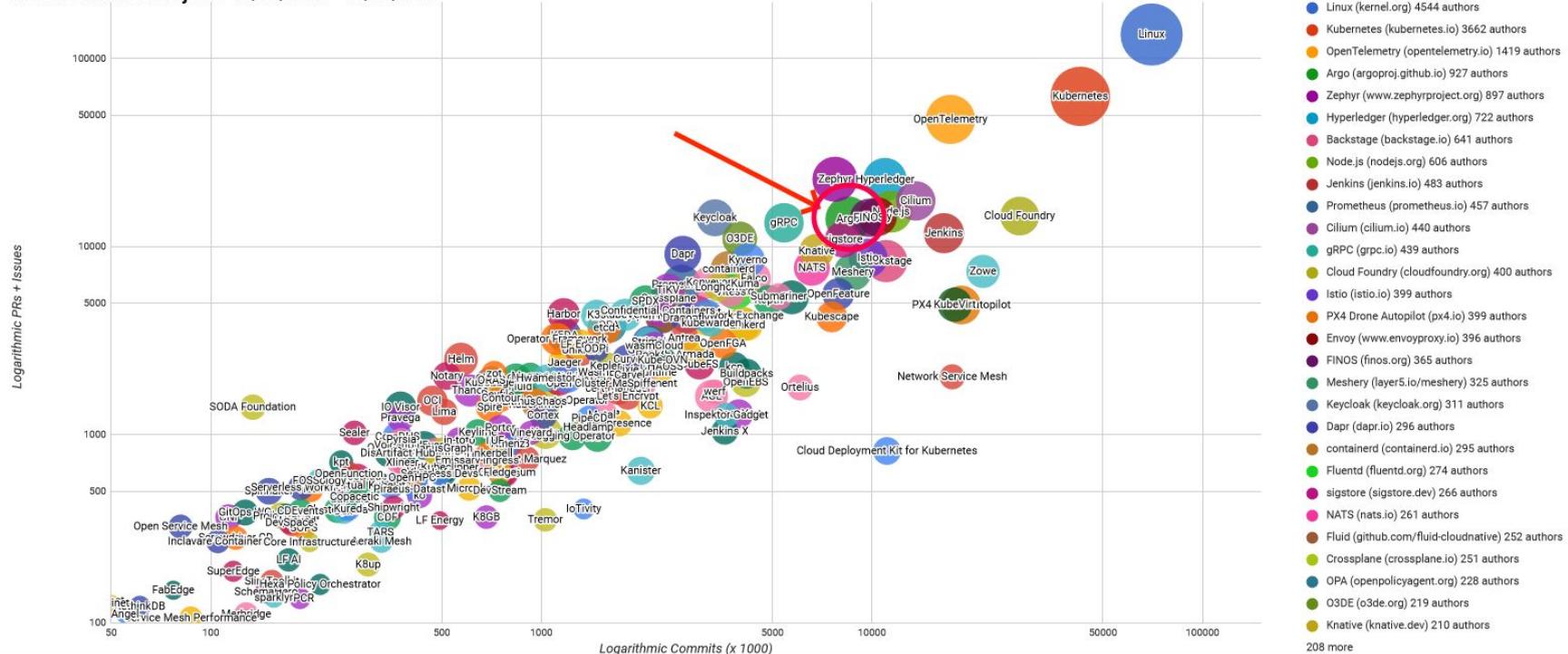
Popularity

Popular/Active CNCF projects



Popular/Active Linux Foundation projects

Linux Foundation Projects 10/12/2022 - 10/12/2023



SYDNEY



5 & 6 SEPTEMBER

Argo Workflows

Argo Workflows

- The original Argo Project
- Workflows/processes
- Kubernetes native
- Alternative to Tekton, Apache Airflow
- Can be used for CI/CD, ML, ETL, Batch jobs etc



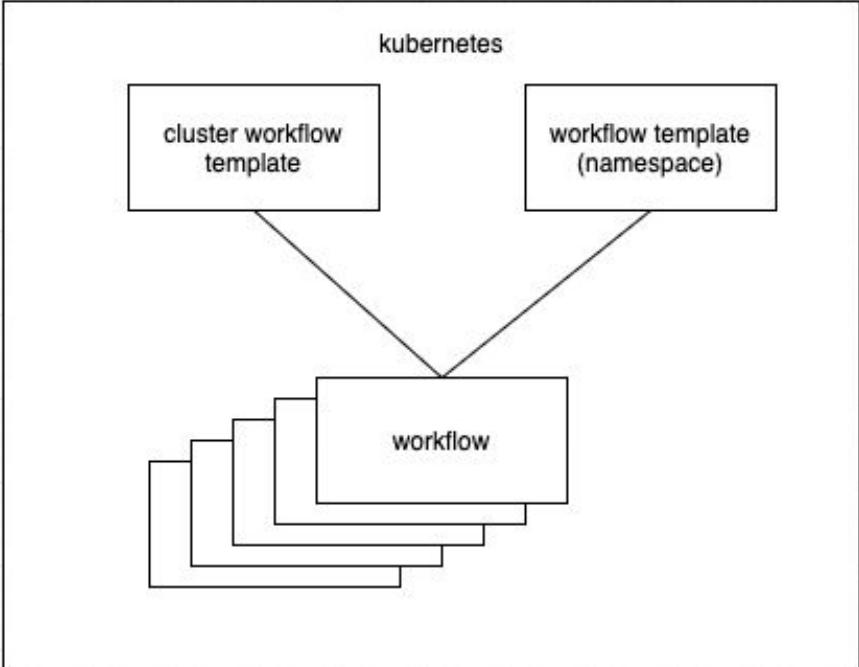


Image credit: pipekit.io



Argo Workflows entities

- **Workflow** - running instance
- **Workflow template** - definition of Workflow
- **CronWorkflows** - on a schedule
- **Cluster Workflow template** - not constrained on a single namespace



```
apiVersion: argoproj.io/v1alpha1
kind: Workflow          # new type of k8s spec
metadata:
  generateName: hello-world-    # name of the workflow spec
spec:
  entrypoint: hello-world      # invoke the hello-world template
  templates:
    - name: hello-world        # name of the template
      container:
        image: busybox
        command: [ echo ]
        args: [ "hello world" ]
        resources: # limit the resources
          limits:
            memory: 32Mi
            cpu: 100m
```



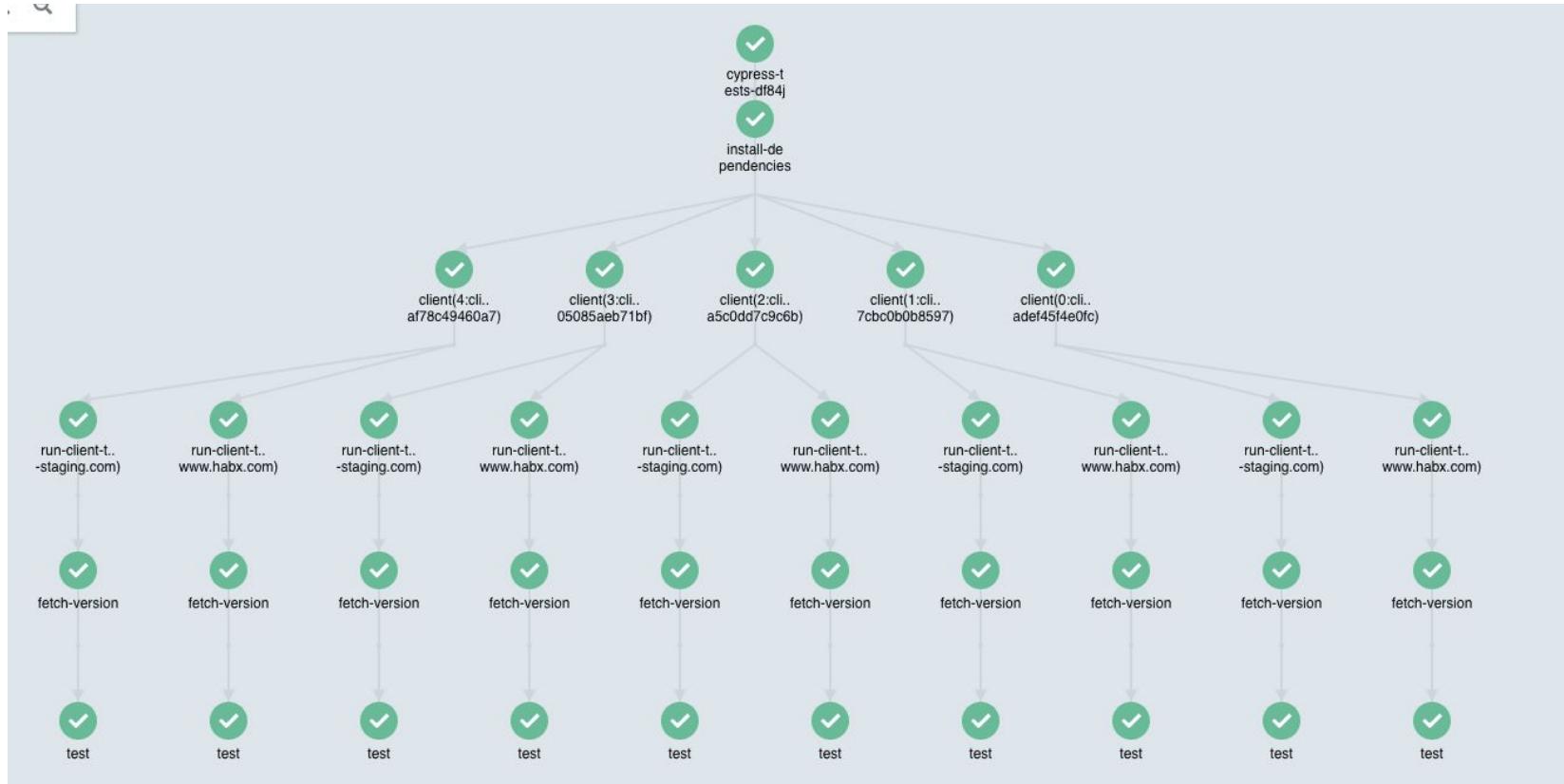
Step-per-pod

- Each step runs on a separate container/pod
- Gain all the advantages of Kubernetes auto-scaling, observability and CRD management

