

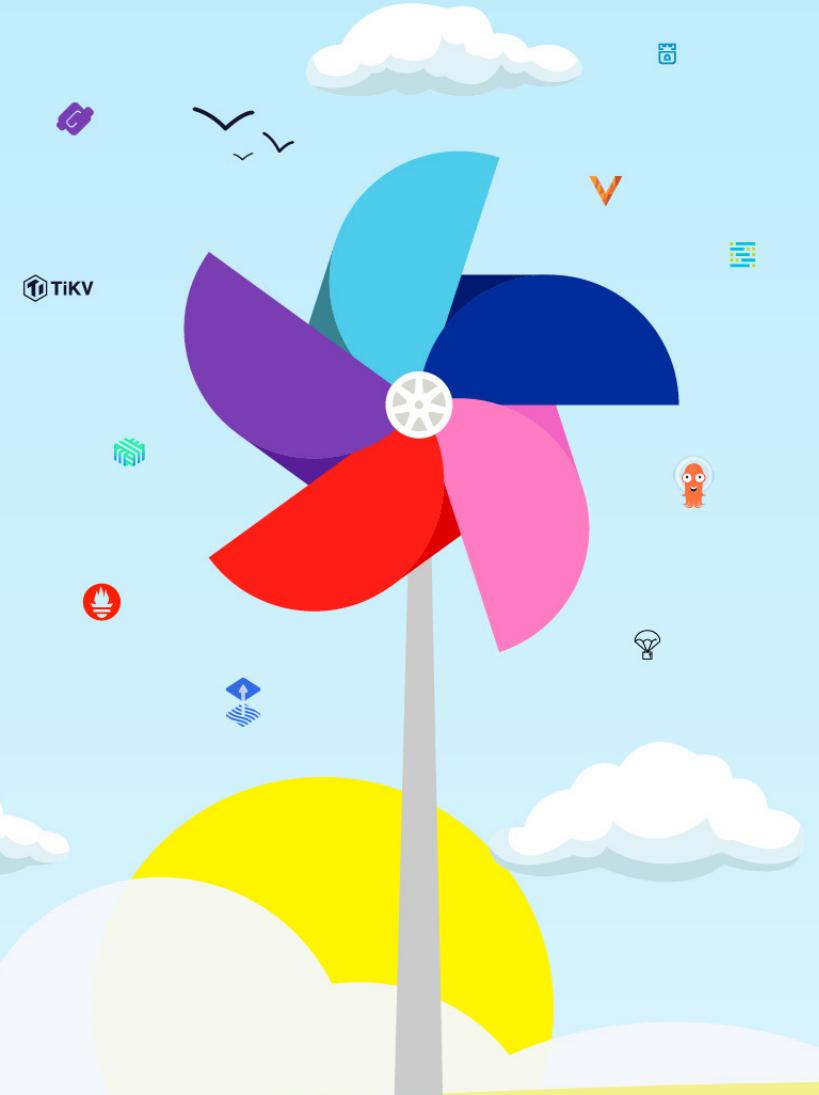


KubeCon



CloudNativeCon

Europe 2023



Tips from the Trenches GitOps at Adobe

*Larisa-Andreea Dănilă
Ionuț-Maxim Mărgelatu*

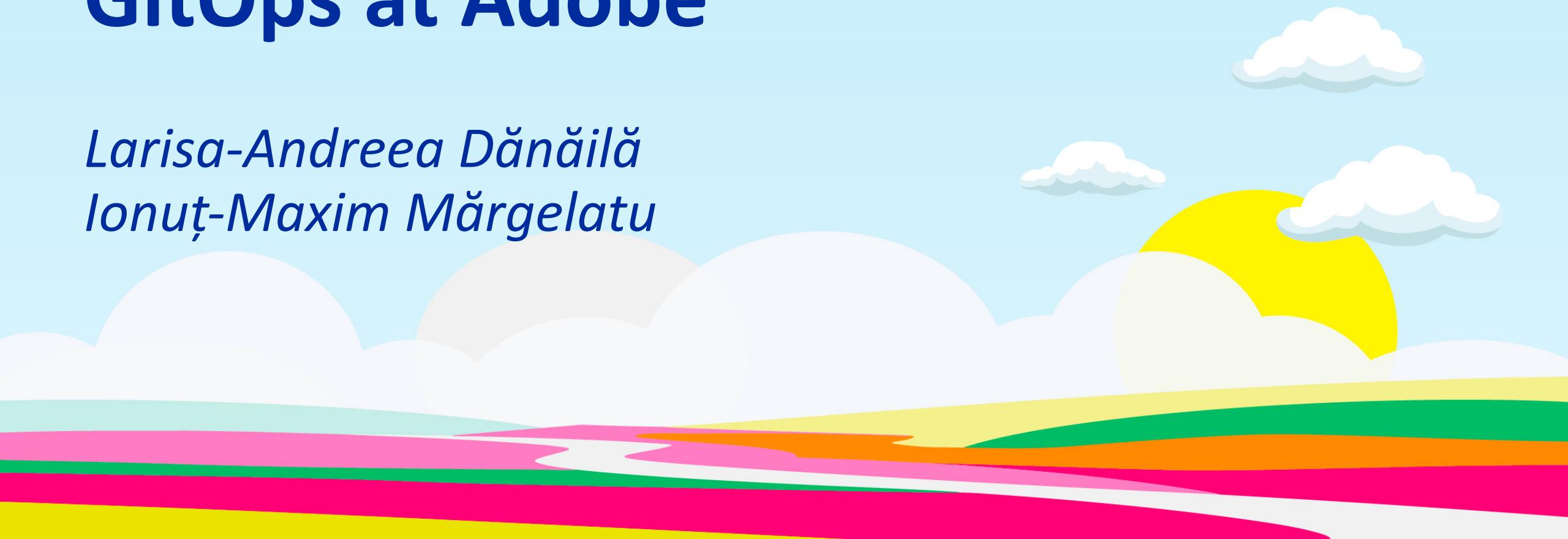


KubeCon



CloudNativeCon

Europe 2023



The story so far

2020

Company-wide migration
to Kubernetes

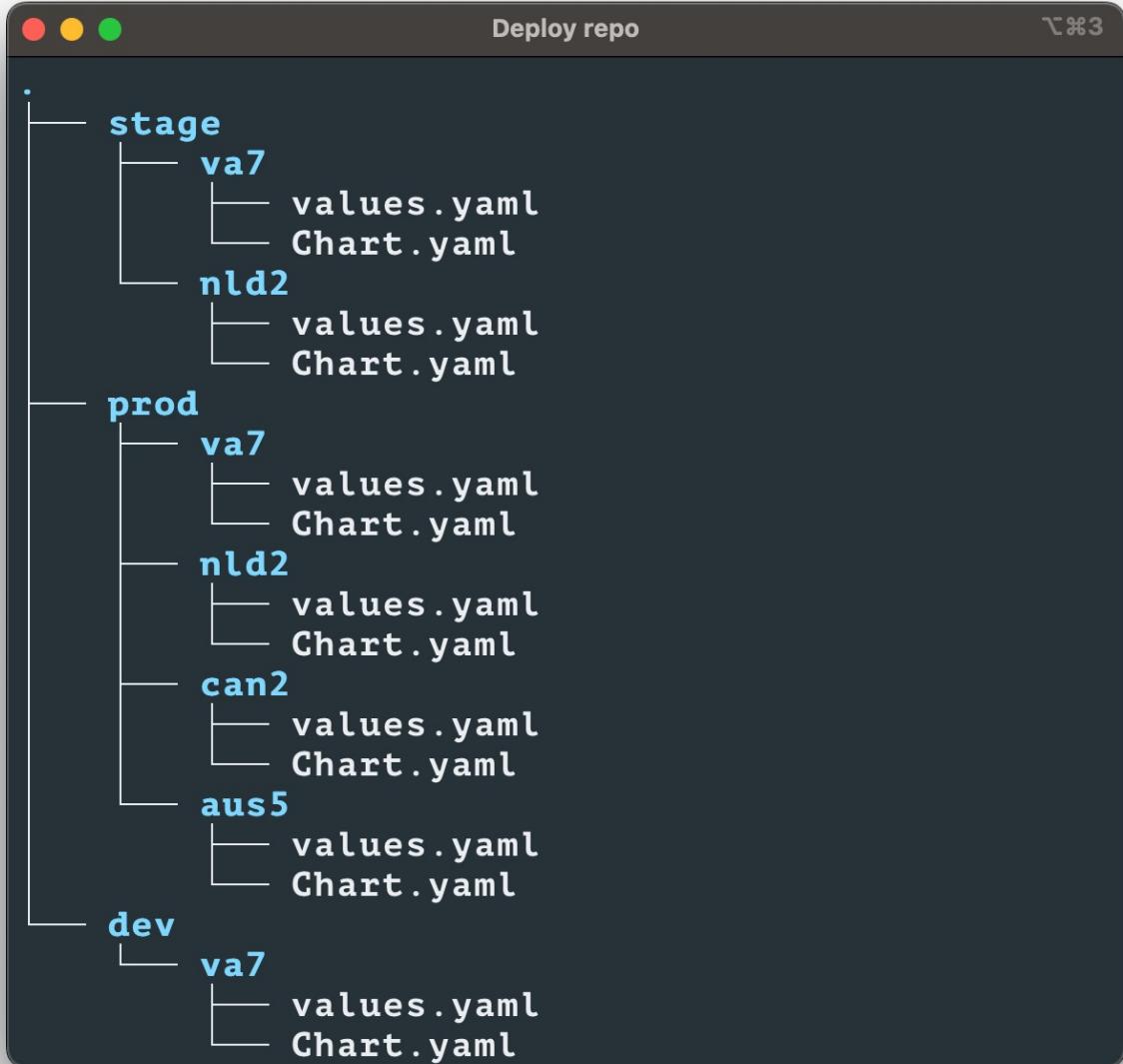
2022

Deployment platform on
top of the Argo project

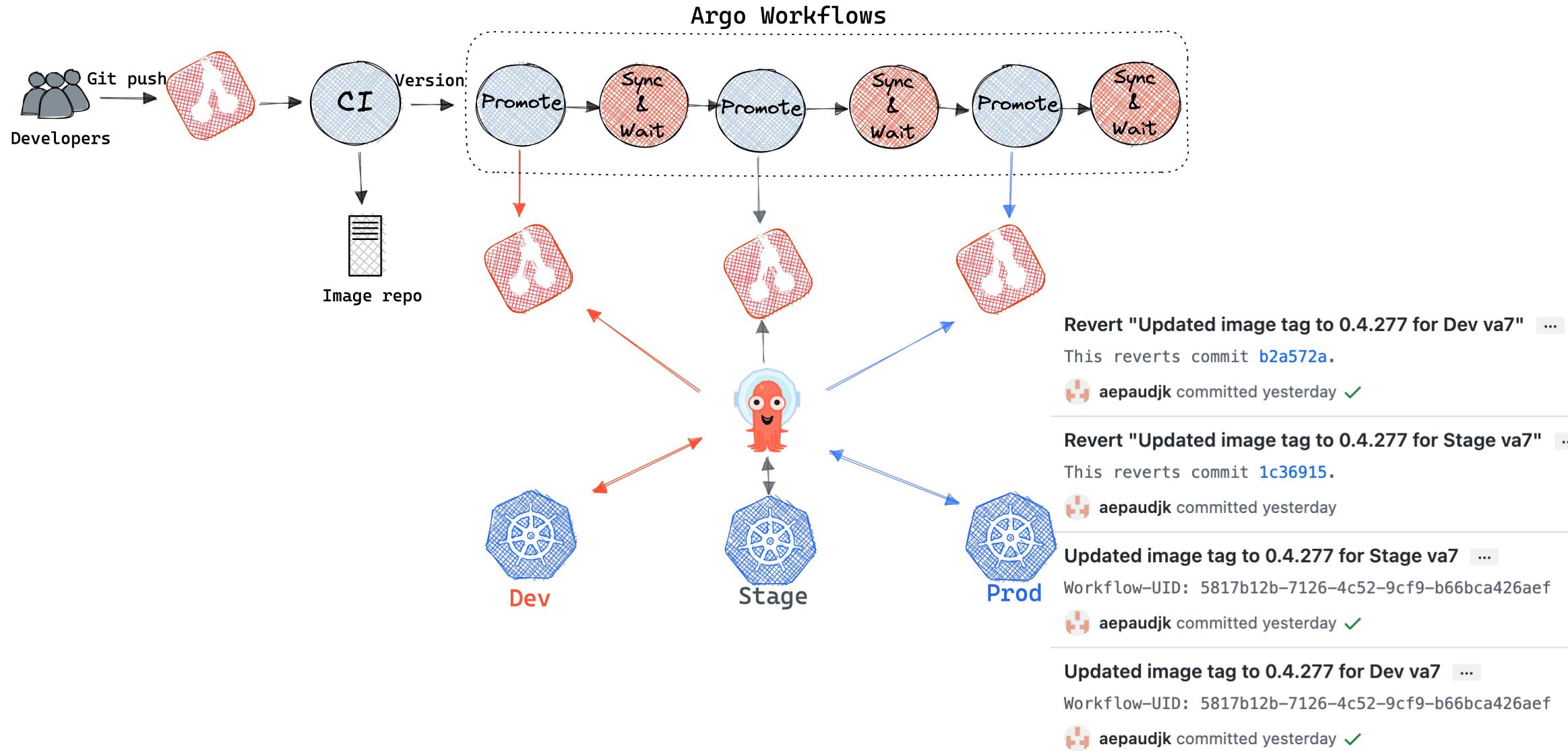
2023

Streamlining DevOps
practices with Argo

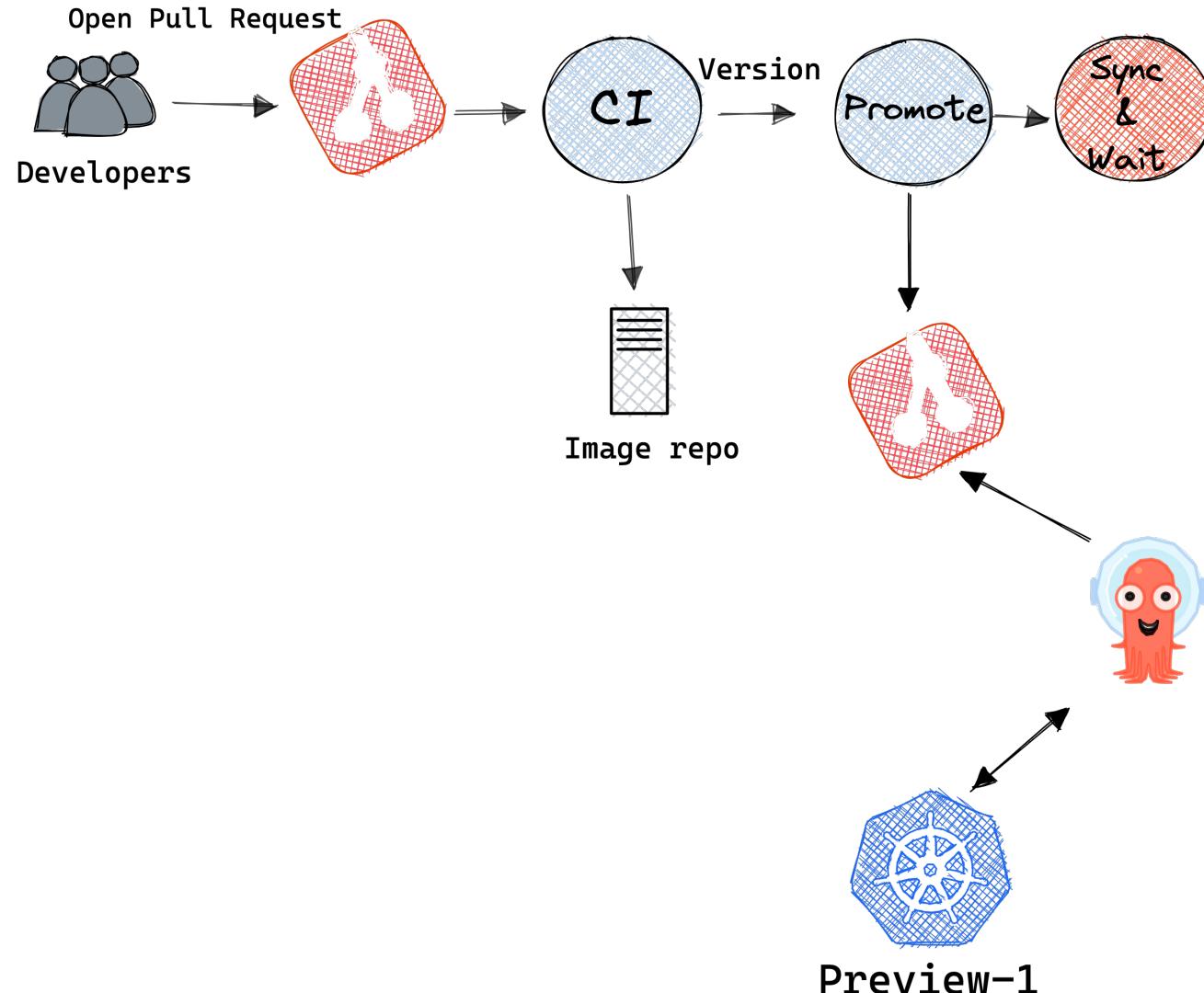
Trunk based development



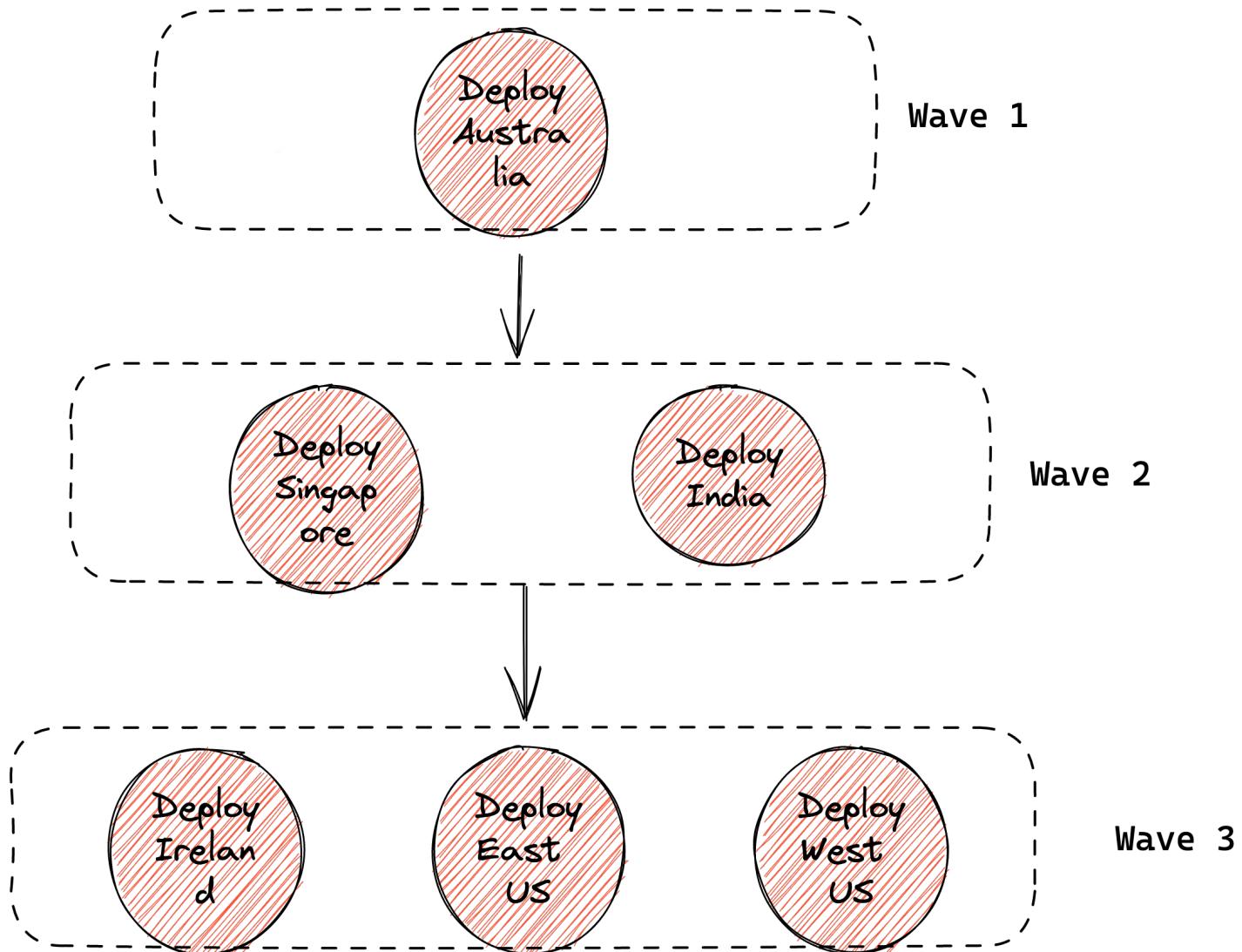
