
KubeCon
CloudNativeCon
Europe 2023
TiKV

Tips from the Trenches GitOps at Adobe

Larisa-Andreea Dănăilă
Ionuț-Maxim Mărgelatu

The story so far

2020

Company-wide migration to Kubernetes

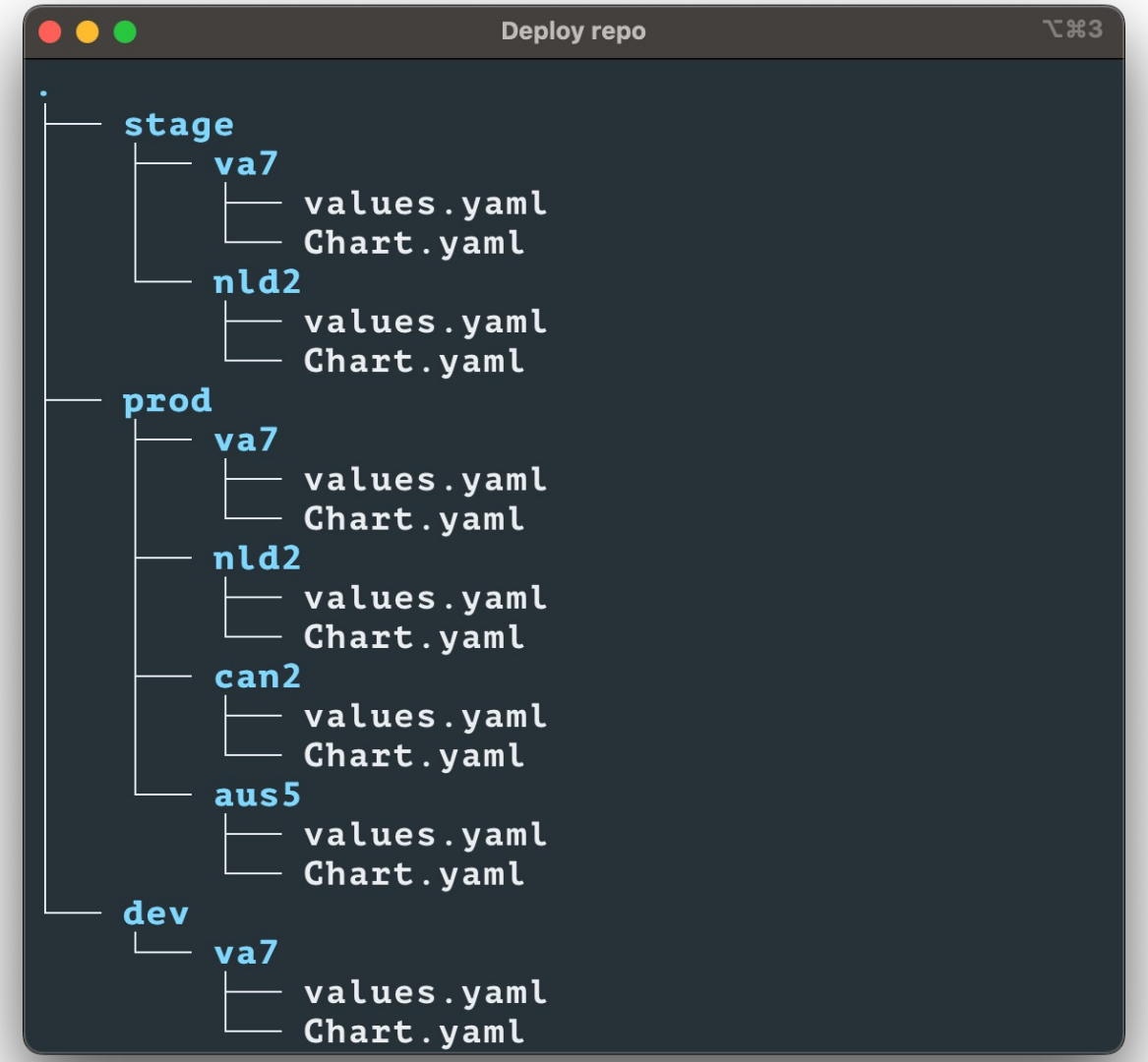
2022