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: scripts-bash-
spec:
  entrypoint: bash-script-example
  templates:
  - name: bash-script-example
    steps:
    - - name: generate
        template: gen-random-int-bash
    - - name: print
        template: print-message
        arguments:
          parameters:
          - name: message
            value: "{{steps.generate.outputs.result}}" # The result of the here-scri
    - name: gen-random-int-bash
      script:
        image: debian:9.4
        command: [bash]
        source: |
          cat /dev/urandom | od -N2 -An -i | awk -v f=1 -v r=100 '{printf "%i\n", f -
# Contents of the here-scri
    - name: gen-random-int-python
      script:
        image: python:alpine3.6
        command: [python]
        source: |
          import random
          i = random.randint(1, 100)
          print(i)
```



CI/CD example



Workflow Details

SUMMARY

NAME	coinflip-1723111800[0] flip-con
ID	coiflip-1723111800-23223386521
POD NAME	coinflip-1723111800-flip-con-23223386521
HOST NODE NAME	gke-ago-demo-apps-default-node-pool-525d6970-bgm
TYPE	Pod
PHASE	Succeeded
START TIME	08/08/2024, 13:10:00 (53m9s ago)
END TIME	08/08/2024, 13:17:24 (45m42s ago)
DURATION	7m24s

CONTAINERS

INPUTS/OUTPUTS

ARTIFACTS

LOGS

SHARE

PREVIOUS RUNS

OPEN WORKFLOW TEMPLATE

SEARCH

```

graph TD
    coinflip[coinflip-1723111800] --> tails[tails]
    coinflip --> heads[heads]
  
```

Workflow Templates

CREATE NEW WORKFLOW TEMPLATE

NAME

artifacts

NAMESPACE

workflow-playground

LABELS

This example shows how to produce different types of artifact.

NAME PATTERN

coiflip

DESCRIPTION

Build and push an image using Docker Buildkit. This does not need privileged access.

distro

This workflow template builds and tests Argo Workflows. It demonstrates: * Cache restore a

github-event

REPORTS

Duration

Average: 189

Duration (seconds)

Legend: Workflow Template, Cron Workflow

Phase: Succeeded, Error, Failed

CRON WORKFLOW DETAILS

STATUS

MANIFEST

```

1 metadata:
2   name: coinflip
3   namespace: workflow-playground
4   uid: 9a72e2ef-5aef-403b-8dce-e4bad46705a1
5   resourceVersion: '21531942'
6   generation: 92367
7   creationTimestamp: '2024-03-01T09:28:14Z'
8   annotations:
9     argoproj.io/tracking-id: workflow-examples:argoproj.io/cronworkflow:workflow-playground/coinflip
10    cronworkflow.argoproj.io/last-used-scheduler: '*/* * * * *'
11    kubectl.kubernetes.io/last-applied-configuration: >
12      {"apiVersion": "argoproj.io/v1alpha1", "kind": "CronWorkflow", "metadata": {"annotations": {"argocd.argoproj.io/tracking-id": "workflow-examples:argoproj.io/cronworkflow:workflow-playground/coinflip"}, "labels": {"argocd.argoproj.io/label": "coinflip"}, "name": "coinflip"}, "spec": {"workloadSpec": {"serviceAccountName": "workflow", "workflowTemplateRef": {"name": "coinflip"}}, "schedule": "0 * * * *"}, "status": {}}
13    controller-revision-count: 1
14    managed-fields:
15      - manager: argocd-controller
16        operation: Update
17        apiVersion: argoproj.io/v1alpha1
18        time: '2024-03-01T09:28:43Z'
19        fields: fieldv1
20        fieldv1:
21          fieldmetadata:
22            fAnnotations:
23              : {}
24            fArgocd.argoproj.io/tracking-id: {}
  
```

WORKFLOW DETAILS

INPUTS/OUTPUTS

ARTIFACTS

LOGS

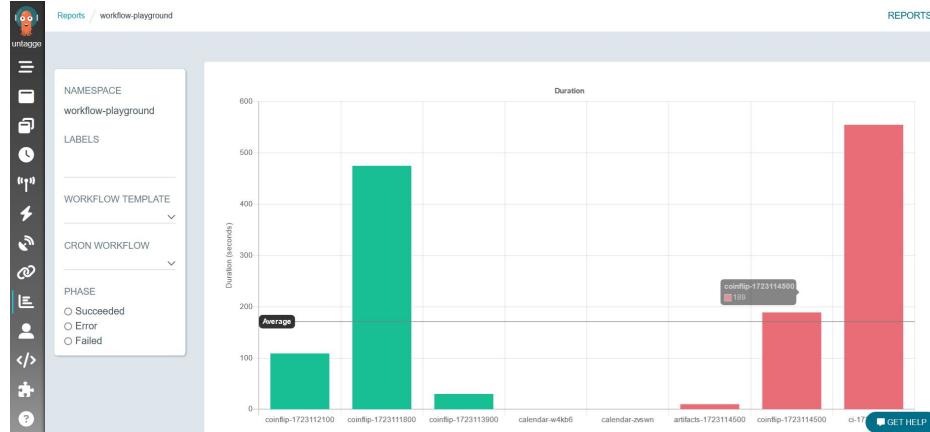
SHARE

PREVIOUS RUNS

OPEN WORKFLOW TEMPLATE

SEARCH

GET HELP



Cron Workflows

COINFLIP

STATUS

MANIFEST