CICD and automated rollbacks



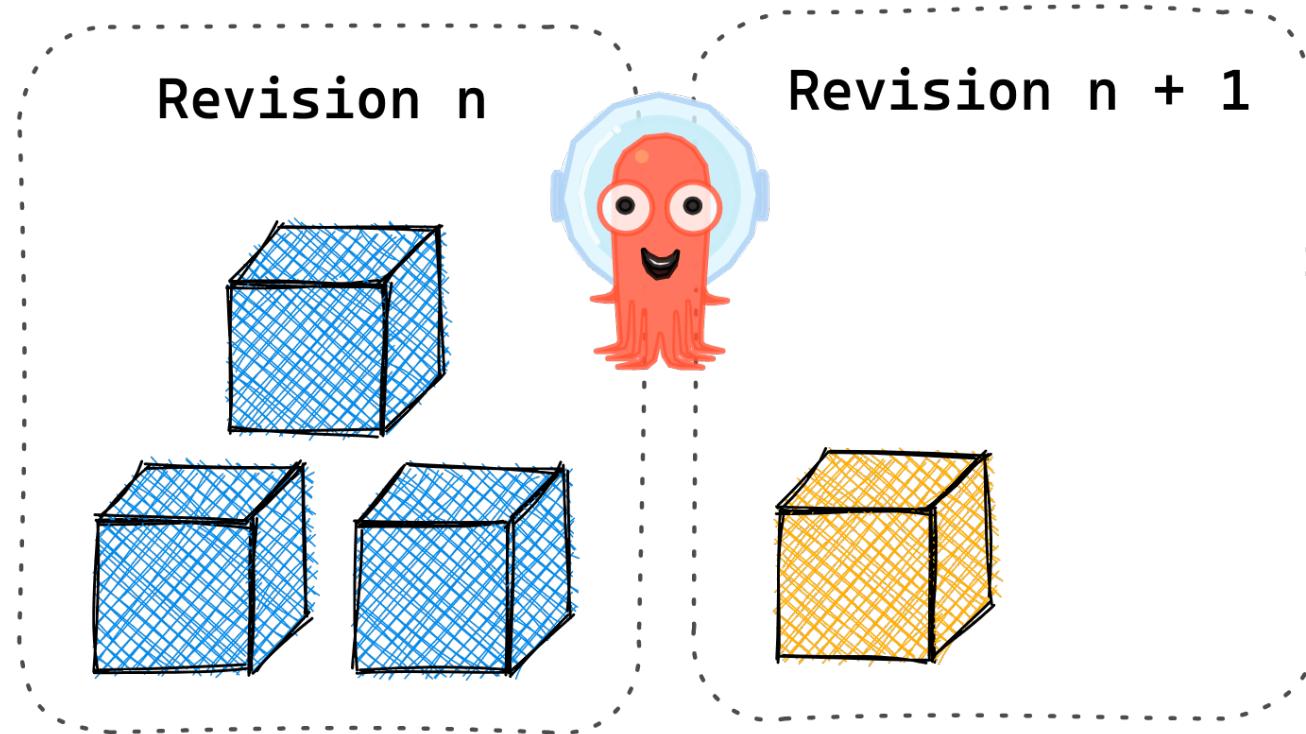
Advanced practices (1)



Advanced practices (2)



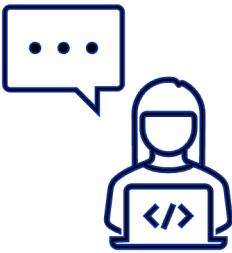
Advanced practices (3)



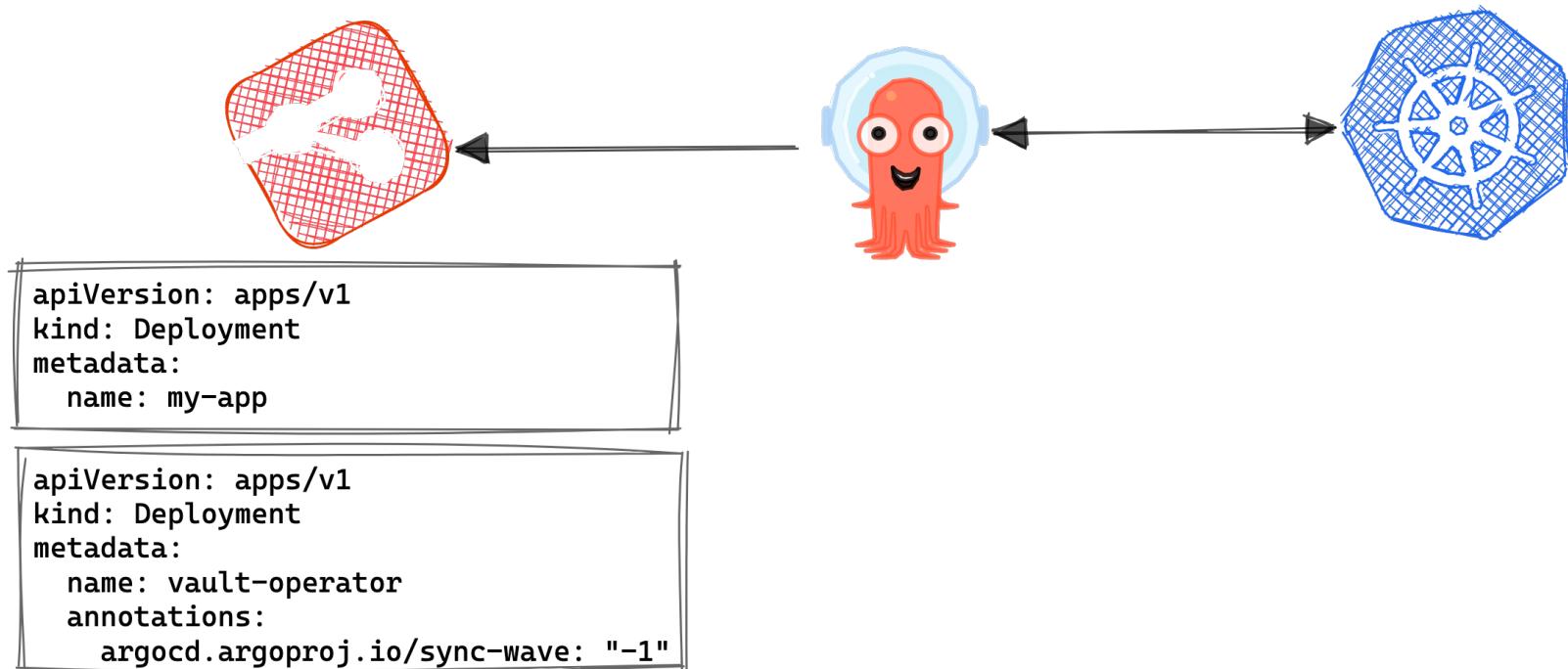
Challenges and obstacles

With great power comes great *creativity*

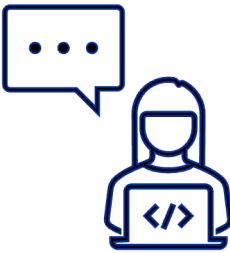
Synchronization phases



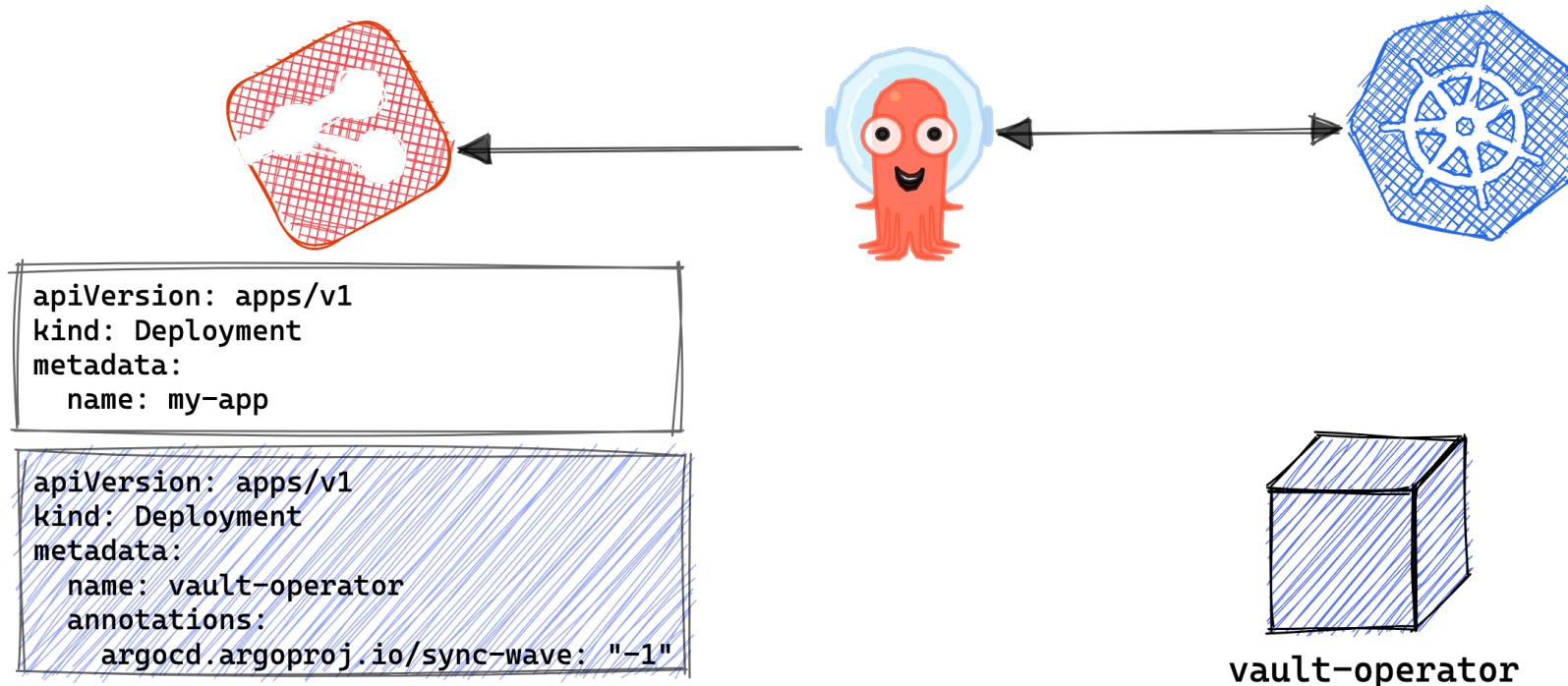
"I want to deploy my applications in a particular order."



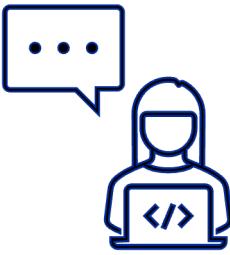
Synchronization phases



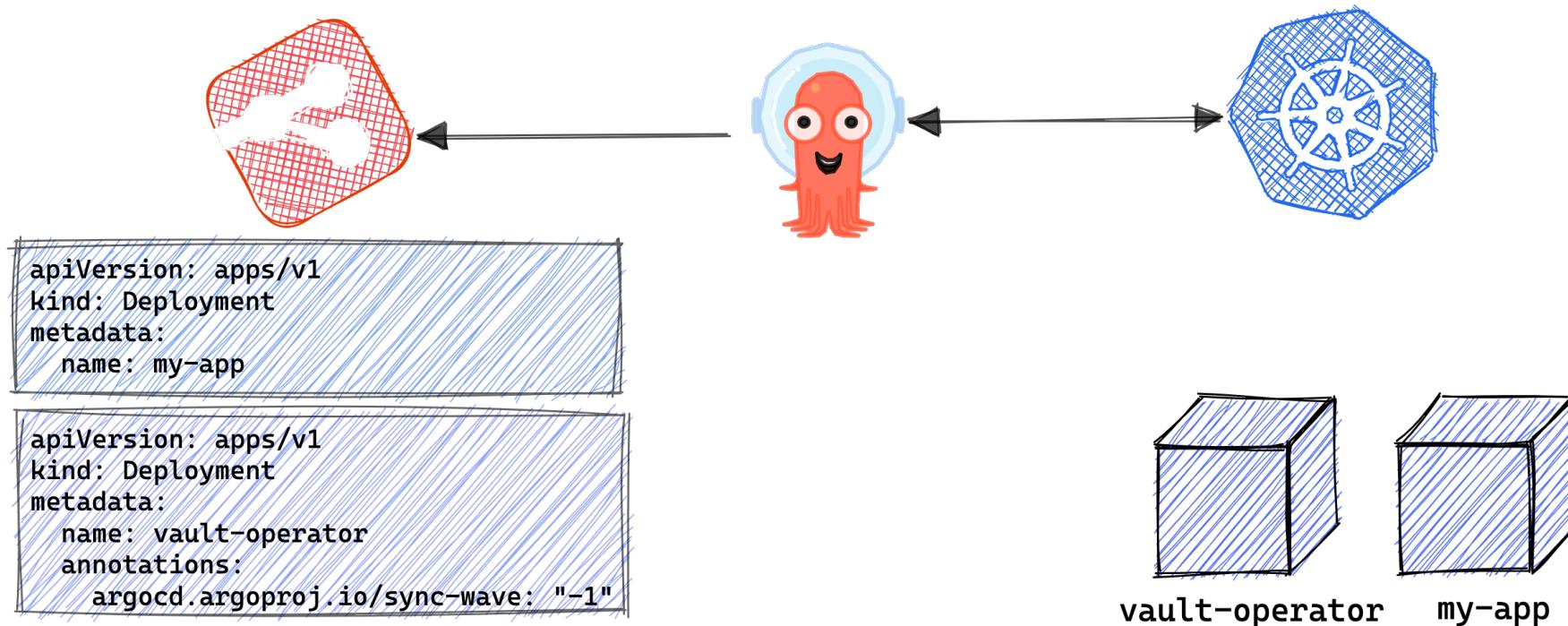
"I want to deploy my applications in a particular order."