Deployment platform on top of the Argo project

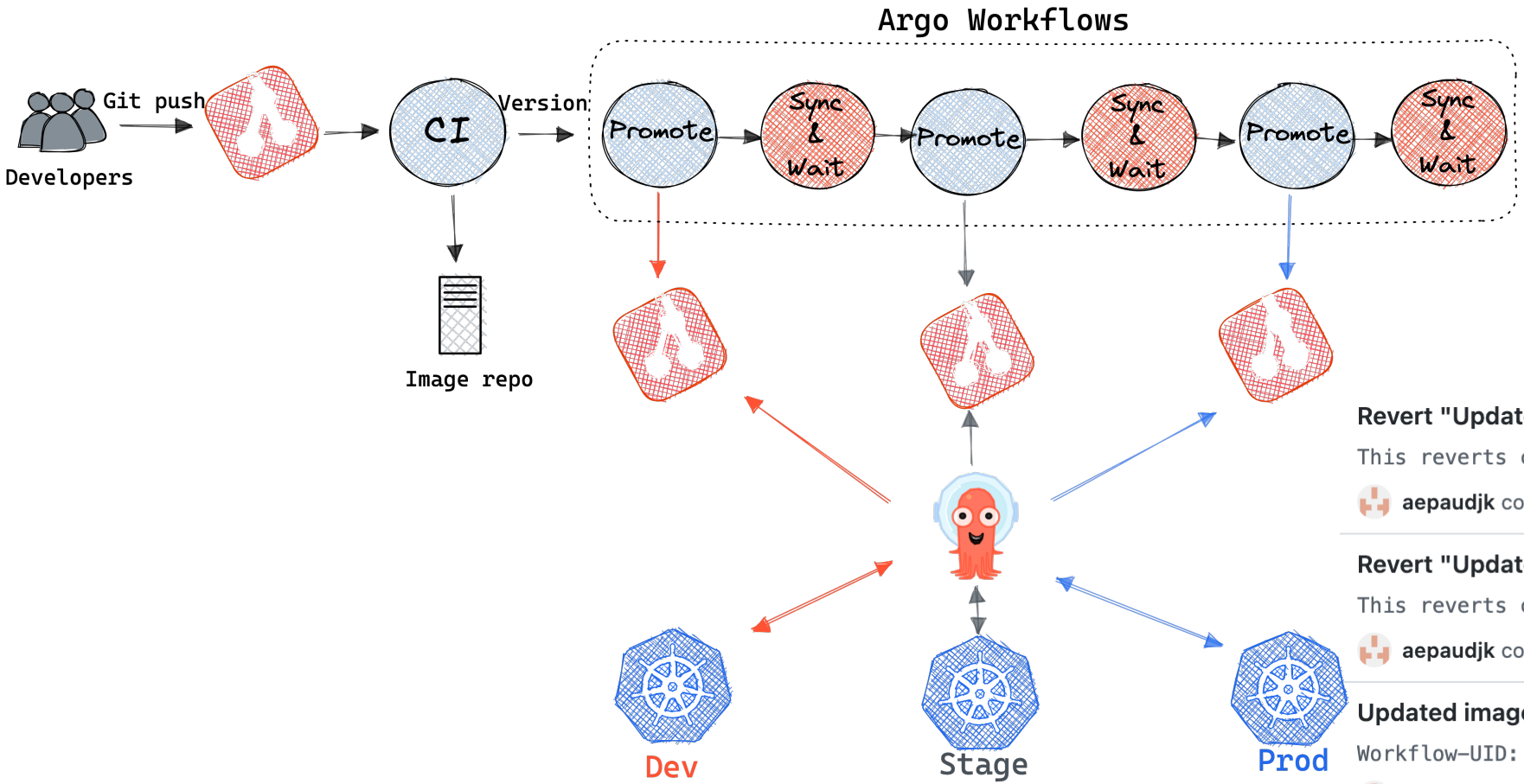
2023

Streamlining DevOps practices with Argo

Trunk based development



CICD and automated rollbacks



Revert "Updated image tag to 0.4.277 for Dev va7" ···

This reverts commit b2a572a.