```

1 metadata:
2   name: coinflip
3   namespace: workflow-playground
4   uid: 9a72e2ef-5aef-403b-8dce-e4bad46705a1
5   resourceVersion: '21531942'
6   generation: 92367
7   creationTimestamp: '2024-03-01T09:28:14Z'
8   annotations:
9     argoproj.io/tracking-id: workflow-examples:argoproj.io/cronworkflow:workflow-playground/coinflip
10    cronworkflow.argoproj.io/last-used-scheduler: '*/* * * * *'
11    kubectl.kubernetes.io/last-applied-configuration: >
12      {"apiVersion": "argoproj.io/v1alpha1", "kind": "CronWorkflow", "metadata": {"annotations": {"argocd.argoproj.io/tracking-id": "workflow-examples:argoproj.io/cronworkflow:workflow-playground/coinflip"}, "labels": {"argocd.argoproj.io/label": "coinflip"}, "name": "coinflip"}, "spec": {"workloadSpec": {"serviceAccountName": "workflow", "workflowTemplateRef": {"name": "coinflip"}}, "schedule": "0 * * * *"}, "status": {}}
13    controller-revision-count: 1
14    managed-fields:
15      - manager: argocd-controller
16        operation: Update
17        apiVersion: argoproj.io/v1alpha1
18        time: '2024-03-01T09:28:43Z'
19        fields: fieldv1
20        fieldv1:
21          fieldmetadata:
22            fAnnotations:
23              : {}
24            fArgocd.argoproj.io/tracking-id: {}
  
```

CRON

METADATA

WORKFLOW

WORKFLOW METADATA

JSON/YAML

METADATA

SPEC

STATUS

GET HELP



Argo Workflows - other features

- Artifact storage/retrieval
- Workflow Archiving
- CLI/API and Analytics
- Retry mechanism/ Timeouts
- Suspend/resume
- Loops/Conditionals
- SSO/RBAC



SYDNEY

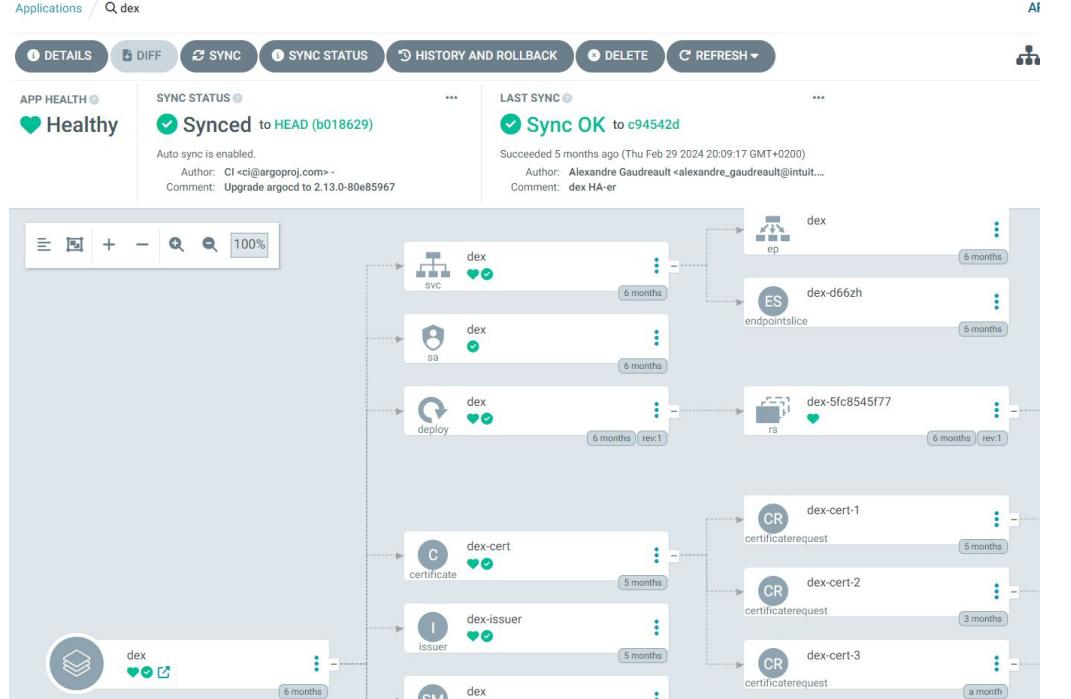


5 & 6 SEPTEMBER

Argo CD

Argo CD

- Deploys applications
- Kubernetes native
- Supports Helm/Kustomize
- Health status analysis
- Multi-tenant/RBAC



Argo CD UI

Applications

+ NEW APP SYNC APPS REFRESH APPS Search applications... / Log in

Previous 1 2 Next Sort: name ▾ Items per page: 10 ▾

argo-events Project: default Labels: Status: Healthy Synced Repository: https://github.com/argoproj/argoproj-de... Target R...: HEAD Path: argo-events Destinati...: in-cluster Namesp...: workflow-playground Created ...: 02/22/2024 21:32:40 (6 months ago) Last Sync: 07/13/2024 18:06:35 (a month ago) SYNC REFRESH DELETE	argo-rollouts Project: default Labels: Status: Healthy Synced Repository: https://github.com/argoproj/argoproj-de... Target R...: HEAD Path: argo-rollouts Destinati...: in-cluster Namesp...: argo-rollouts Created ...: 02/22/2024 21:32:41 (6 months ago) Last Sync: 02/23/2024 19:36:19 (6 months ago) SYNC REFRESH DELETE	argo-workflows Project: default Labels: Status: Healthy Synced Repository: https://github.com/argoproj/argoproj-de... Target R...: HEAD Path: argo-workflows Destinati...: in-cluster Namesp...: argo Created ...: 02/22/2024 21:32:41 (6 months ago) Last Sync: 03/11/2024 13:40:15 (5 months ago) SYNC REFRESH DELETE
argocd-image-updater Project: default Labels: Status: Healthy Synced Repository: https://github.com/argoproj/argoproj-de... Target R...: HEAD Path: argocd-image-updater Destinati...: in-cluster	dex Project: default Labels: Status: Healthy Synced Repository: https://github.com/argoproj/argoproj-de... Target R...: HEAD Path: dex Destinati...: in-cluster	example.guestbook Project: default Labels: Status: Healthy Synced Repository: https://github.com/agaudreault/argocd-... Target R...: sync-from-demo Path: guestbook Destinati...: in-cluster

Argo CD UI

APPLICATION DETAILS TREE

Log in

Applications / Q example.guestbook

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH Healthy

Synced to sync-from-demo (0d6bc4b)

Auto sync is enabled.

Author: Alexandre Gaudreault <alexandre_gaudreault@intuit...

Comment: update readme

LAST SYNC Sync OK to d7927a2

Succeeded 5 months ago (Mon Mar 11 2024 20:23:46 GMT+0200)

Author: Anand Francis Joseph <anandfrancis.joseph@gmail.com...

Comment: Template variable for container port (#251)

NAME

KINDS

SYNC STATUS

Synced 2
OutOfSync 0

HEALTH STATUS

Healthy 4
Progressing 0
Degraded 0

example.guestbook

guestbook-ui

guestbook-ui-kfh6c

guestbook-ui-56c646849b

guestbook-ui-56c1

svc

deploy

ep

endpointslice

rs

pod

5 months

5 months

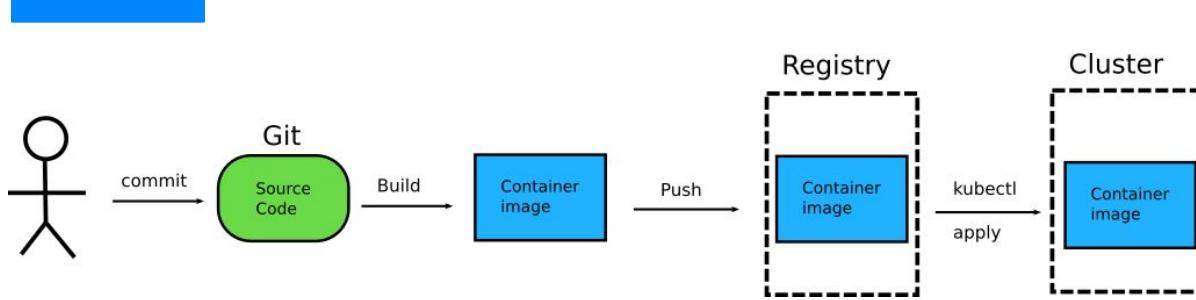
5 months

5 months

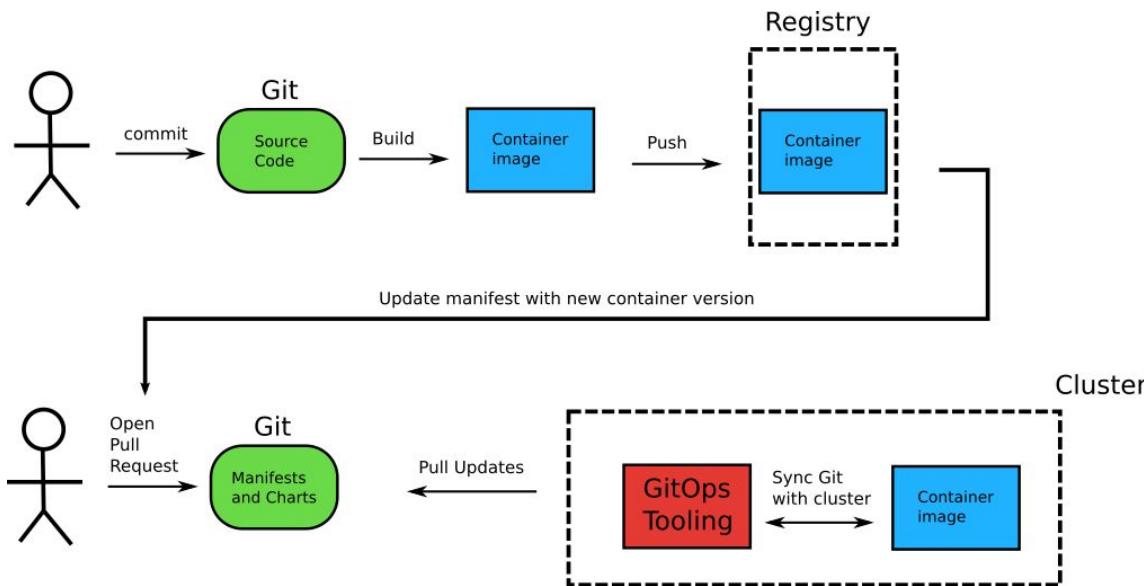
rev:1

13

27



Abusing CI as CD



With Argo CD



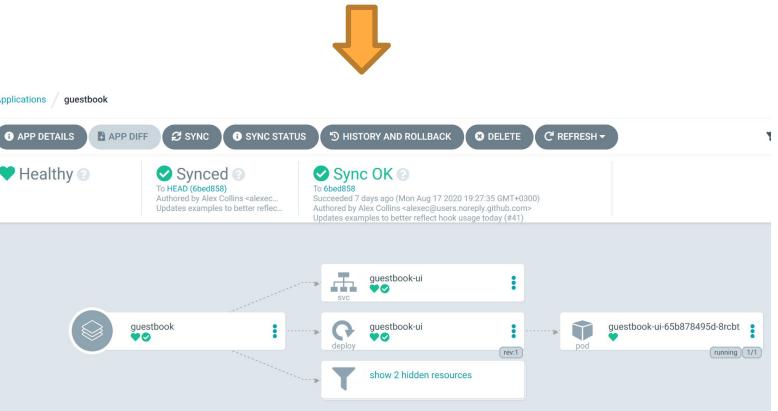
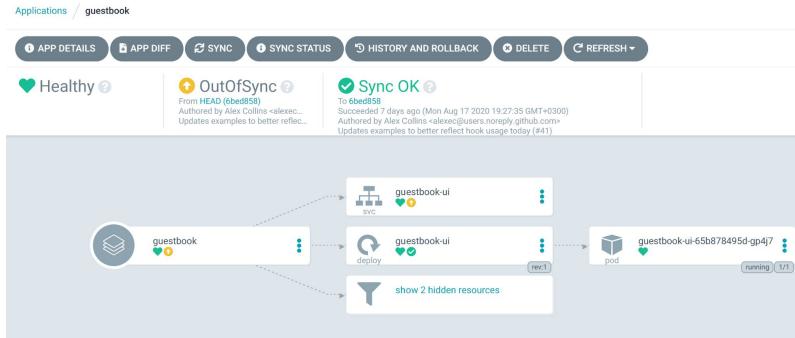
Avoid Configuration Drift

SUMMARY PARAMETERS MANIFEST DIFF EVENTS

Compact diff Inline Diff

/Service/default/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: 8080
10    targetPort: 80
11   selector:
12     app: guestbook-ui
13
14
```



Argo CD entities

- **Application**- Link between a cluster and Git repo
- **Project** - RBAC for Applications
- **ApplicationSet**- Generator/grouping for applications

The screenshot shows the Argo CD web interface with a sidebar on the left and a main content area with several application cards.