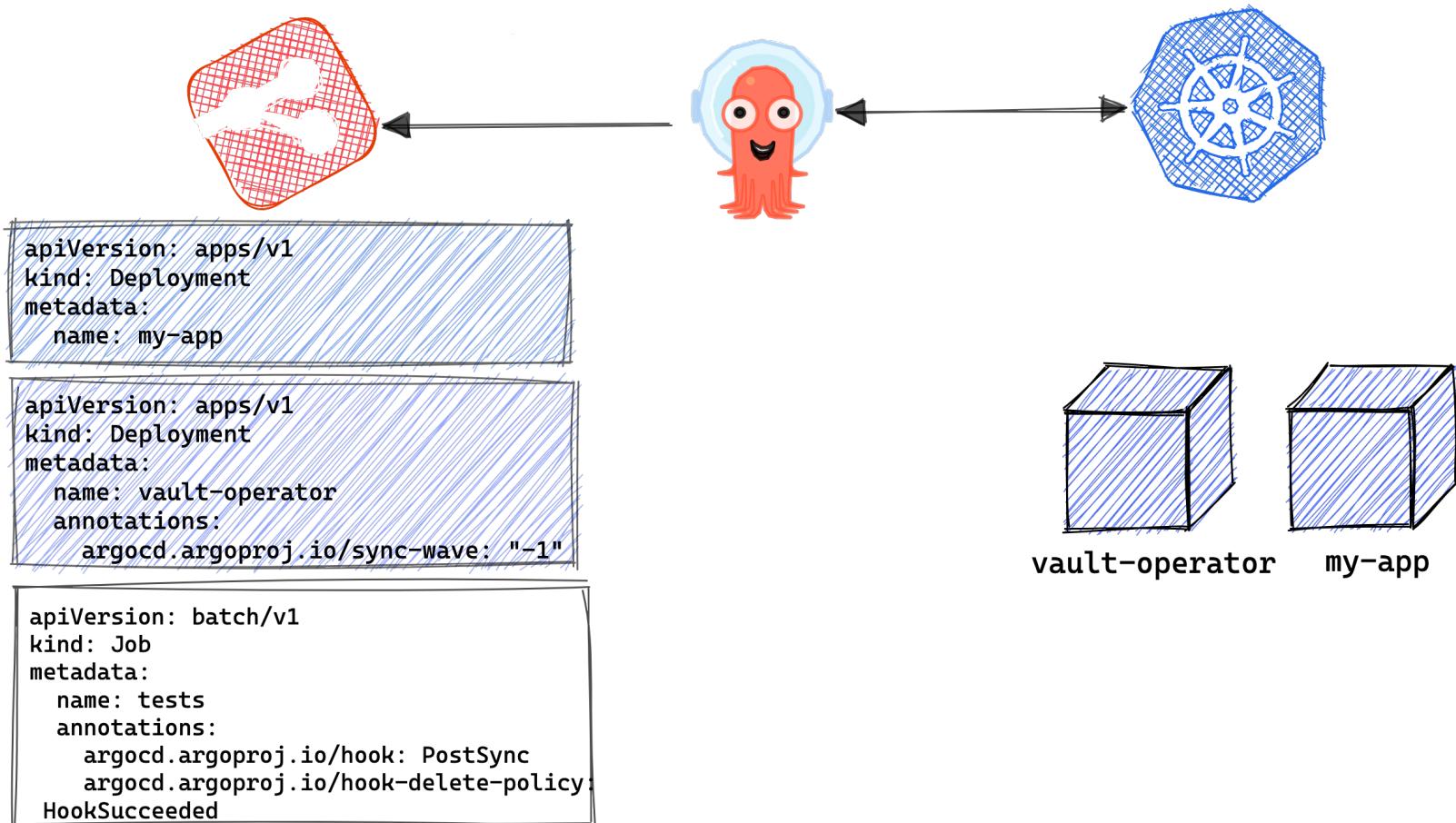
Synchronization phases



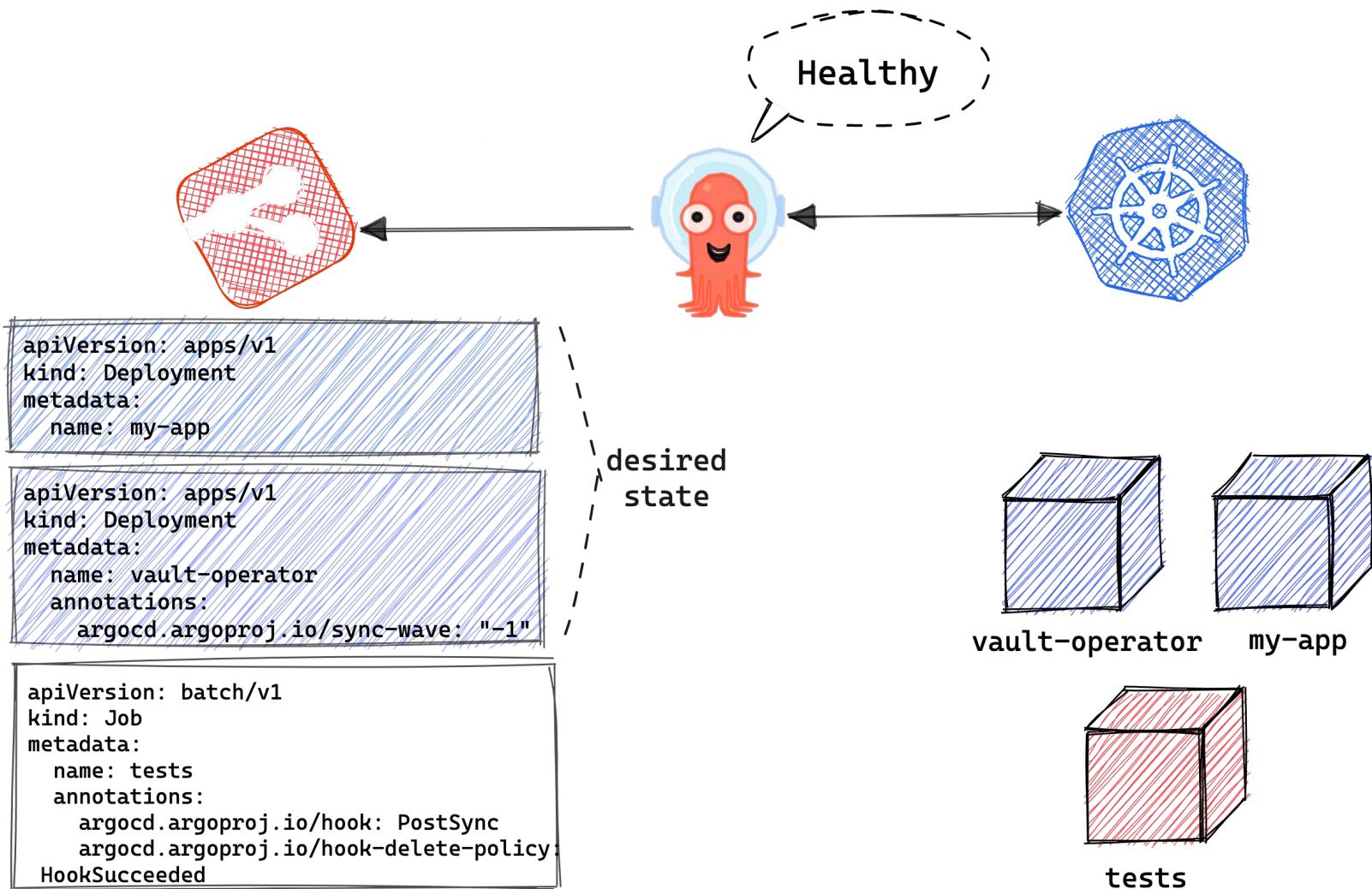
"I want to deploy my applications in a particular order."



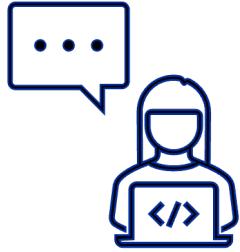
Hooks are not desired state



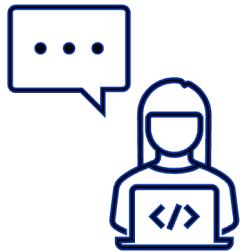
Hooks are not desired state



Running tests

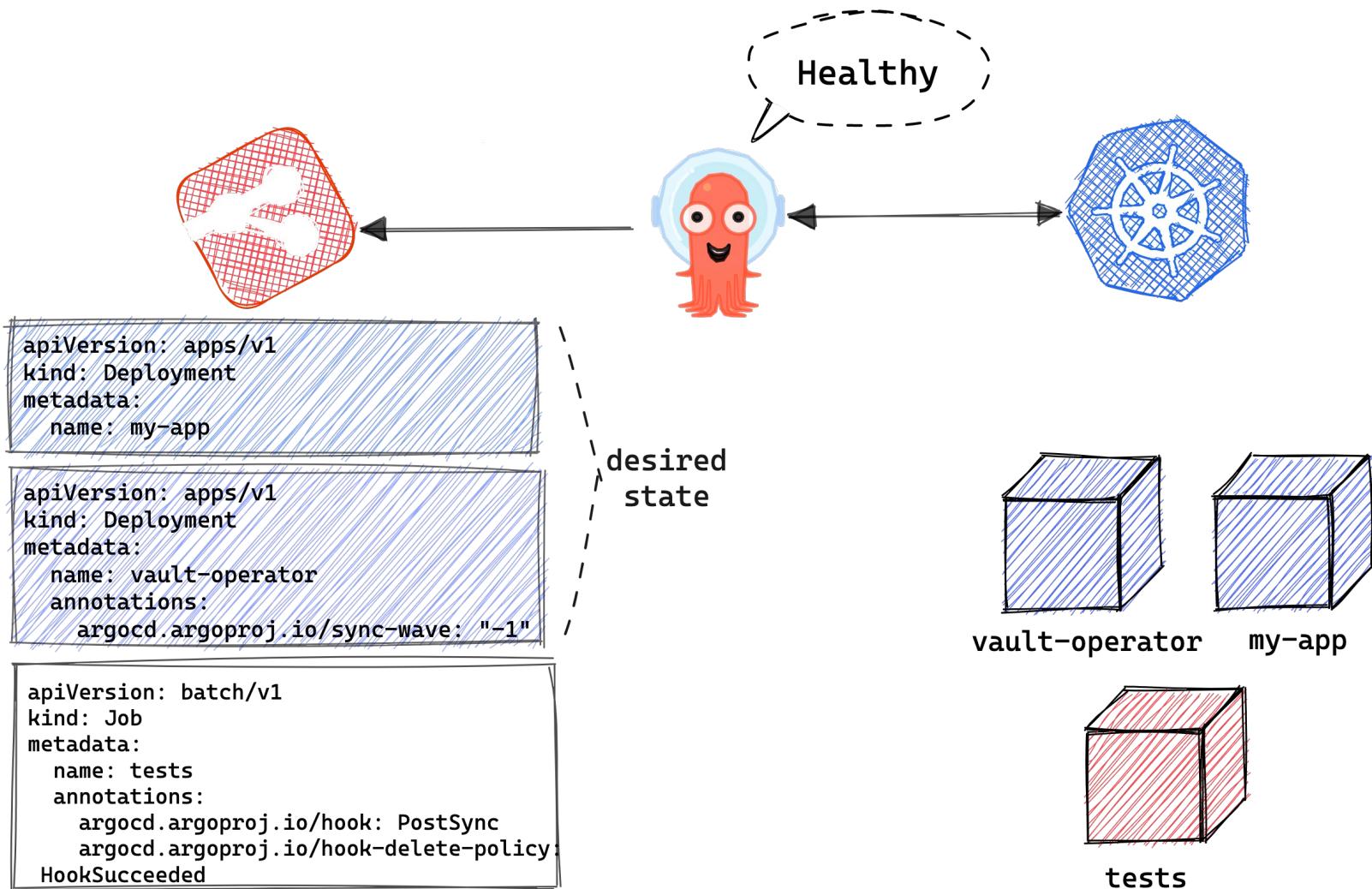


"I want to run tests post deployment.."

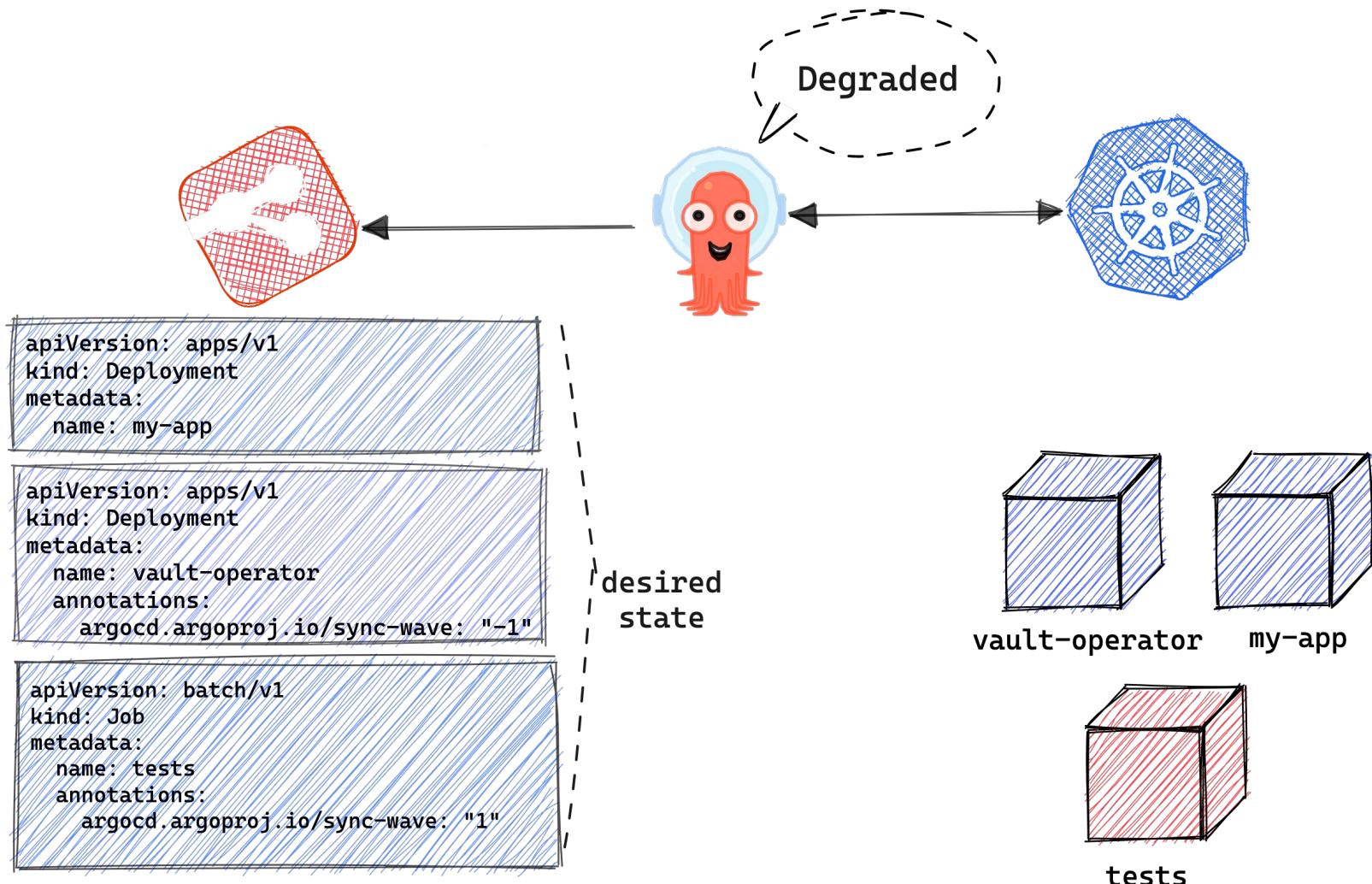


"...for dev environments and streaming applications."