aepaudjk committed yesterday ✓

Revert "Updated image tag to 0.4.277 for Stage va7" ···

This reverts commit 1c36915.

aepaudjk committed yesterday

Updated image tag to 0.4.277 for Stage va7 ···

Workflow-UID: 5817b12b-7126-4c52-9cf9-b66bca426aef

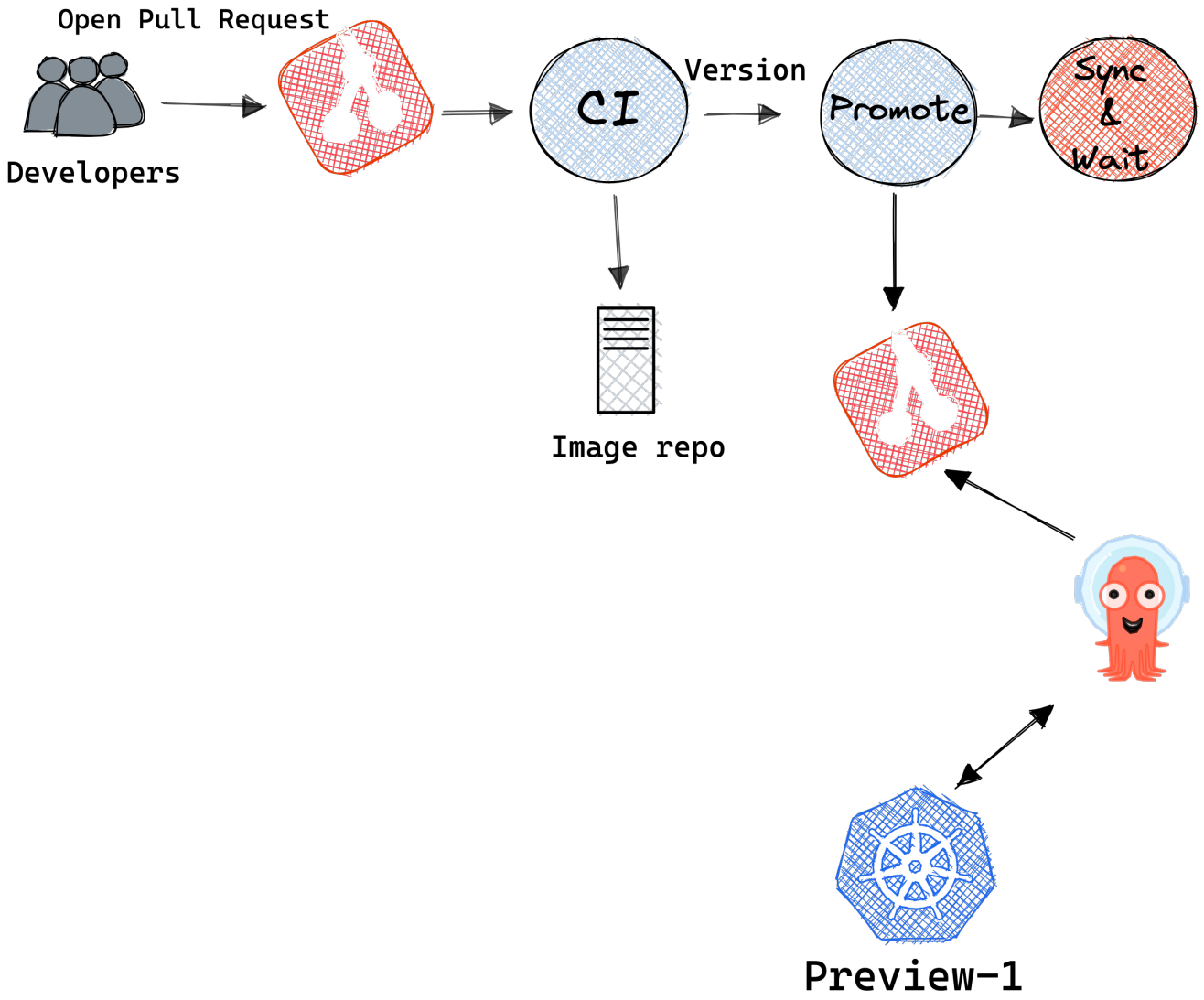
aepaudjk committed yesterday ✓

Updated image tag to 0.4.277 for Dev va7 ···

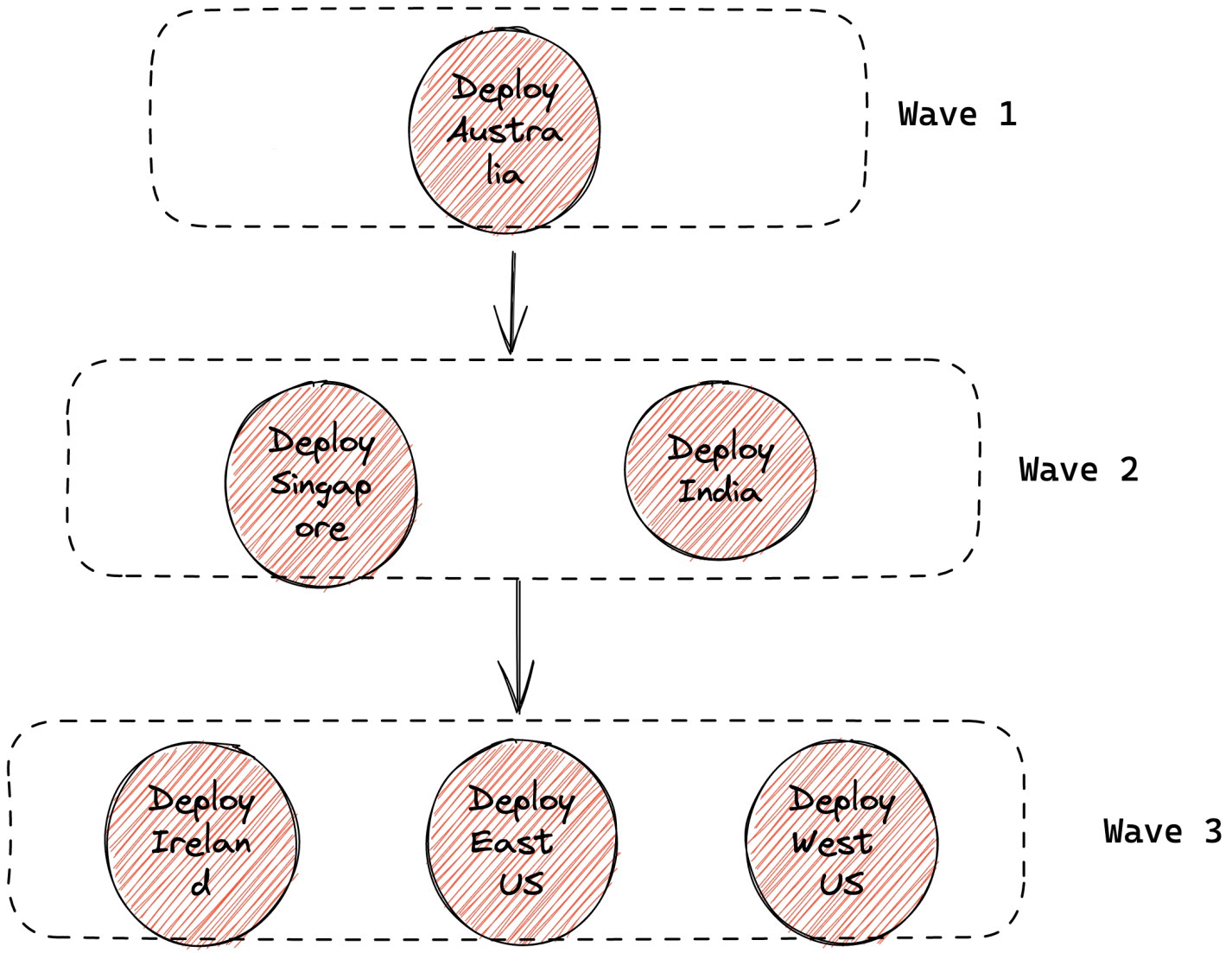
Workflow-UID: 5817b12b-7126-4c52-9cf9-b66bca426aef