Left Sidebar:

- Argo CD v2.11.0+10ffaf5
- Applications
- Settings
- User Info
- Documentation
- SYNC STATUS: 20 Synced, 0 OutOfSync
- HEALTH STATUS: 20 Healthy, 0 Unknown, 6 Progressing, 2 Suspended, 2 Degraded, 0 Missing

Main Content Area (Applications List):

Entity	Project	Status	Last Sync
argo-events	default	Healthy	02/22/2024 21:32:40 (6 months ago)
argo-rollouts	default	Healthy	02/22/2024 21:32:41 (6 months ago)
argo-workflows	default	Healthy	02/22/2024 21:32:41 (6 months ago)
argod-image-updater	default	Healthy	02/13/2024 18:06:35 (a month ago)
dex	default	Healthy	02/23/2024 19:36:19 (6 months ago)
example.guestbook	default	Healthy	03/11/2024 13:40:15 (5 months ago)

Top Bar: Applications TILES, NEW APP, SYNC APPS, REFRESH APPS, Search applications...

Sync manifests to Cluster

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: guestbook Name of application
  namespace: argocd
spec:
  project: default
  source: Where to read the Kubernetes manifest
    repoURL: https://github.com/argoproj/argocd-example-apps.git
    targetRevision: HEAD
    path: guestbook
  destination: Which cluster to deploy the application to
    server: https://kubernetes.default.svc
    namespace: guestbook
```



```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: my-qa-appset
  namespace: argocd
spec:
  goTemplate: true
  goTemplateOptions: ["missingkey=error"]
  generators:
    - git:
        repoURL: https://github.com/kostis-codefresh/many-appsets-demo.git
        revision: HEAD
        directories:
          - path: apps/*/envs/qa
  template:
  metadata:
    name: '{{index .path.segments 1}}-{{index .path.segments 3}}'
  spec:
    # The project the application belongs to.
    project: default

    # Source of the application manifests
    source:
      repoURL: https://github.com/kostis-codefresh/many-appsets-demo.git
      targetRevision: HEAD
      path: '{{.path.path}}'