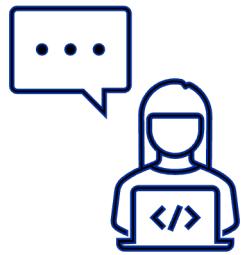
Assert health outside of Argo CD



Make tests desired state



Environment configuration

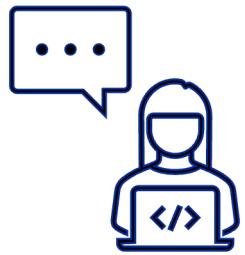


"I want to continuously deploy new changes (outside image tags)."

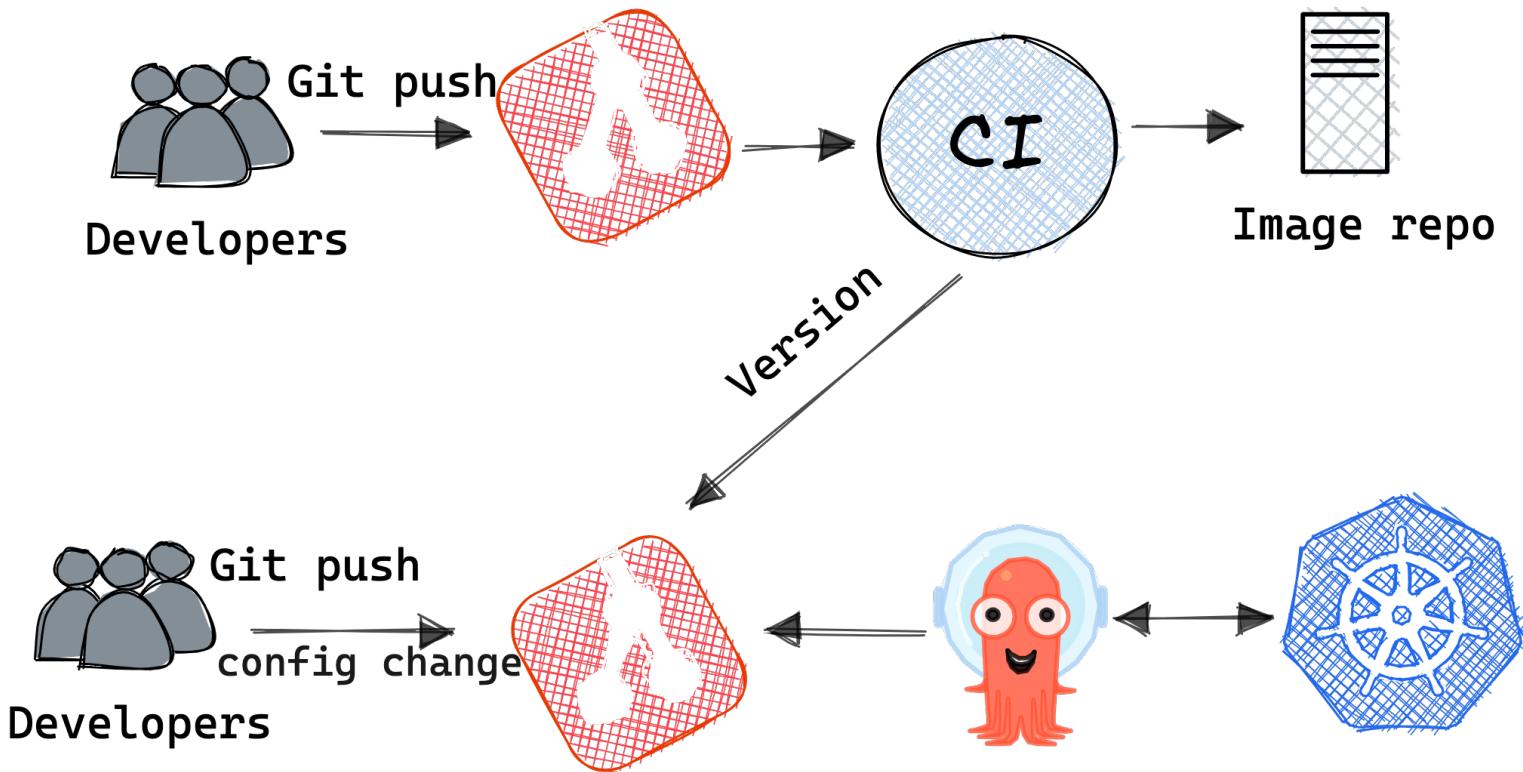
As a developer, I often need to change:

- Business configuration
- Deployment configuration

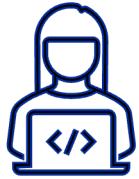
Environment configuration



“Opening 2 PRs breaks my Continuous Deployment.”



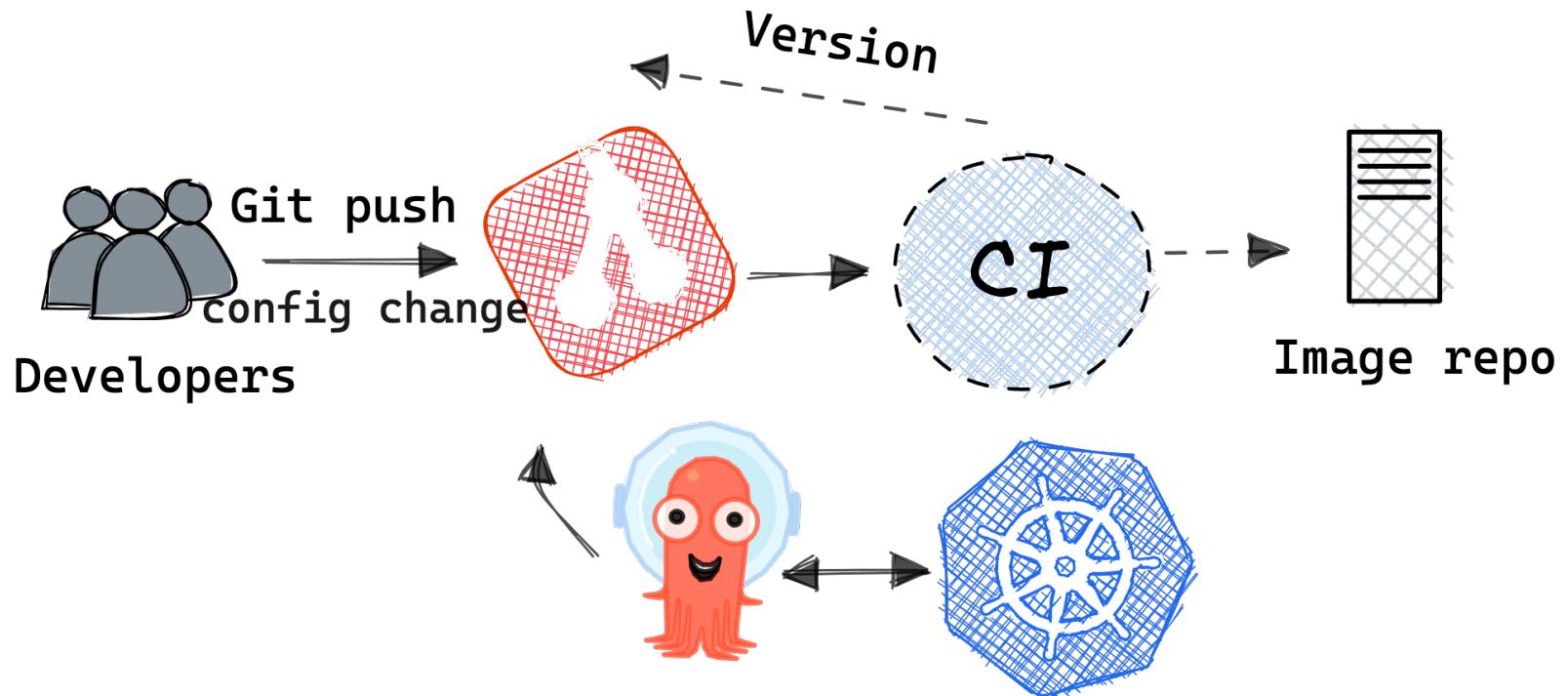
Environment configuration



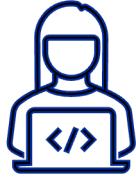
"Opening 2 PRs breaks my Continuous Deployment."



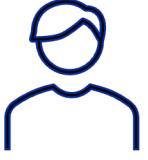
"Just do it in a single repo then."



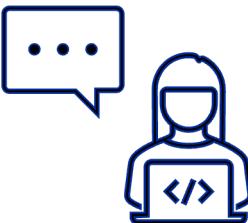
Environment configuration



“Opening 2 PRs breaks my Continuous Deployment.”

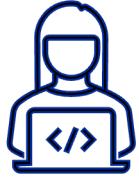


“Just do it in a single repo then.”

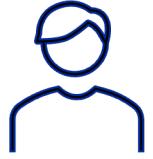


“My CI and part of my CD start at the same time.”

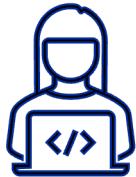
Environment configuration



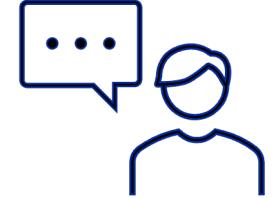
“Opening 2 PRs breaks my Continuous Deployment.”



“Just do it in a single repo then.”

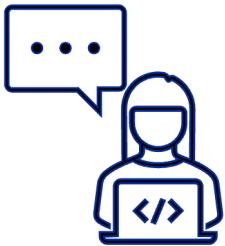


“My CI and part of my CD start at the same time.”

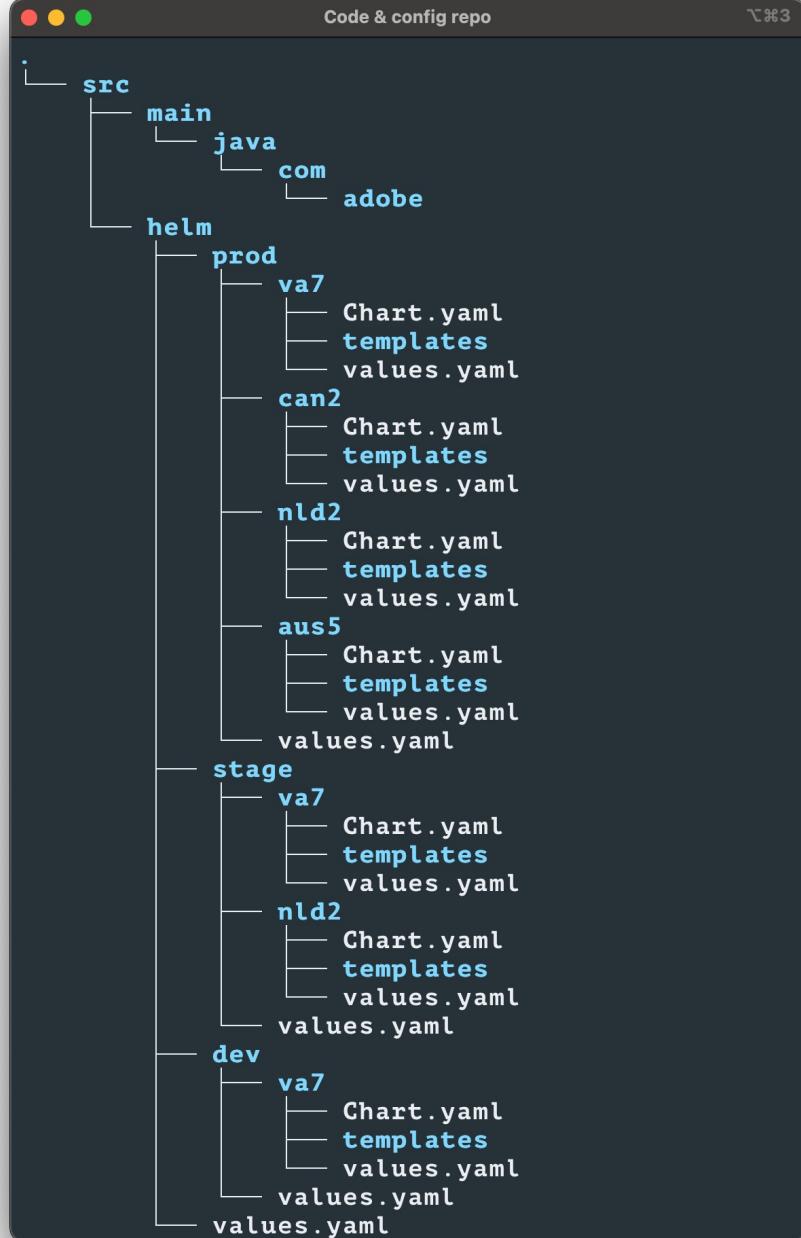


“Can’t you use a PreSync hook to wait for the CI?”

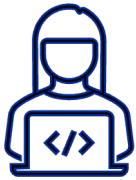
Environment configuration



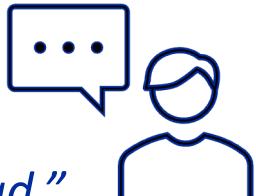
“I don’t want to duplicate my k8s manifests.”



Environment configuration



"I don't want to duplicate my k8s manifests."



"Use an umbrella chart instead."

The screenshot shows a dark-themed file explorer window titled "Code & config repo". The directory structure is as follows:

- src
 - main
 - java
 - com
 - adobe
 - helm
 - prod
 - va7
 - Chart.yaml
 - templates
 - values.yaml
 - can2
 - Chart.yaml
 - templates
 - values.yaml
 - nld2
 - Chart.yaml
 - templates
 - values.yaml
 - aus5
 - Chart.yaml
 - templates
 - values.yaml
 - values.yaml
 - stage
 - va7
 - Chart.yaml
 - templates
 - values.yaml
 - nld2
 - Chart.yaml
 - templates
 - values.yaml
 - dev
 - va7
 - Chart.yaml
 - templates
 - values.yaml
 - values.yaml

Environment configuration

The screenshot shows a terminal window titled "Code & config repo" displaying a file tree structure. The root directory contains a "src" folder, which contains a "main" folder and a "k8s" folder. The "main" folder contains a "java" folder, which has a "com" folder, which in turn has an "adobe" folder. The "k8s" folder contains several subfolders: "stage", "nld2", "prod", "can2", "aus5", "dev", and "helm". Each of these subfolders contains a "values.yaml" file and a "Chart.yaml" file. The "helm" folder also contains "templates" and "Chart.yaml". The terminal interface includes standard Mac OS X window controls (red, yellow, green) and a status bar at the bottom right.

```
.  
└── src  
    ├── main  
    │   └── java  
    │       └── com  
    │           └── adobe  
    └── k8s  
        ├── values.yaml  
        ├── stage  
        │   ├── values.yaml  
        │   └── va7  
        │       ├── values.yaml  
        │       └── Chart.yaml  
        ├── nld2  
        │   ├── values.yaml  
        │   └── Chart.yaml  
        ├── prod  
        │   ├── values.yaml  
        │   └── va7  
        │       ├── values.yaml  
        │       └── Chart.yaml  
        ├── nld2  
        │   ├── values.yaml  
        │   └── Chart.yaml  
        ├── can2  
        │   ├── values.yaml  
        │   └── Chart.yaml  
        ├── aus5  
        │   ├── values.yaml  
        │   └── Chart.yaml  
        └── dev  
            ├── values.yaml  
            └── va7  
                ├── values.yaml  
                └── Chart.yaml  
└── helm  
    ├── values.yaml  
    ├── templates  
    └── Chart.yaml
```

Environment configuration

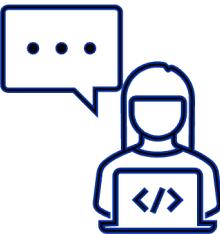
```
Umbrella chart

apiVersion: v2
name: Dev-va7
description: A Helm chart for Kubernetes
type: application
version: 0.0.1
appVersion: "0.0.1"
dependencies:
  - name: audit-consumer
    repository: https://artifactory-uw2.adobeitc.com/artifactory/helm-experienceplatform-release
    version: 0.4.2
  - name: audit-service
    repository: https://artifactory-uw2.adobeitc.com/artifactory/helm-experienceplatform-release
    version: 0.4.2
  - name: audit-functional-tests
    repository: https://artifactory-uw2.adobeitc.com/artifactory/helm-experienceplatform-release
    version: 0.4.2
    alias: tests
  - name: audit-tracer
    repository: https://artifactory-uw2.adobeitc.com/artifactory/helm-experienceplatform-release
    version: 0.2.1
  - name: audit-adx-exporter
    repository: https://artifactory-uw2.adobeitc.com/artifactory/helm-experienceplatform-release
    version: 0.1.3
```

Environment configuration

A screenshot of a terminal window titled "Code & config repo". The file tree structure is as follows:

- src
 - main
 - java
 - com
 - adobe
 - k8s
 - values.yaml
 - stage
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - nld2
 - values.yaml
 - Chart.yaml
 - prod
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - nld2
 - values.yaml
 - Chart.yaml
 - can2
 - values.yaml
 - Chart.yaml
 - aus5
 - values.yaml
 - Chart.yaml
 - dev
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - helm
 - values.yaml
 - templates
 - Chart.yaml

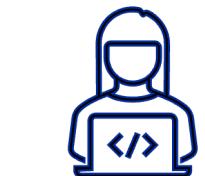


"I don't want to deploy my change directly to prod.."