    # Destination cluster and namespace to deploy the application
    destination:
      server: https://kubernetes.default.svc
      namespace: '{{index .path.segments 1}}-{{index .path.segments 3}}'
```

The screenshot shows the Codefresh interface. On the left, there's a sidebar with a user icon and the text "remove duplicate namespaces". Below it is a "Name" field and a list of files: .., deployment.yml, kustomization.yml, replicas.yml, settings.yml, and version.yml. In the center, there's a "Files" section with a search bar and a tree view of the repository structure:

- apps
- billing
- base
- envs
 - prod-eu
 - prod-us
- fake-invoices
- invoices
- orders
- payments



Generate applications from Git folders



Cluster bootstrapping



Argo CD topologies



Argo CD - other features

- Sync policies
- Sync waves/phases/windows
- Git webhooks
- CLI/API
- SSO/RBAC
- Plugins
- Notifications

The screenshot shows the Argo CD application dashboard. At the top left is the Argo logo and version v2.11.0+1efaf15. Below it is a sidebar with links: Applications, Settings, User Info, Documentation, and a Favorites Only section. The sidebar also shows SYNC STATUS: 20 Synced, 0 OutOfSync, 0 Unknown, 6 Progressing, 0 Suspended, 2 Degraded, and 0 Missing. At the bottom of the sidebar are buttons for SYNC, REFRESH, and DELETE.

The main area is titled "Applications" and shows a list of sync configurations:

- argo-events**: Project default, Status Healthy, Synced, Repository https://github.com/argoproj/argoproj-de..., Target R... HEAD, Path argo-events, Destination in-cluster, Namespace workflow-playground, Created 02/22/2024 21:32:40 (6 months ago), Last Sync 07/13/2024 18:06:35 (a month ago). Buttons: SYNC, REFRESH, DELETE.
- argo-rollouts**: Project default, Status Healthy, Synced, Repository https://github.com/argoproj/argoproj-de..., Target R... HEAD, Path argo-rollouts, Destination in-cluster, Namespace argo-rollouts, Created 02/22/2024 21:32:41 (6 months ago), Last Sync 02/23/2024 19:36:19 (6 months ago). Buttons: SYNC, REFRESH, DELETE.
- argo-workflows**: Project default, Status Healthy, Synced, Repository https://github.com/argoproj/argoproj-de..., Target R... HEAD, Path argo-workflows, Destination in-cluster, Namespace argo, Created 02/22/2024 21:32:41 (6 months ago), Last Sync 03/11/2024 13:40:15 (5 months ago). Buttons: SYNC, REFRESH, DELETE.
- argod-image-updater**: Project default, Status Healthy, Synced, Repository https://github.com/argoproj/argoproj-de..., Target R... HEAD, Path argod-image-updater, Destination in-cluster. Buttons: SYNC, REFRESH, DELETE.
- dex**: Project default, Status Healthy, Synced, Repository https://github.com/argoproj/argoproj-de..., Target R... HEAD, Path dex, Destination in-cluster. Buttons: SYNC, REFRESH, DELETE.
- example.guestbook**: Project default, Status Healthy, Synced, Repository https://github.com/agauadreault/argod..., Target R... sync-from-demo, Path guestbook, Destination in-cluster. Buttons: SYNC, REFRESH, DELETE.

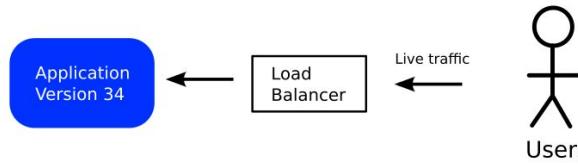
SYDNEY



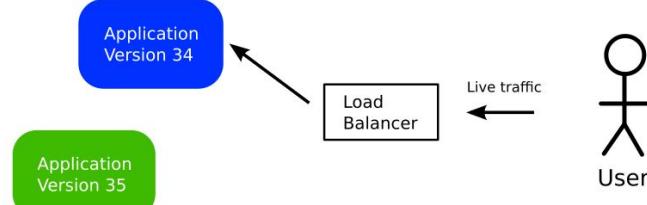
5 & 6 SEPTEMBER

Argo Rollouts

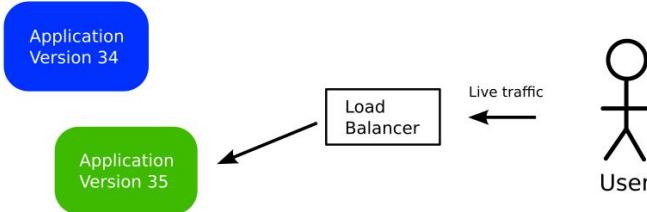
1- Initial version



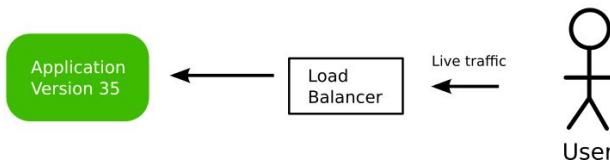
2- New version deployed



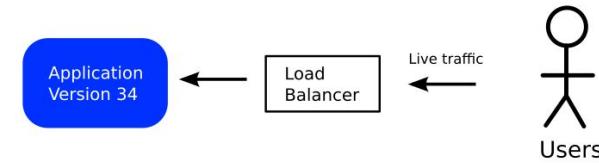
3- Switch Traffic



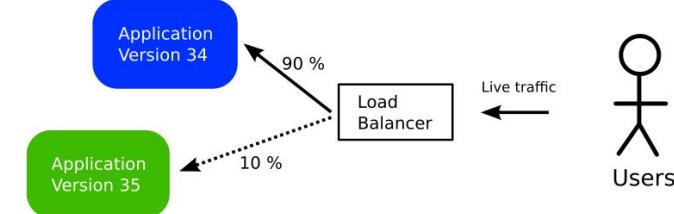
4- Finish



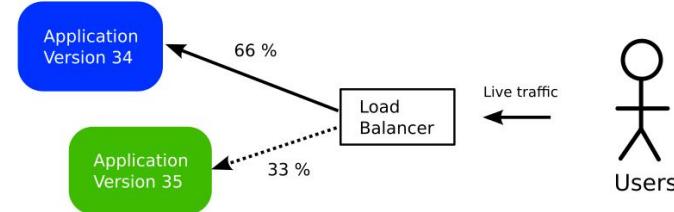
1- Initial version



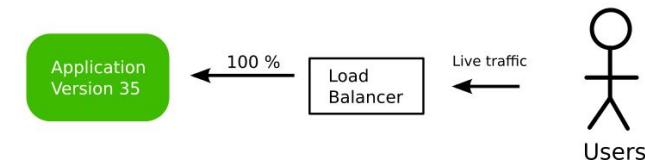
2- New version used by 10% of users



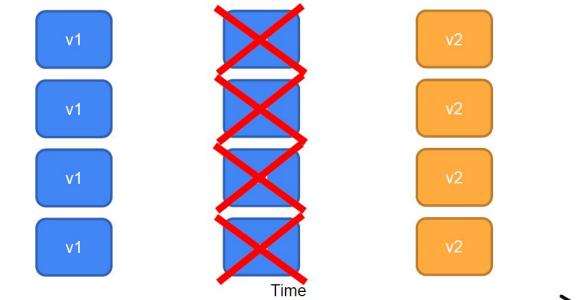
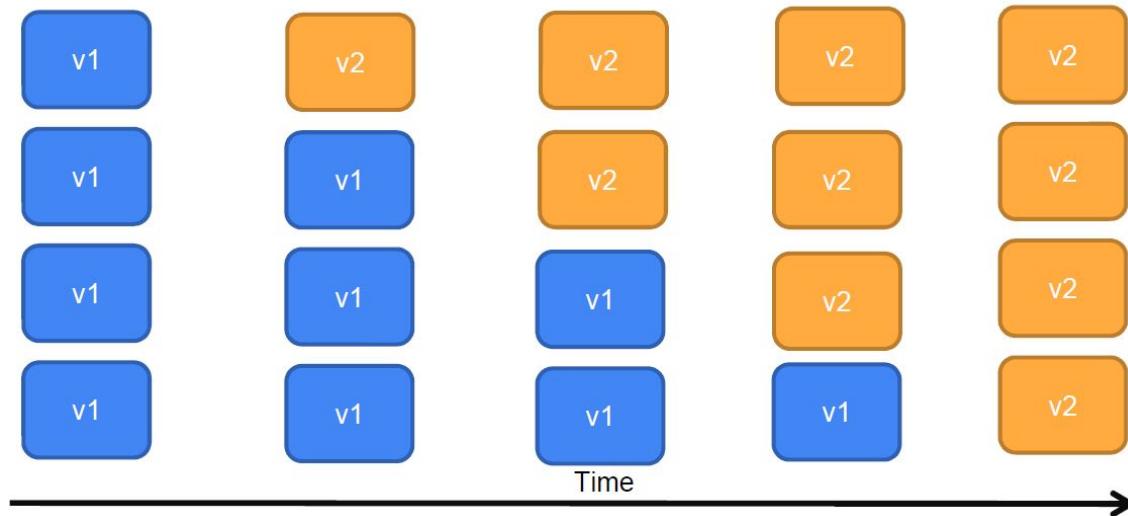