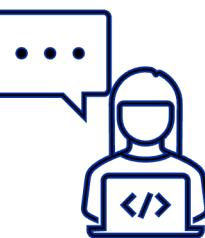
Environment configuration

A screenshot of a terminal window titled "Code & config repo". The file tree structure is as follows:

- src
 - main
 - java
 - com
 - adobe
 - k8s
 - values.yaml
 - stage
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - nld2
 - values.yaml
 - Chart.yaml
 - prod
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - nld2
 - values.yaml
 - Chart.yaml
 - can2
 - values.yaml
 - Chart.yaml
 - aus5
 - values.yaml
 - Chart.yaml
 - dev
 - values.yaml
 - va7
 - values.yaml
 - Chart.yaml
 - helm
 - values.yaml
 - templates
 - Chart.yaml

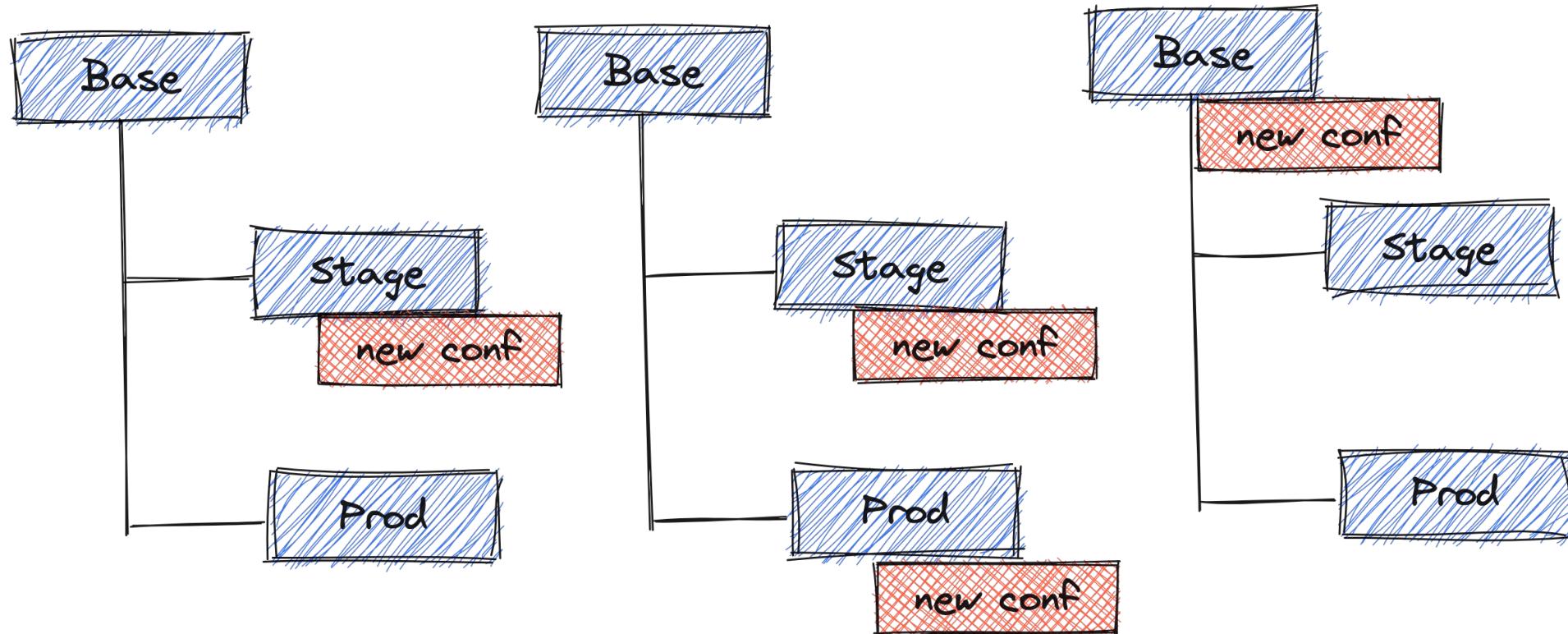


"I don't want to deploy my change directly to prod.."

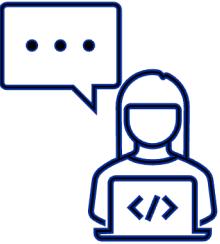


"... but I want to use DRY for my config definition."

Environment configuration



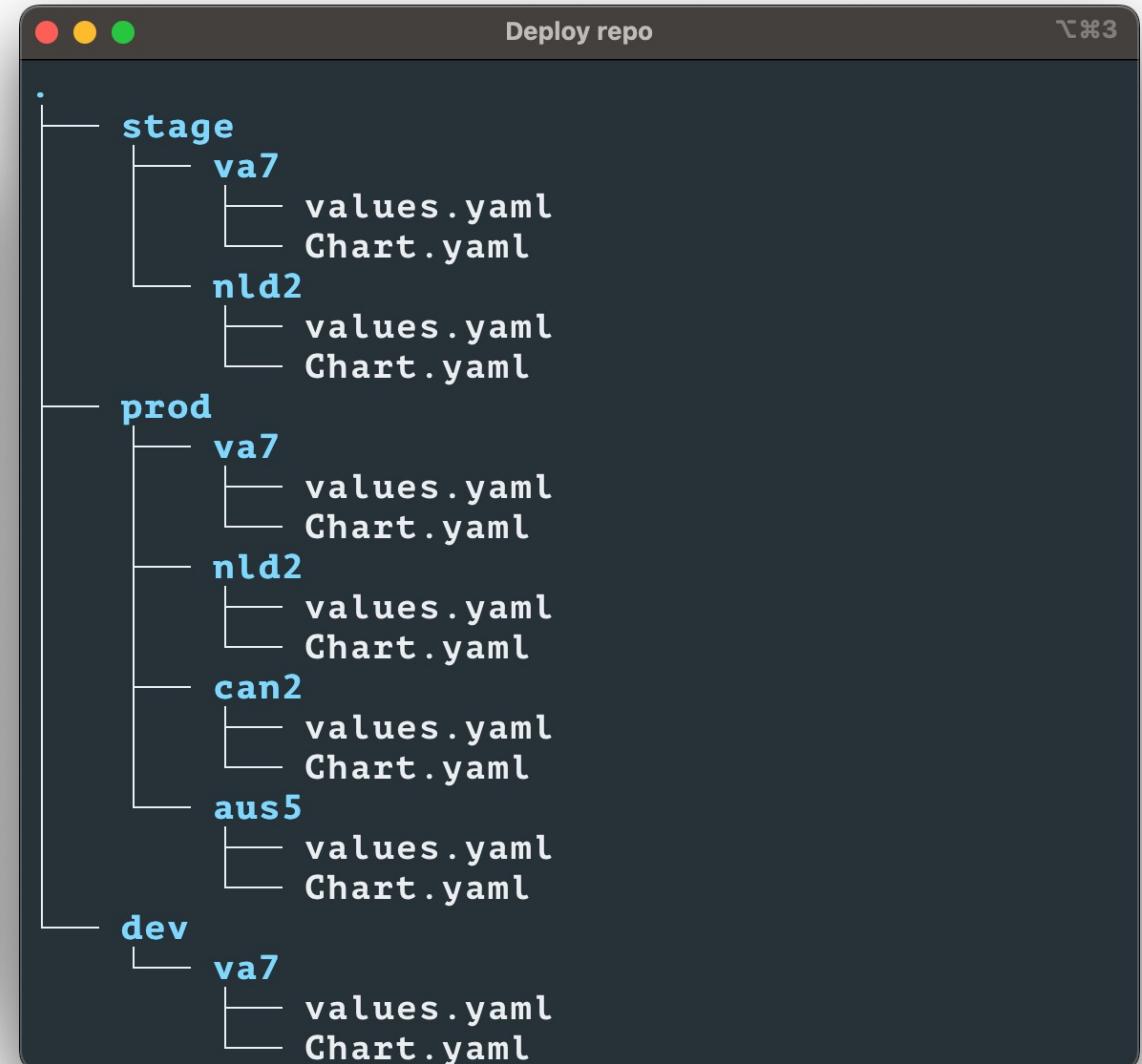
What we do



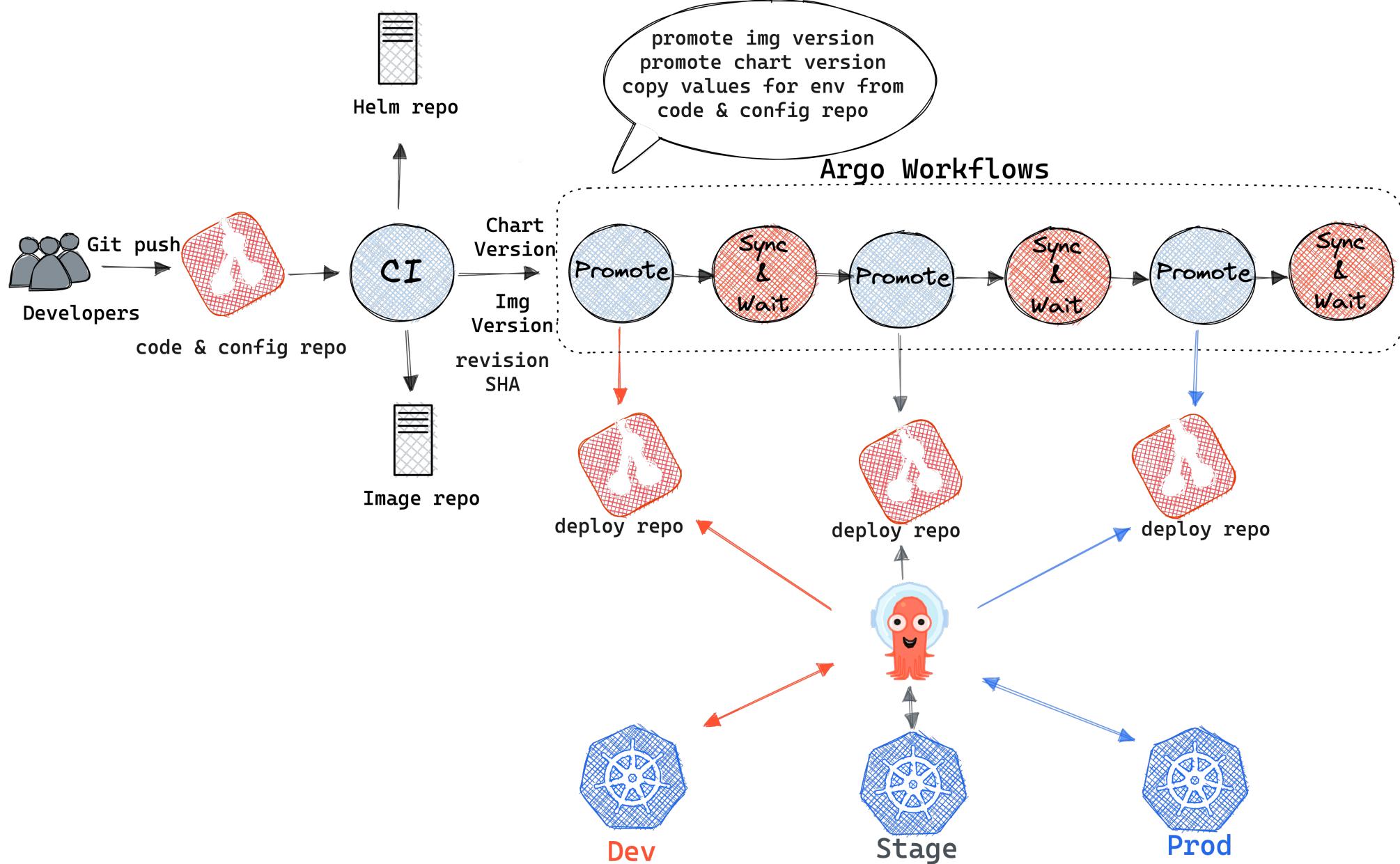
"I can't use the environment git state to declare charts or organize them with overlays."

Code & config repo

```
.  
└── src  
    ├── main  
    │   └── java  
    │       └── com  
    │           └── adobe  
    └── helm  
        ├── values.yaml  
        ├── values-stage-va7.yaml  
        ├── values-stage-nld2.yaml  
        ├── values-prod-va7.yaml  
        ├── values-prod-nld2.yaml  
        ├── values-prod-can2.yaml  
        ├── values-prod-aus5.yaml  
        ├── values-dev-va7.yaml  
        └── templates  
            └── Chart.yaml
```



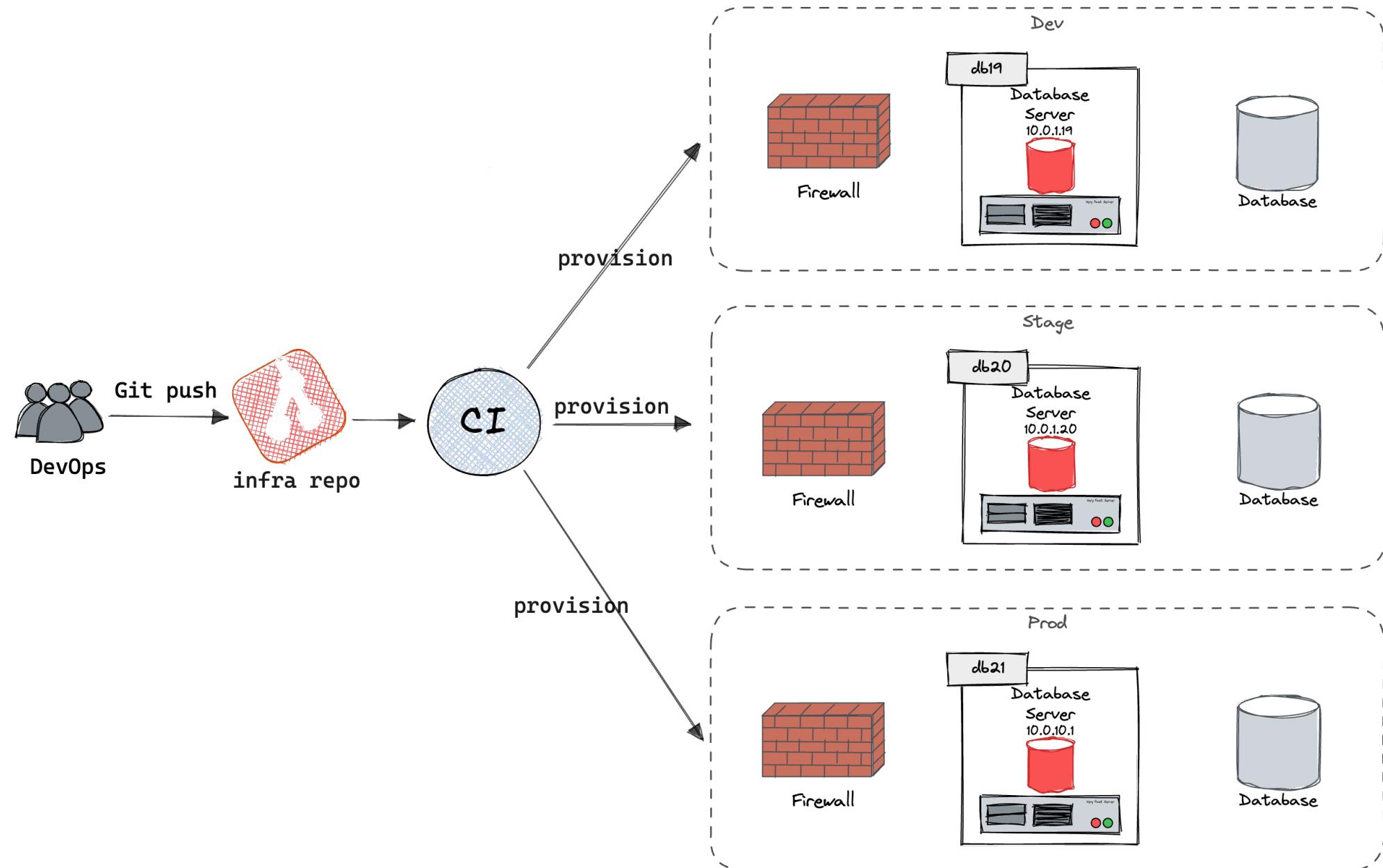
What we do



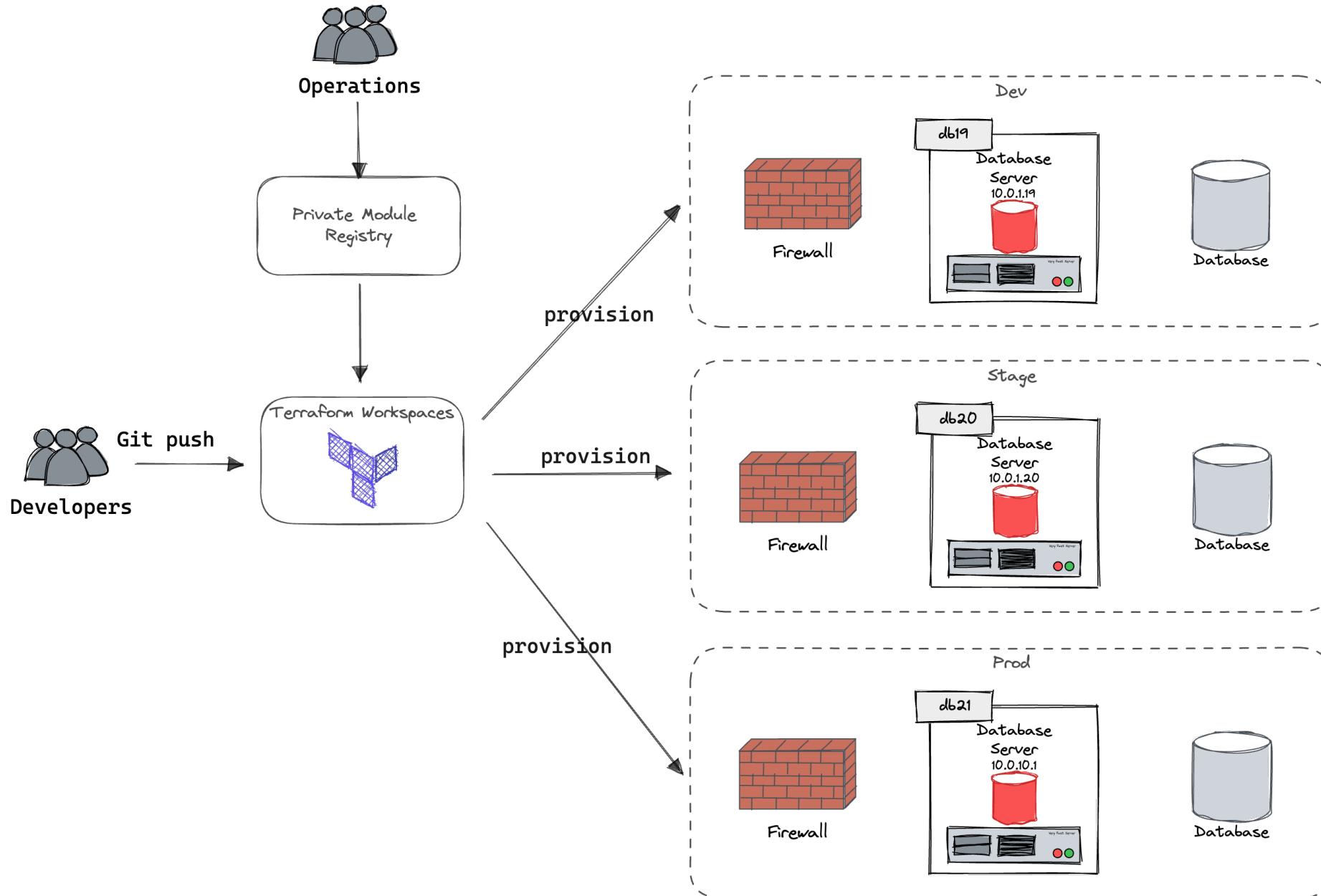
Managing apps & infrastructure

Bridging the gaps

What about infrastructure?

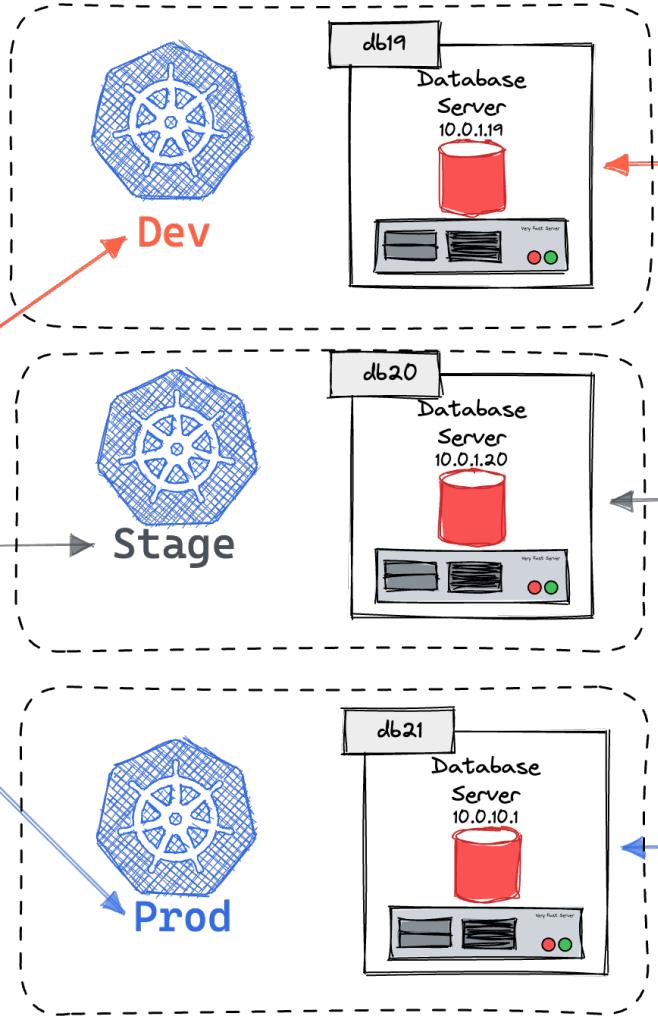
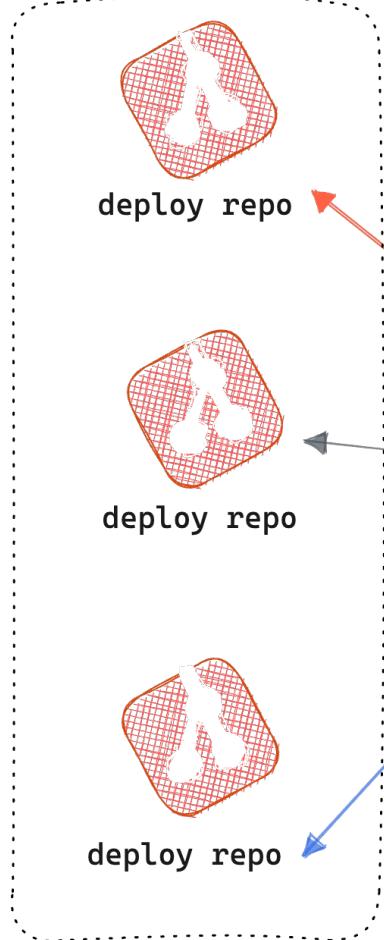


What about infrastructure? (2)

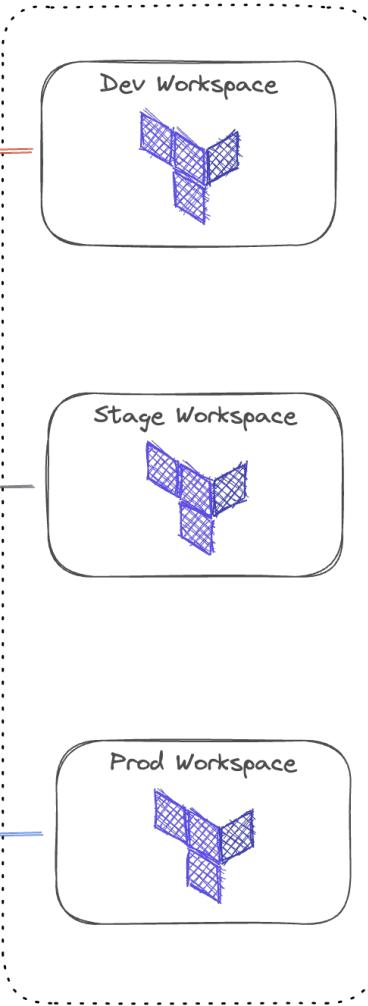


What about infrastructure? (3)

App Promotion Workflow



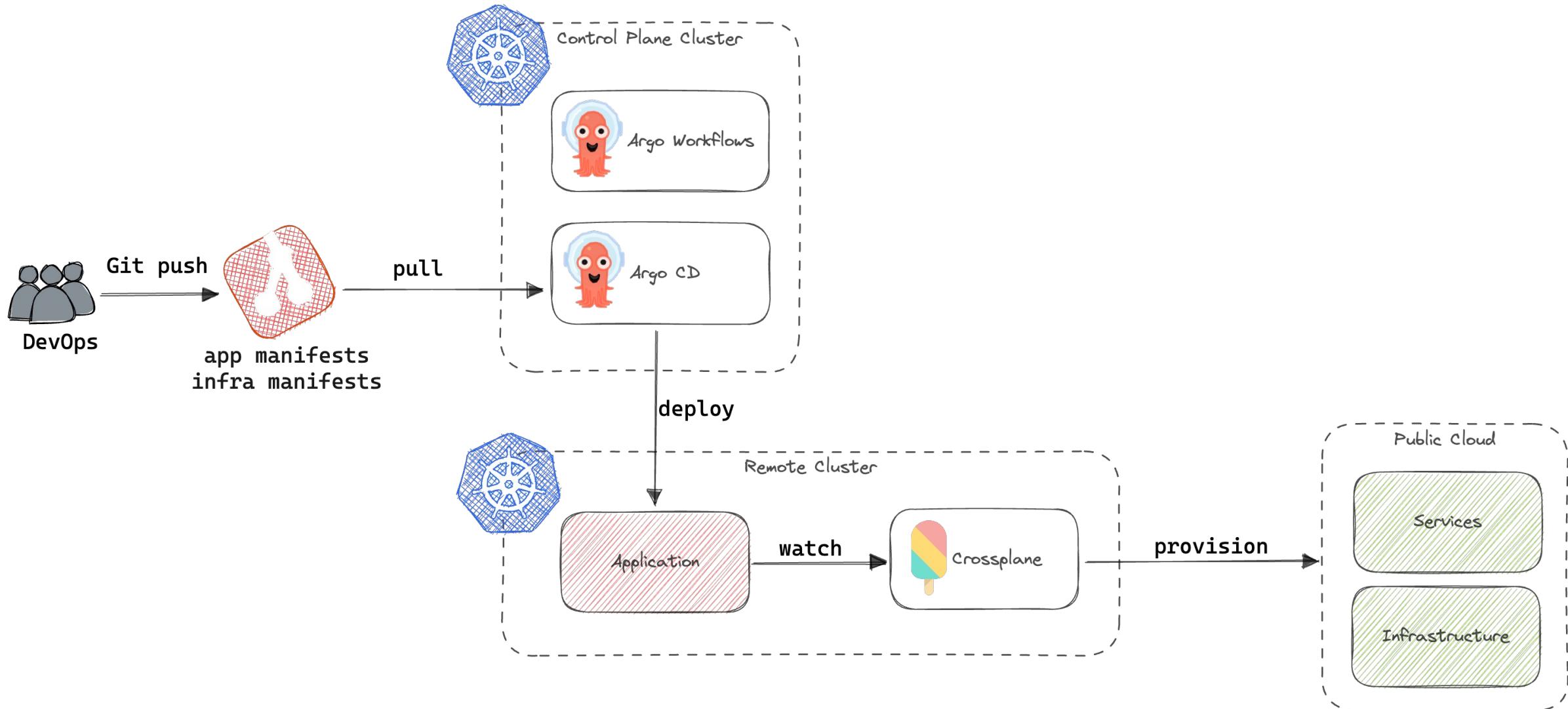
Infrastructure Provisioning Workflow



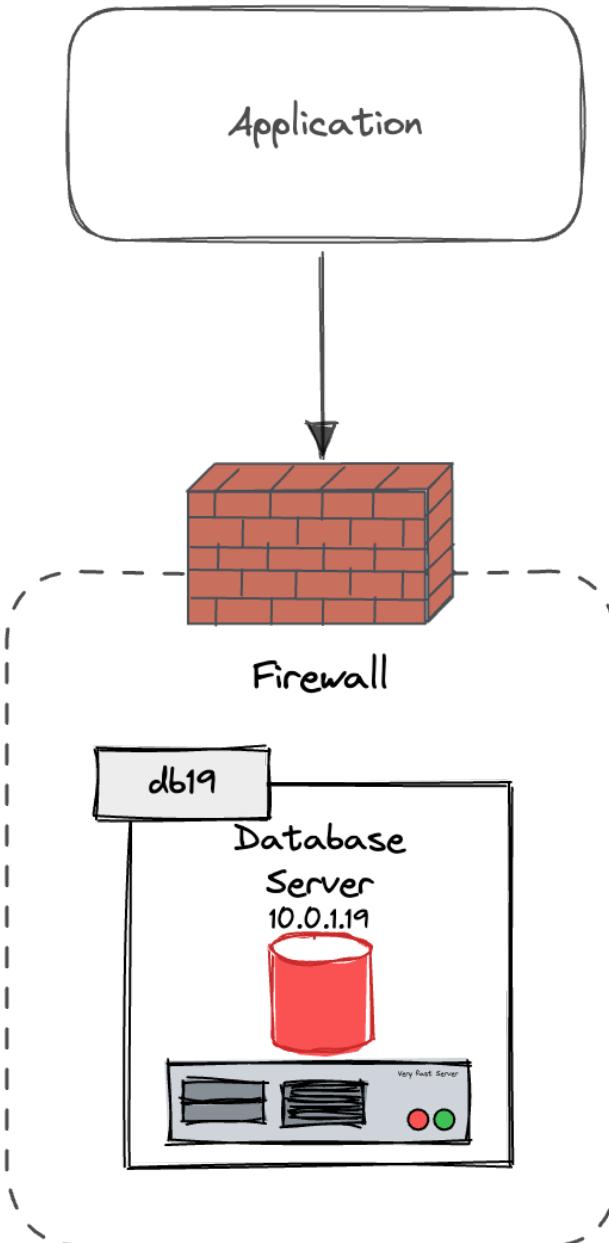
Introducing Crossplane

An open-source Kubernetes extension that transforms your
Kubernetes cluster into a **universal control plane**.