3- New version used by 33% of users



4- New version is used by all users



Default Kubernetes deployments



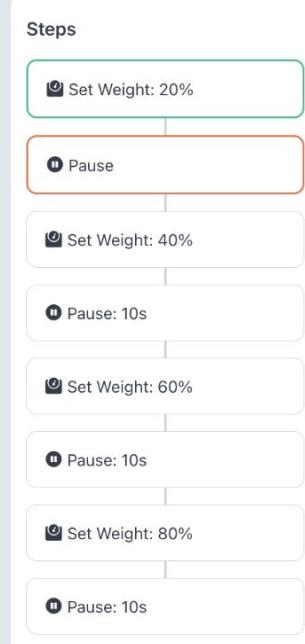
Argo Rollouts

- Rollouts (new CRD)
- Extends Deployment
- Blue/Green/Canaries
- Minimal dashboard
- Pre/Post checks

The screenshot shows the Argo Rollouts dashboard for a deployment named "rollouts-demo". The top navigation bar includes buttons for "Restart", "Retry", "Abort", "Promote", and "PromoteFull". The main interface is divided into several sections:

- Steps:** A list of deployment steps:
 - Set Weight: 20% (status: green)
 - Pause (status: orange)
 - Set Weight: 40% (status: green)
 - Pause: 10s (status: green)
 - Set Weight: 60% (status: green)
 - Pause: 10s (status: green)
 - Set Weight: 80% (status: green)
 - Pause: 10s (status: green)
- Summary:** Displays the strategy (Canary), step (1/8), set weight (20), and actual weight (20).
- Containers:** Shows the container "rollouts-demo" with the image "argoproj/rollouts-demo:yellow".
- Revisions:** Lists two revisions:
 - Revision 2:** "argoproj/rollouts-demo:yellow" (canary status)
 - Revision 1:** "argoproj/rollouts-demo:blue" (stable status)
- Logs:** A section showing log entries for the deployment.





Summary

Strategy	Canary
Step	1/8
Set Weight	20
Actual Weight	20

Containers

[Edit](#)

rollouts-demo	argoproj/rollouts-demo:yellow
---------------	-------------------------------

Revisions

Revision	Image	Status
Revision 2	canary	
Revision 1	stable	



Argo Rollouts Entities

- **Rollout** - main spec
- **AnalysisTemplate** - define pre/post checks
- **ClusterAnalysisTemplate** - clusterwide
- **AnalysisRun** - result of check
- **Experiment** - a/b testing

The screenshot shows the Argo Rollouts web interface for a rollout named "rollouts-demo".

Summary: Strategy is set to "Canary" (1/8). Step 1 has an "Actual Weight" of 20%. Step 2 has an "Actual Weight" of 20%. Step 3 has an "Actual Weight" of 20%. Step 4 has an "Actual Weight" of 20%.

Containers: The container "rollouts-demo" is running the revision "argoproj/rollouts-demo:yellow".

Revisions: Revision 2 (argoproj/rollouts-demo:yellow) is marked as "canary". Revision 1 (argoproj/rollouts-demo:blue) is marked as "stable".

Buttons: Top right buttons include "Restart", "Abort", "Promote", and "PromoteFull". Bottom right buttons include "Rollback" and "Stable".



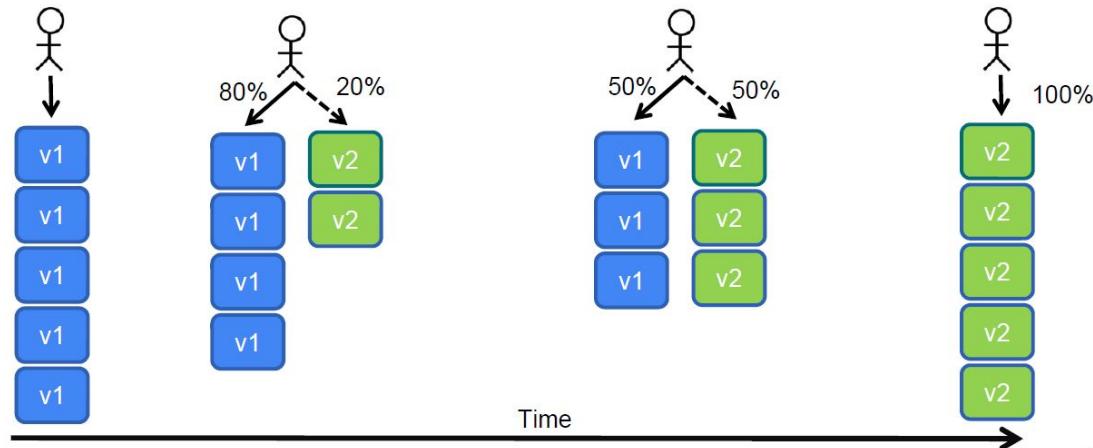
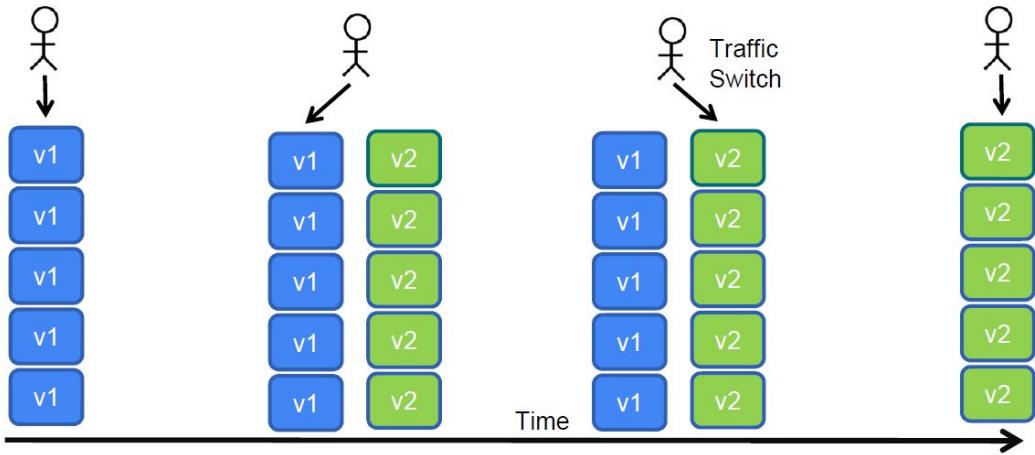
```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: example-rollout
spec:
  replicas: 10
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.15.4
          ports:
            - containerPort: 80
  minReadySeconds: 30
  revisionHistoryLimit: 3
strategy:
  canary: #Indicates that the rollout should use the Canary strategy
  maxSurge: "25%"
  maxUnavailable: 0
  steps:
    - setWeight: 10
    - pause:
        duration: 1h # 1 hour
    - setWeight: 20
    - pause: {} # pause indefinitely
```

Strategy

Rollout extends K8s deployment



Kubernetes Progressive Delivery



Without/With traffic management

Prod
50%



Canary
50%



Prod
75%



Canary
25%



Prod
90%



Canary
10%



Linkerd



Prod
95%



Canary
5%



Linkerd



Prod
70%



Canary
30%



Linkerd



Prod
20%



Canary
80%



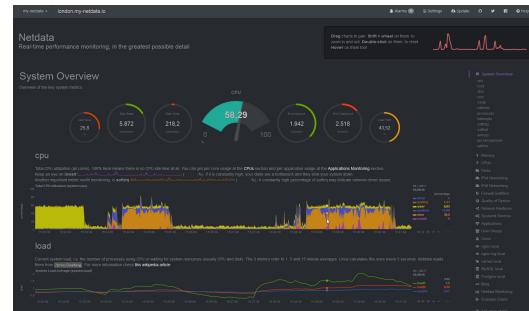
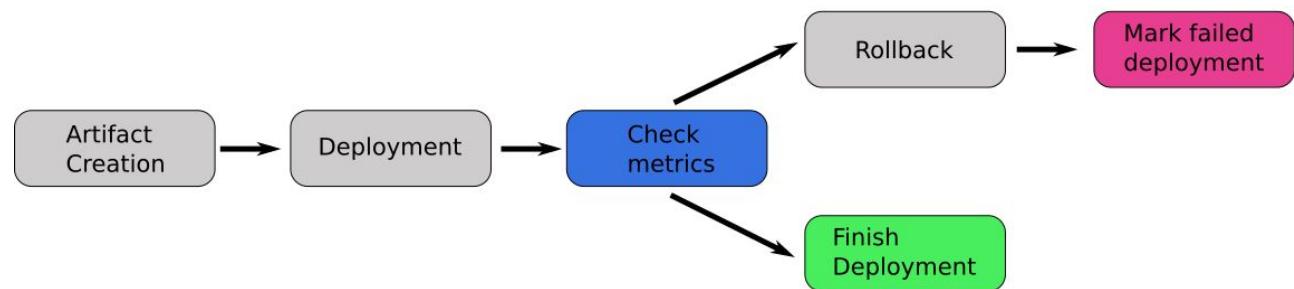
Supported Traffic managers

- AWS Ingress Controller
- Ambassador Labs
- Apache APISIX
- Linkerd
- Istio
- Kong
- Nginx
- Traefik
- Openshift Routes
- Gloo Gateway
- Contour
- Cilium
- Envoy Gateway
- Gateway API



Pre/Post checks

Fully Automated Rollbacks