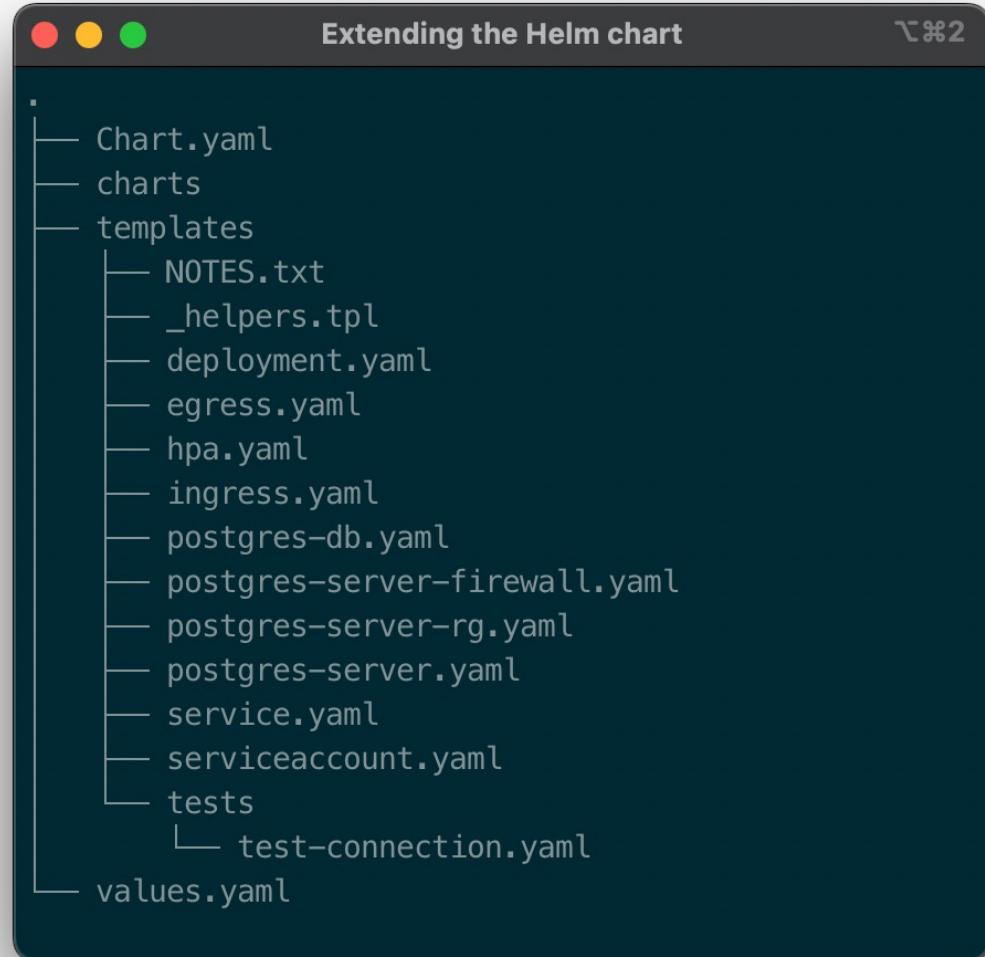
Argo CD and Crossplane



A simple example



A simple example (2)



The terminal window title is "Extending the Helm chart". The directory structure shown is:

- Chart.yaml
- charts
- templates
 - NOTES.txt
 - _helpers.tpl
 - deployment.yaml
 - egress.yaml
 - hpa.yaml
 - ingress.yaml
 - postgres-db.yaml
 - postgres-server-firewall.yaml
 - postgres-server-rg.yaml
 - postgres-server.yaml
 - service.yaml
 - serviceaccount.yaml
 - tests
 - test-connection.yaml
- values.yaml

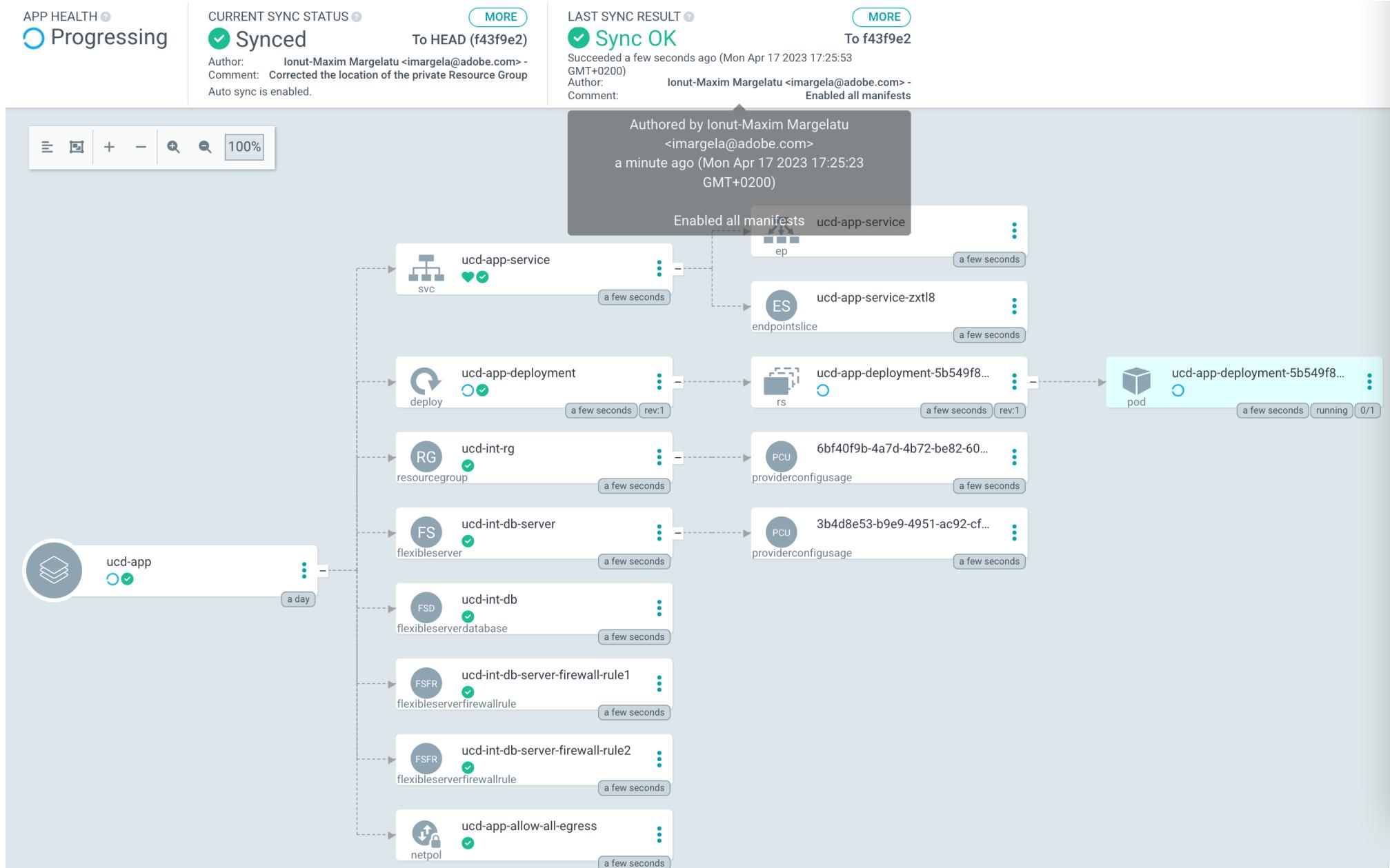
A simple example (3)

```
apiVersion: dbforpostgresql.azure.upbound.io/v1beta1
kind: FlexibleServer
metadata:
  name: {{ .Values.application.dbServerName | quote }}
spec:
  forProvider:
    administratorLogin: {{ .Values.application.dbUser | quote }}
    administratorPasswordSecretRef:
      key: password
      name: postgres-server-creds
      namespace: ucd
    location: West Europe
    resourceGroupName: ucd-int-rg
    skuName: B_Standard_B1ms
    storageMb: 32768
    version: "14"
```

Ordering via sync waves

```
1  apiVersion: dbforpostgresql.azure.upbound.io/v1beta1
2  kind: FlexibleServer
3  metadata:
4    name: {{ .Values.application.dbServerName | quote }}
5  annotations:
6    argocd.argoproj.io/sync-wave: "-90"
7  spec:
8    forProvider:
9      administratorLogin: {{ .Values.application.dbUser | quote }}
10     administratorPasswordSecretRef:
11       key: password
12       name: postgres-server-creds
13       namespace: ucd
14     location: West Europe
15     resourceGroupName: ucd-int-rg
16     skuName: B_Standard_B1ms
17     storageMb: 32768
```

Deploying application and infrastructure



Deploying application and infrastructure

The screenshot shows the Argo CD interface for managing Kubernetes resources. On the left, there's a sidebar with icons for 'APP HEALTH' (red), 'Program' (blue), and 'Manifests' (grey). The main area has tabs for 'SUMMARY' (selected) and 'EVENTS'. The 'SUMMARY' tab displays details for a 'FlexibleServer' named 'ucd-int-db-server' created on '04/17/2023 17:25:46' in the 'Synced' status.

KIND	FlexibleServer
NAME	ucd-int-db-server
NAMESPACE	
CREATED AT	04/17/2023 17:25:46 (a minute ago)
STATUS	Synced
LINKS	

Below this, there are tabs for 'LIVE MANIFEST', 'DIFF', and 'DESIRED MANIFEST'. The 'LIVE MANIFEST' tab shows the YAML configuration for the 'FlexibleServer' resource:

```
1 apiVersion: dbforpostgresql.azure.upbound.io/v1beta1
2 kind: FlexibleServer
3 metadata:
4   annotations:
5     argocd.argoproj.io/sync-wave: '-90'
6     argocd.argoproj.io/tracking-id: >-
7       ucd-app:dbforpostgresql.azure.upbound.io/FlexibleServer:ucd/ucd-int-db-server
8     crossplane.io/external-create-pending: '2023-04-17T15:25:54Z'
9     crossplane.io/external-create-succeeded: '2023-04-17T15:25:54Z'
10    crossplane.io/external-name: ucd-int-db-server
11    kubectl.kubernetes.io/last-applied-configuration: >
12      {"apiVersion":"dbforpostgresql.azure.upbound.io/v1beta1","kind":"FlexibleServer","metadata":{"annotations":{"argocd.argoproj.io/sync-wave":"-90","argocd.argoproj.io/tracking-id":>"ucd-app:dbforpostgresql.azure.upbound.io/FlexibleServer:ucd/ucd-int-db-server"}, "resourceGroup": "ucd-int-rg", "skuName": "B_Standard_B1ms", "storageMb": 32768, "version": "14"}}
13    creationTimestamp: '2023-04-17T15:25:46Z'
14    finalizers:
15      - finalizer.managedresource.crossplane.io
16    generation: 1
17    name: ucd-int-db-server
18    resourceVersion: '11042023'
```

There are buttons for 'EDIT' and 'Hide Managed Fields' (with a checked checkbox).

Deploying application and infrastructure

The screenshot shows a user interface for managing a Kubernetes pod. The top navigation bar includes tabs for 'SUMMARY', 'EVENTS' (with a red notification badge), and 'LOGS'. The 'SUMMARY' tab is active, displaying the following pod details:

KIND	Pod
NAME	ucd-app-deployment-5b549f8cf7-vzq5p
NAMESPACE	ucd
CREATED AT	04/17/2023 17:25:53 (2 minutes ago)
IMAGES	docker-experienceplatform-release.dr-uw2.adobeitc.com/imargela/ucd-service:0.2.1
STATE	Running
HEALTH	Degraded
LINKS	(empty)

Below the summary, there is a 'LIVE MANIFEST' section containing the YAML configuration for the pod:

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: '2023-04-17T15:25:53Z'
  generateName: ucd-app-deployment-5b549f8cf7-
  labels:
    app.kubernetes.io/instance: ucd-app
    app.kubernetes.io/name: ucd
    pod-template-hash: 5b549f8cf7
  name: ucd-app-deployment-5b549f8cf7-vzq5p
  namespace: ucd
  ownerReferences:
    - apiVersion: apps/v1
      kind: Deployment
      name: ucd-app-deployment
      uid: 5b549f8cf7-1234-4567-8901-234567890123
```

On the right side of the manifest, there are buttons for 'Hide Managed Fields' (with a checked checkbox) and 'EDIT'.

Deploying application and infrastructure



KubeCon
Europe 2023



CloudNativeCon
Europe 2023

The screenshot shows a user interface for monitoring application health and logs. On the left, there's a sidebar with icons for 'APP HEALTH' (green), 'Program' (blue), and a stack icon. The main area has a toolbar with various buttons (refresh, download, etc.) and a search bar labeled 'Filter string'. The central part displays a stack trace with line numbers and file paths. The trace starts with a 'Caused by' section and ends with a timestamped error message.

```
Page 1 (Lines 1 to 100)

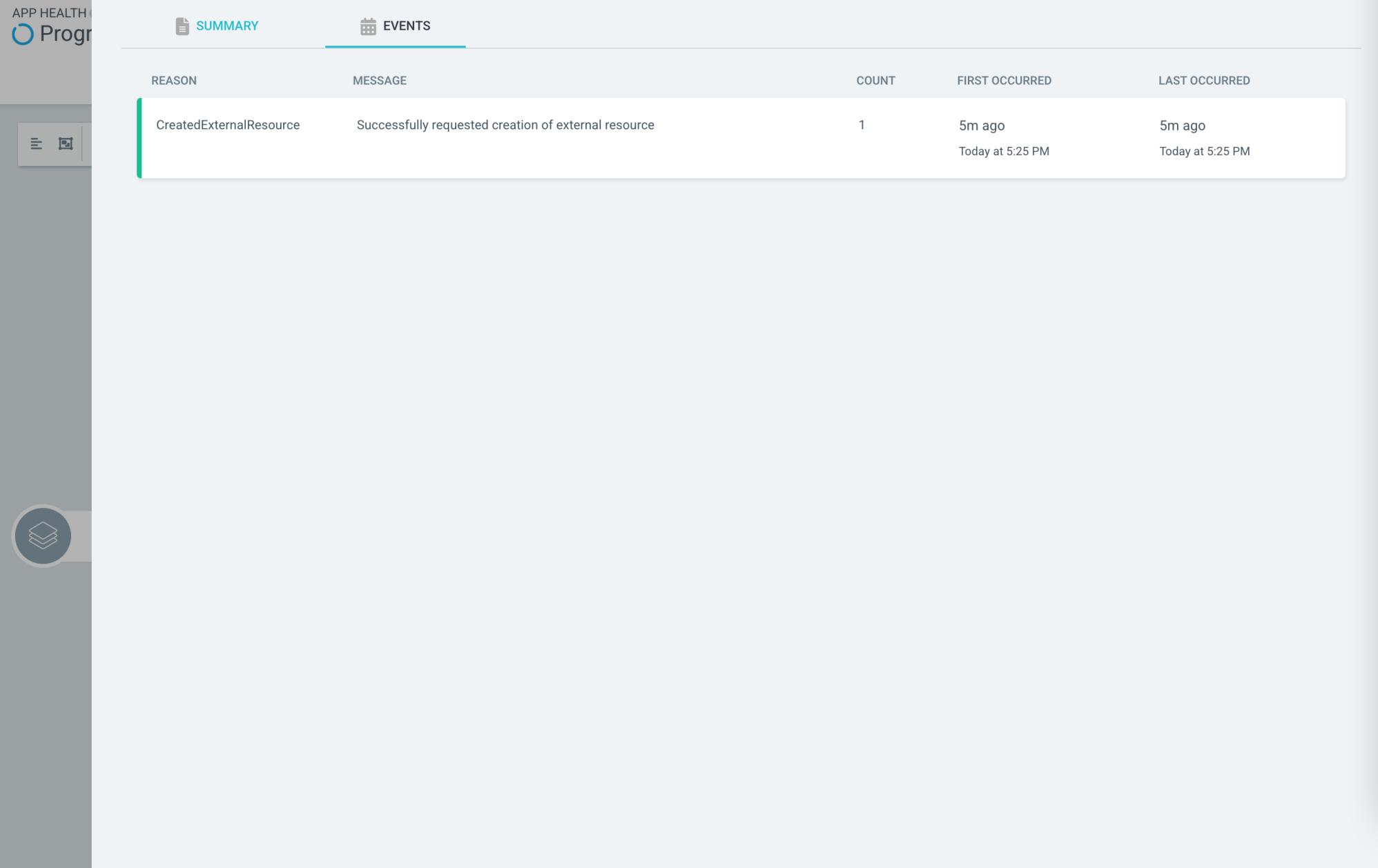
45 at org.hibernate.tool.schema.spi.SchemaManagementToolCoordinator.performDatabaseAction(SchemaManagementToolCoordinator.java:157)
44 at org.hibernate.tool.schema.spi.SchemaManagementToolCoordinator.process(SchemaManagementToolCoordinator.java:85) ~[hibernate-core-5.6.15.Final.jar:5.6.15.Final]
43 at org.hibernate.internal.SessionFactoryImpl.<init>(SessionFactoryImpl.java:335) ~[hibernate-core-5.6.15.Final.jar:5.6.15.Final]
42 at org.hibernate.boot.internal.SessionFactoryBuilderImpl.build(SessionFactoryBuilderImpl.java:471) ~[hibernate-core-5.6.15.Final.jar:5.6.15.Final]
41 at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.build(EntityManagerFactoryBuilderImpl.java:1498) ~[hibernate-core-5.6.15.Final.jar:5.6.15.Final]
40 at org.springframework.orm.jpa.vendor.SpringHibernateJpaPersistenceProvider.createContainerEntityManagerFactory(SpringHibernateJpaPersistenceProvider.java:52) ~[spring-orm-5.3.26.jar:5.3.26]
39 at org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean.createNativeEntityManagerFactory(LocalContainerEntityManagerFactoryBean.java:88) ~[spring-orm-5.3.26.jar:5.3.26]
38 at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.buildNativeEntityManagerFactory(AbstractEntityManagerFactoryBean.java:172) ~[spring-orm-5.3.26.jar:5.3.26]
37 at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.afterPropertiesSet(AbstractEntityManagerFactoryBean.java:396) ~[spring-orm-5.3.26.jar:5.3.26]
36 at org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean.afterPropertiesSet(LocalContainerEntityManagerFactoryBean.java:105) ~[spring-orm-5.3.26.jar:5.3.26]
35 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:180) ~[spring-beans-5.3.26.jar:5.3.26]
34 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:156) ~[spring-beans-5.3.26.jar:5.3.26]
33 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:542) ~[spring-beans-5.3.26.jar:5.3.26]
32 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:482) ~[spring-beans-5.3.26.jar:5.3.26]
31 at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:335) ~[spring-beans-5.3.26.jar:5.3.26]
30 at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:234) ~[spring-beans-5.3.26.jar:5.3.26]
29 at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:333) ~[spring-beans-5.3.26.jar:5.3.26]
28 at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:208) ~[spring-beans-5.3.26.jar:5.3.26]
27 at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:1156) ~[spring-context-5.3.26.jar:5.3.26]
26 at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:867) ~[spring-context-5.3.26.jar:5.3.26]
25 at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:583) ~[spring-context-5.3.26.jar:5.3.26]
24 at org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refresh(ServletWebServerApplicationContext.java:145) ~[spring-boot-2.7.10.jar:2.7.10]
23 at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:731) ~[spring-boot-2.7.10.jar:2.7.10]
22 at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:408) ~[spring-boot-2.7.10.jar:2.7.10]
21 at org.springframework.boot.SpringApplication.run(SpringApplication.java:307) ~[spring-boot-2.7.10.jar:2.7.10]
20 at org.springframework.boot.SpringApplication.run(SpringApplication.java:1303) ~[spring-boot-2.7.10.jar:2.7.10]
19 at org.springframework.boot.SpringApplication.run(SpringApplication.java:1292) ~[spring-boot-2.7.10.jar:2.7.10]
18 at com.adobe.ucd.service.UcdApplicationKt.main(UcdApplication.kt:13) ~[classes:/na]
17 Caused by: java.net.SocketTimeoutException: connect timed out
16 at java.base/java.net.PlainSocketImpl.socketConnect(Native Method) ~[na:na]
15 at java.base/java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:412) ~[na:na]
14 at java.base/java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:255) ~[na:na]
13 at java.base/java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:237) ~[na:na]
12 at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392) ~[na:na]
11 at java.base/java.net.Socket.connect(Socket.java:608) ~[na:na]
10 at org.postgresql.core.PGStream.createSocket(PGStream.java:241) ~[postgresql-42.3.8.jar:42.3.8]
9 at org.postgresql.core.PGStream.<init>(PGStream.java:98) ~[postgresql-42.3.8.jar:42.3.8]
8 at org.postgresql.core.v3.ConnectionFactoryImpl.tryConnect(ConnectionFactoryImpl.java:109) ~[postgresql-42.3.8.jar:42.3.8]
7 at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:235) ~[postgresql-42.3.8.jar:42.3.8]
6 ... 50 common frames omitted
5
4 2023-04-17 15:28:25.394  WARN 1 --- [           main] o.h.engine.jdbc.spi.SqlExceptionHelper : SQL Error: 0, SQLState: 08001
3 2023-04-17 15:28:25.395 ERROR 1 --- [           main] o.h.engine.jdbc.spi.SqlExceptionHelper : The connection attempt failed.
2 2023-04-17 15:28:25.452 ERROR 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Failed to initialize JPA EntityManager.
```

Deploying application and infrastructure

The screenshot shows a web-based application monitoring tool. On the left, a sidebar displays "APP HEALTH" and "Program". The main area has tabs for "SUMMARY" and "EVENTS" (with 10 notifications). A table lists events with columns: REASON, MESSAGE, COUNT, FIRST OCCURRED, and LAST OCCURRED. One event is highlighted with a red border.

REASON	MESSAGE	COUNT	FIRST OCCURRED	LAST OCCURRED
CannotResolveResourceRef...	cannot resolve references: mg.Spec.ForProvider.ServerID: referenced field was empty (referenced resource may not yet be ready)	10	4m ago Today at 5:25 PM	5s ago Today at 5:29 PM

Deploying application and infrastructure



The screenshot shows a user interface for monitoring application and infrastructure health. On the left, there's a sidebar with icons for 'APP HEALTH' (red), 'Program' (blue), and 'Logs' (green). The main area has tabs for 'SUMMARY' and 'EVENTS', with 'EVENTS' being active. A table lists an event:

REASON	MESSAGE	COUNT	FIRST OCCURRED	LAST OCCURRED
CreatedExternalResource	Successfully requested creation of external resource	1	5m ago Today at 5:25 PM	5m ago Today at 5:25 PM

Crossplane resource state in Argo CD

By default, Argo uses the state of the Crossplane resource,
NOT the actual state of the provisioned infrastructure.

Custom resource health in Argo CD

```
data:
  application.resourceTrackingMethod: annotation
  resource.customizations: |-
    azure.upbound.io/ResourceGroup:
      health.lua: |
        health_status = {
          status = "Progressing",
          message = "Provisioning Resource Group..."
        }

        if obj.status == nil or obj.status.conditions == nil then
          return hs
        end

        for i, condition in ipairs(obj.status.conditions) do
          if condition.type == "Ready" then
            if condition.status == "True" then
              health_status.status = "Healthy"
              health_status.message = "The Resource Group is up-to-date."
              return health_status
            end
          end

          if condition.type == "LastAsyncOperation" then
            if condition.status == "False" then
              health_status.status = "Degraded"
              health_status.message = condition.message
              return health_status
            end
          end

          if condition.type == "Synced" then
            if condition.status == "False" then
              health_status.status = "Degraded"
              health_status.message = condition.message
              return health_status
            end
          end
        end

        return health_status
      
```

Synchronized deployment

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

APP HEALTH Missing

CURRENT SYNC STATUS OutOfSync From HEAD (00293cd)

Author: Ionut-Maxim Margelatu <imargela@adobe.com> - Comment: Consistent naming for resources Auto sync is enabled.

LAST SYNC RESULT Syncing

Running a few seconds ago (Sun Apr 16 2023 19:58:37 GMT+0200)
waiting for healthy state of dbforpostgresql.azure.upbound.io/FlexibleServer/ucd-int-db-server

MORE MORE

☰ + - 🔍 🔍 100%

The diagram illustrates the synchronized deployment of the **ucd-app** application across multiple cloud resources. The application itself is shown as a stack icon with a yellow ghost icon. It is connected via dashed lines to several other resources, each represented by a specific icon and labeled with its name and type:

- ucd-app-service**: svc (Service icon)
- ucd-app-deployment**: deploy (Deployment icon)
- ucd-int-rg**: resourcegroup (Resource Group icon)
- ucd-int-db-server**: flexibleserver (Flexible Server icon)
- ucd-int-db**: flexibleserverdatabase (Flexible Server Database icon)
- ucd-int-db-server-firewall-rule1**: fsfr (Flexible Server Firewall Rule icon)
- ucd-int-db-server-firewall-rule2**: fsfr (Flexible Server Firewall Rule icon)
- ucd-app-allow-all-egress**: netpol (Network Policy icon)
- providerconfigusage**: pcu (Provider Configuration Usage icon)
- 196c4e8c-a87d-41cf-8d64-52...**: pcu (Provider Configuration Usage icon)
- 7073bd96-b3b3-4fdc-a104-6f...**: pcu (Provider Configuration Usage icon)

Each resource is accompanied by a yellow ghost icon, indicating its status. A timestamp of "5 minutes" is shown between the main application and each of the connected resources, suggesting a regular update or synchronization interval.

Synchronized deployment

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

APP HEALTH Missing

CURRENT SYNC STATUS OutOfSync From HEAD (00293cd)

Author: Ionut-Maxim Margelatu <imargela@adobe.com> - Comment: Consistent naming for resources Auto sync is enabled.

LAST SYNC RESULT Syncing

Running a few seconds ago (Sun Apr 16 2023 20:00:16 GMT+0200)
waiting for healthy state of dbforpostgresql.azure.upbound.io/FlexibleServerFirewallRule/ucd-int...

☰ + 🔍 🔍 100%

ucd-app

ucd-app-service

ucd-app-deployment

ucd-int-rg

ucd-int-db-server

ucd-int-db

ucd-int-db-server-firewall-rule1

ucd-int-db-server-firewall-rule2

ucd-app-allow-all-egress

providerconfigusage

providerconfigusage

providerconfigusage

providerconfigusage

providerconfigusage

providerconfigusage

Diagram illustrating the synchronized deployment of the ucd-app application across various cloud resources. The application is deployed via a ucd-app-deployment (status: OutOfSync) which depends on a ucd-app-service (status: Missing). This deployment is associated with a resource group (ucd-int-rg) and a flexible server database (ucd-int-db). The database is connected to two firewall rules (ucd-int-db-server-firewall-rule1 and ucd-int-db-server-firewall-rule2), both of which depend on providerconfigusage resources. Finally, a netpol rule (ucd-app-allow-all-egress) is applied to the application service. All components show a 6-minute delay between deployment and synchronization, except for the firewall rules which show a 2-minute delay.

Synchronized deployment

Synchronized deployment

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

APP HEALTH Progressing

CURRENT SYNC STATUS Synced To HEAD (00293cd)
Author: Ionut-Maxim Margelatu <imargela@adobe.com> -
Comment: Consistent naming for resources
Auto sync is enabled.

LAST SYNC RESULT Sync OK To 00293cd
Succeeded a minute ago (Sun Apr 16 2023 20:01:16 GMT+0200)
Author: Ionut-Maxim Margelatu <imargela@adobe.com> -
Comment: Consistent naming for resources

100%

ucd-app-service ep a minute

ucd-app-service-b7sqx endpointslice a minute

ucd-app-deployment deploy a minute rev:1

ucd-app-deployment-5b549f8... rs a minute rev:1

ucd-app-deployment-5b549f8... pod a minute running 0/1

ucd-int-rg resourcegroup 8 minutes

196c4e8c-a87d-41cf-8d64-52... providerconfigusage 8 minutes

7073bd96-b3b3-4fdc-a104-6f... providerconfigusage 8 minutes

75b83bc3-7578-48da-87b5-7... providerconfigusage a minute

113af6ae-eec1-4398-a4c0-8b... providerconfigusage 3 minutes

2a79cb65-f079-4f52-b360-bf... providerconfigusage 3 minutes

ucd-int-db-server flexibleserver 8 minutes

ucd-int-db flexibleserverdatabase a minute

ucd-int-db-server-firewall-rule1 flexibleserverfirewallrule 3 minutes

ucd-int-db-server-firewall-rule2 flexibleserverfirewallrule 3 minutes

ucd-app-allow-all-egress netpol a minute

Synchronized deployment

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

APP HEALTH Healthy CURRENT SYNC STATUS Synced To HEAD (00293cd) LAST SYNC RESULT Sync OK To 00293cd

Author: Ionut-Maxim Margelatu <imargela@adobe.com> - Comment: Consistent naming for resources

Author: Ionut-Maxim Margelatu <imargela@adobe.com> - Comment: Consistent naming for resources

100%

ucd-app

ucd-app-service

ucd-app-deployment

ucd-int-rg

ucd-int-db-server

ucd-int-db

ucd-int-db-server-firewall-rule1

ucd-int-db-server-firewall-rule2

ucd-app-allow-all-egress

ep

ES

deploy

RG

FS

FSD

FSFR

FSFR

netpol

ucd-app-service-b7sqx

ucd-app-deployment-5b549f8...

196c4e8c-a87d-41cf-8d64-52...

7073bd96-b3b3-4fdc-a104-6f...

75b83bc3-7578-48da-87b5-7...

113af6ae-eec1-4398-a4c0-8b...

2a79cb65-f079-4f52-b360-bf...

pcu

pcu

pcu

pcu

pcu

pcu

3 minutes

3 minutes

3 minutes

3 minutes

3 minutes

11 minutes

10 minutes

4 minutes

6 minutes

6 minutes

3 minutes

3 minutes

running 1/1

Infrastructure up & running

```
~ ➔ kubectl get managed
NAME                                READY   SYNCED   EXTERNAL-NAME   AGE
resourcegroup.azure.upbound.io/ucd-int-rg   True    True     ucd-int-rg      22m

NAME                                         READY   SYNCED   EXTERNAL-NAME   AGE
flexibleserverfirewallrule.dbforpostgresql.azure.upbound.io/ucd-int-db-server-firewall-rule1   True    True     ucd-int-db-server-firewall-rule1  22m
flexibleserverfirewallrule.dbforpostgresql.azure.upbound.io/ucd-int-db-server-firewall-rule2   True    True     ucd-int-db-server-firewall-rule2  22m

NAME                                READY   SYNCED   EXTERNAL-NAME   AGE
flexibleserver.dbforpostgresql.azure.upbound.io/ucd-int-db-server   True    True     ucd-int-db-server      22m

NAME                                READY   SYNCED   EXTERNAL-NAME   AGE
flexibleserverdatabase.dbforpostgresql.azure.upbound.io/ucd-int-db   True    True     ucd                      22m
```

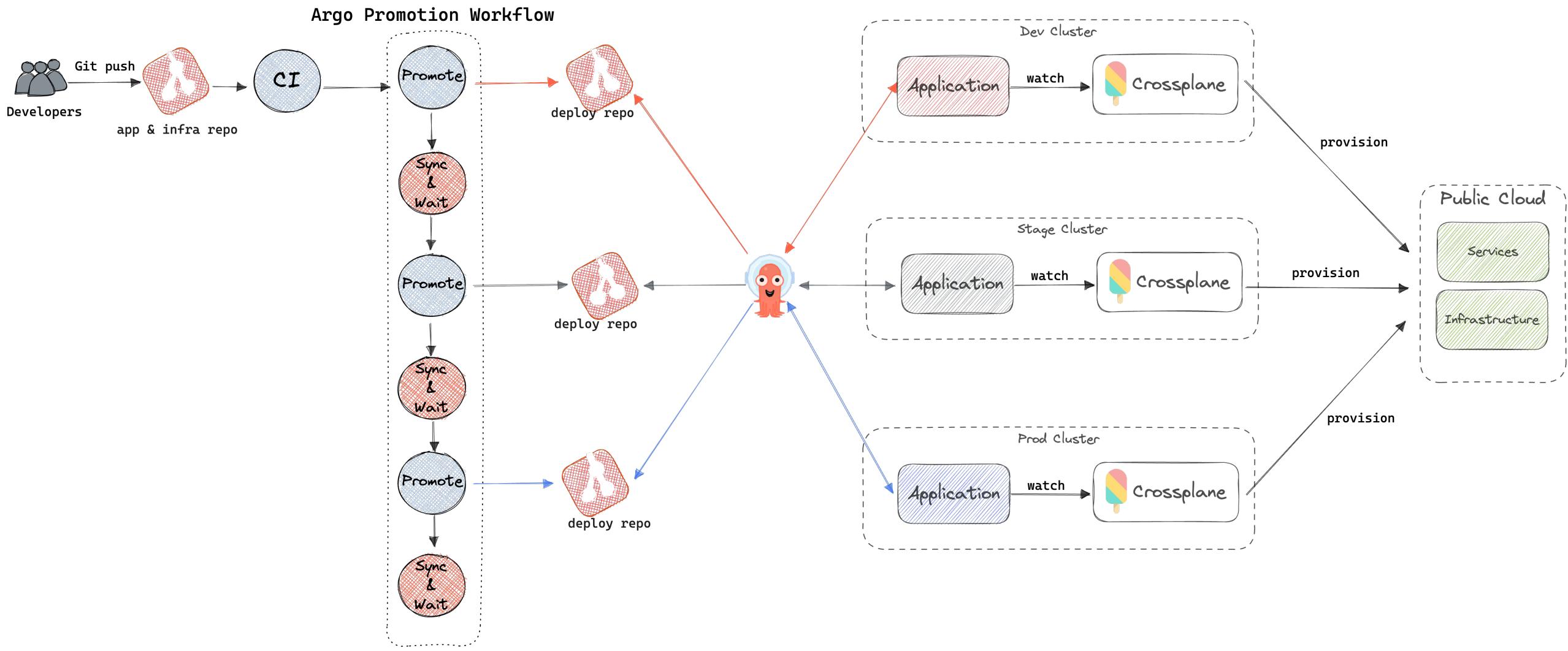
Application running

```
~ ➔ http POST localhost:7070/events/ action=write resourceType=Schema userEmail=imargela@adobe.com date="2023-04-16T12:01:06"  
HTTP/1.1 201  
Connection: keep-alive  
Content-Type: application/json  
Date: Sun, 16 Apr 2023 18:59:04 GMT  
Keep-Alive: timeout=60  
Transfer-Encoding: chunked  
  
{  
  "action": "write",  
  "date": "2023-04-16T12:01:06.000+00:00",  
  "id": "fc86a439-1059-499b-a596-1238ddaec03d",  
  "resourceType": "Schema",  
  "userEmail": "imargela@adobe.com"  
}
```

Application running

```
~ ➔ http localhost:7070/events/  
HTTP/1.1 200  
Connection: keep-alive  
Content-Type: application/json  
Date: Sun, 16 Apr 2023 18:59:16 GMT  
Keep-Alive: timeout=60  
Transfer-Encoding: chunked  
  
[  
  {  
    "action": "write",  
    "date": "2023-04-16T12:01:06.000+00:00",  
    "id": "fc86a439-1059-499b-a596-1238ddaec03d",  
    "resourceType": "Schema",  
    "userEmail": "imargela@adobe.com"  
  }  
]
```

Complete CD for application and infra



Session QR Codes will be
sent via email before the event

Please scan the QR Code above
to leave feedback on this session