```
apiVersion: argoproj.io/v1alpha1
kind: AnalysisTemplate
metadata:
  name: success-rate
spec:
  args:
  - name: service-name
  metrics:
  - name: success-rate
    interval: 5m
    # NOTE: prometheus queries return results in the form of a vector.
    # So it is common to access the index 0 of the returned array to obtain the value
    successCondition: result[0] >= 0.95
    failureLimit: 3
  provider:
    prometheus:
      address: http://prometheus.example.com:9090
      query: |
        sum(irate(
          istio_requests_total{reporter="source",destination_service=~"{{args.service-name}}",response_code!~"5.*"}[5m]
        )) /
        sum(irate(
          istio_requests_total{reporter="source",destination_service=~"{{args.service-name}}"}[5m]
        ))|
```

Supported Metric providers

- Prometheus
- Datadog
- New Relic
- Wavefront
- CloudWatch
- Apache SkyWalking
- Graphite
- Custom Web call
- Custom Job
- Custom plugin



Argo Rollouts - Other features

- A/B testing
- Header based routing
- Argo CD UI extension
- Notifications
- Plugins
- CLI/Metrics

The screenshot shows the Argo Rollouts UI for a deployment named 'rollouts-demo'. The interface is divided into several sections:

- Steps:** A vertical list of steps defined for the rollout:
 - Set Weight: 20%
 - Pause
 - Set Weight: 40%
 - Pause: 10s
 - Set Weight: 60%
 - Pause: 10s
 - Set Weight: 80%
 - Pause: 10s
- Summary:** Displays the current strategy (Canary), step count (1/8), and actual weight (20).
- Containers:** Shows the containers involved: 'rollouts-demo' and 'argoproj/rollouts-demo:yellow'.
- Revisions:** Lists the current revisions:
 - Revision 2: argoproj/rollouts-demo:yellow (canary)
 - Revision 1: argoproj/rollouts-demo:blue (stable)
- Buttons:** Top right buttons include 'Restart', 'Abort', 'Promote', and 'Promote full'.



SYDNEY



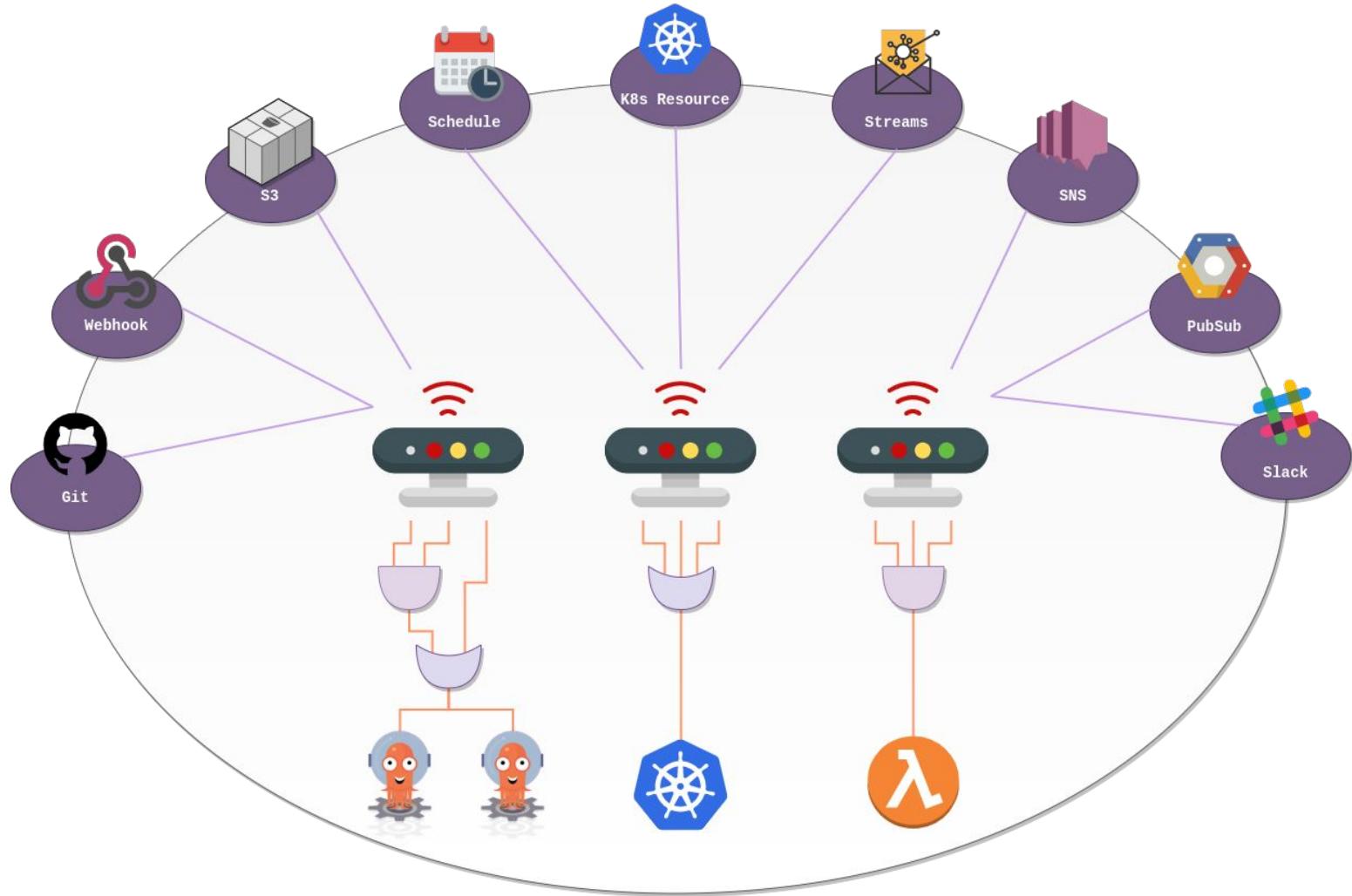
5 & 6 SEPTEMBER

Argo Events

Argo Events

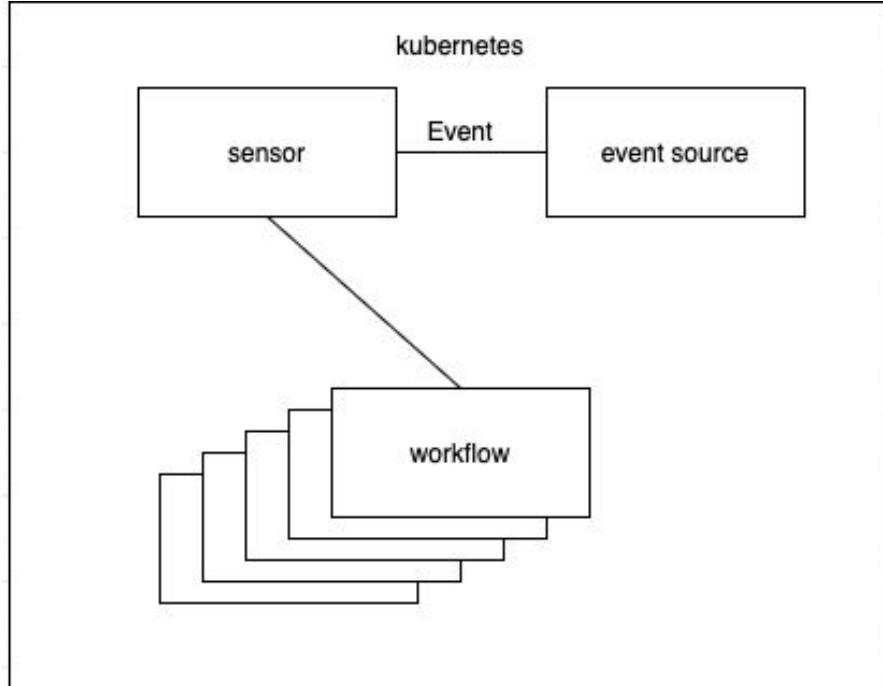
- Generic Event mechanism
- Kubernetes native
- Connects several sources such as AMPQ, SQS, PubSub, Kafka, MQTT, Slack, Webhooks
- cloudevents.io compliant





Argo Events entities

- **EventSource** - where to read events from
- **Trigger** - what to do when an event happens
- **Sensor** - connects sources and triggers
- **EventBus** - connects Sources and Sensors together



Creating events from webhooks

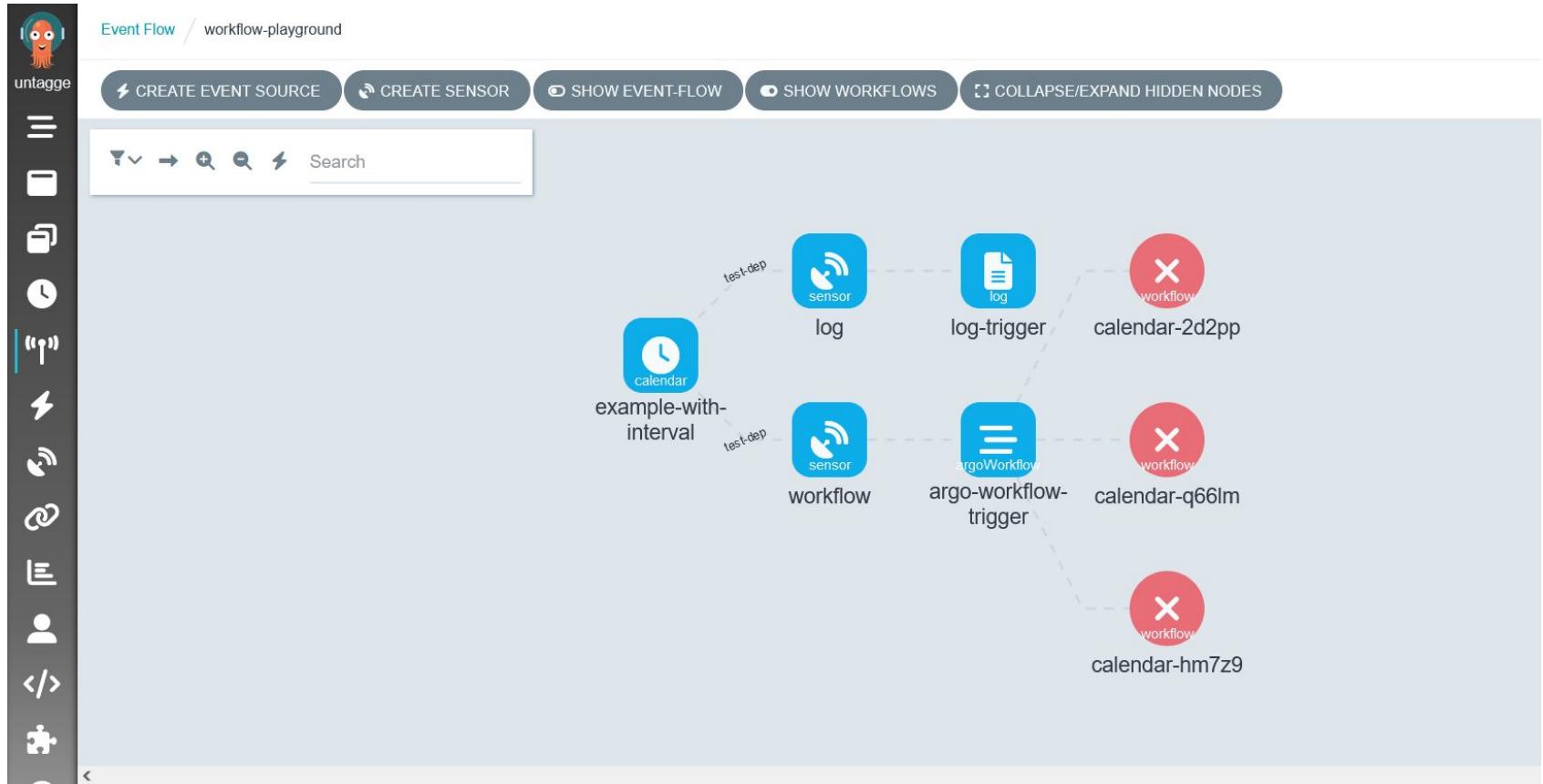
```
apiVersion: argoproj.io/v1alpha1
kind: EventSource
metadata:
  name: webhook
spec:
  service:
    ports:
      - port: 12000
        targetPort: 12000
  webhook:
    # event-source can run multiple HTTP servers. Simply define a unique port to start a new HTTP server
  example:
    # port to run HTTP server on
    port: "12000"
    # endpoint to listen to
    endpoint: /example
    # HTTP request method to allow. In this case, only POST requests are accepted
    method: POST
```

```
apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: webhook
spec:
  template:
    serviceAccountName: operate-workflow-sa
dependencies:
  - name: test-dep
    eventSourceName: webhook
    eventName: example
triggers:
  - template:
      name: webhook-workflow-trigger
      k8s:
        operation: create
        source:
          resource:
            apiVersion: argoproj.io/v1alpha1
            kind: Workflow
            metadata:
              generateName: webhook-
            spec:
              entrypoint: whalesay
            arguments:
              parameters:
                - name: message
                  # the value will get overridden by event payload from test-dep
                  value: hello world
```

Starting a workflow from a webhook event



Argo Workflows UI also works for Argo Events



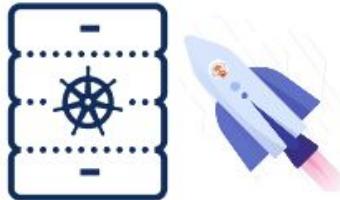
SYDNEY



5 & 6 SEPTEMBER

Use cases

Argo CD and Argo Rollouts



QA



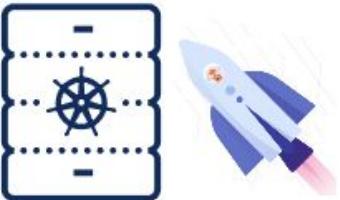
Staging



Production US



UAT



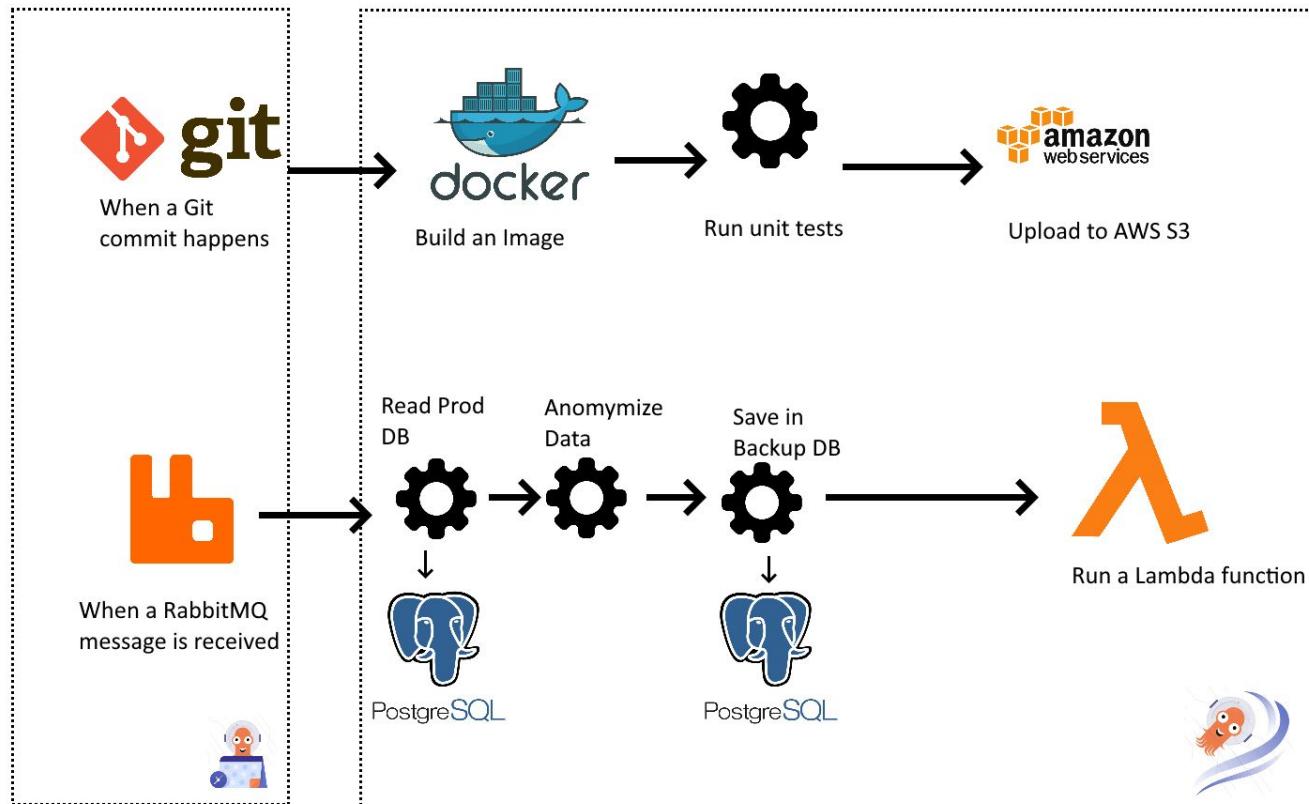
Load Testing



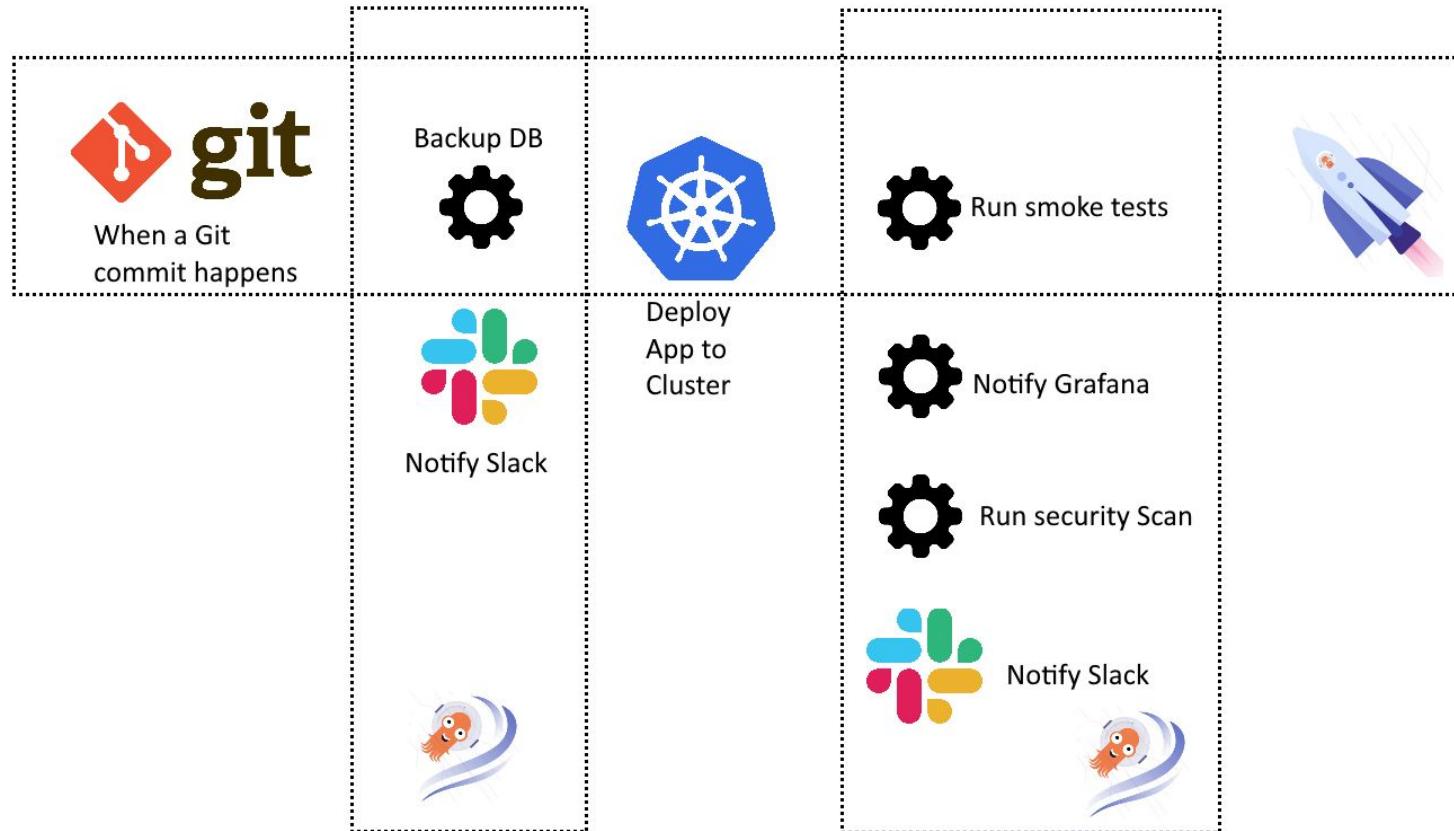
Production EU



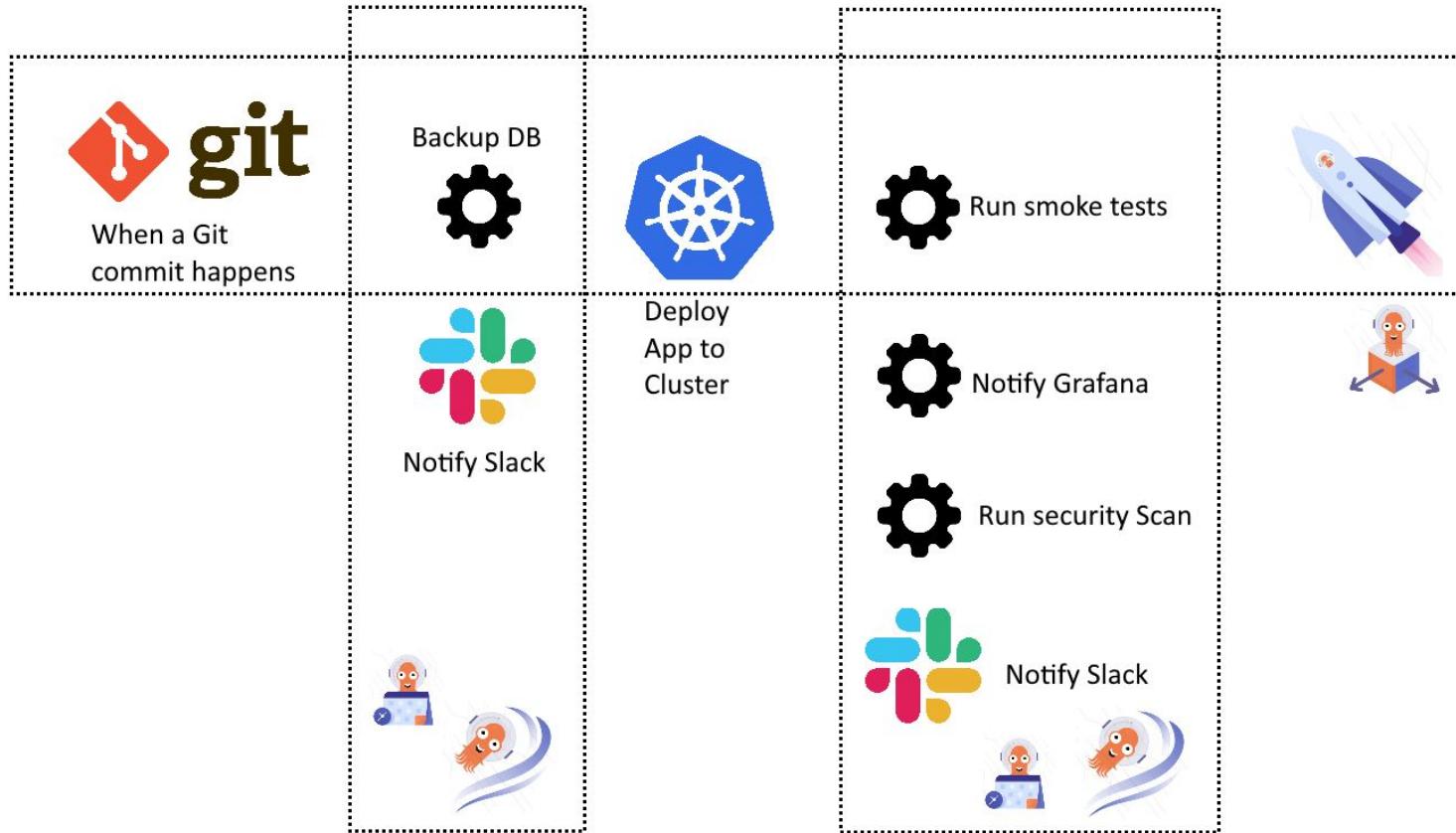
Argo Workflows and Argo Events



Argo CD and Argo Workflows



All 4 Argo projects (developer portal)



Thank you

- Questions:
kostis.kapelonis@octopus.com
- GitOps/Argo CD certification
<http://learning.codefresh.io>
- CNCF Slack <https://slack.cncf.io/>
- Blog <https://blog.argoproj.io/>



SYDNEY



5 & 6 SEPTEMBER

Backup slides

GitOps Principles

v1.0.0

1 Declarative

A system managed by GitOps must have its desired state expressed declaratively.

2 Versioned and Immutable

Desired state is stored in a way that enforces immutability, versioning and retains a complete version history.

3 Pulled Automatically

Software agents automatically pull the desired state declarations from the source.

4 Continuously Reconciled

Software agents continuously observe actual system state and attempt to apply the desired state.



Project history

1. Startup Applatix was formed (2015)
2. Argo Workflows was released by Applatix (2017)
3. Applatix was acquired by Intuit (2018)
4. Argo CD and Argo Rollouts were created by Intuit (2018 and 2019)
5. Argo Events was donated by Blackrock Inc (2018)
6. Incubating open source software of the CNCF (accepted in 2020)
7. Graduated from CNCF in 2022 😎