aepaudjk committed yesterday ✓

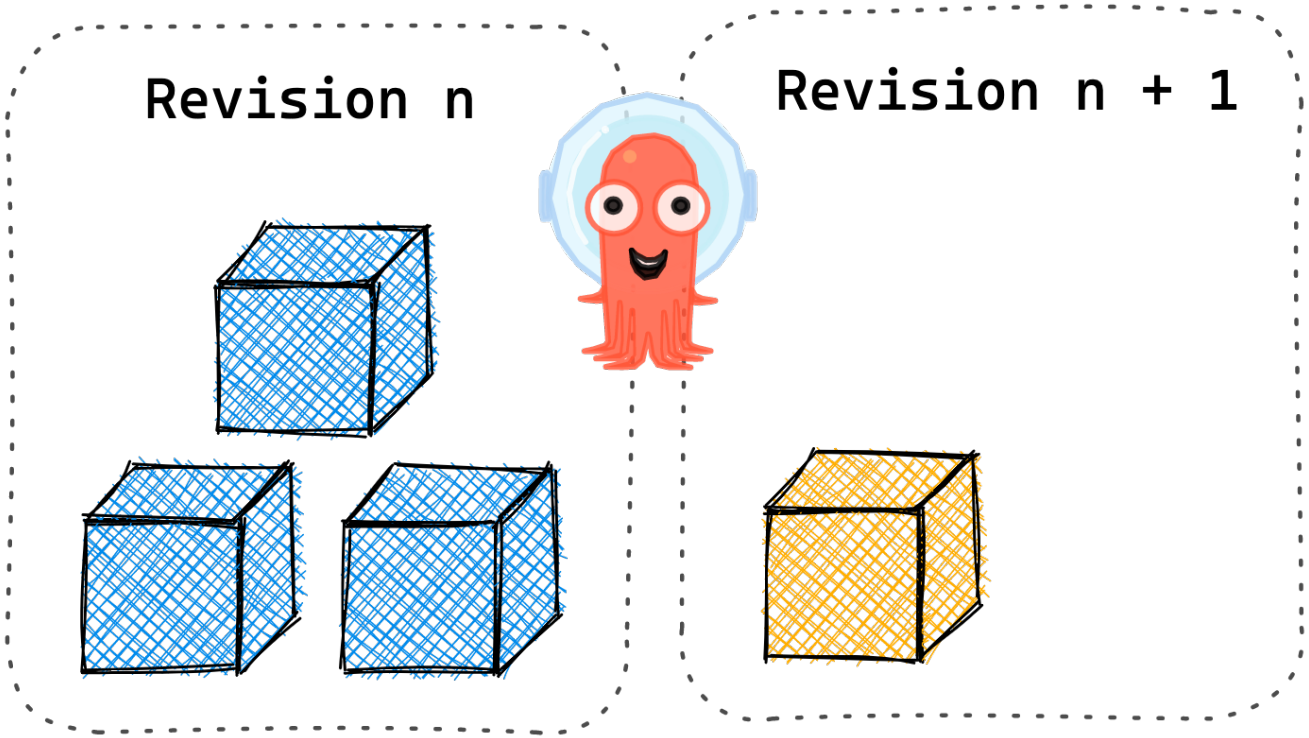
Advanced practices (1)



Advanced practices (2)



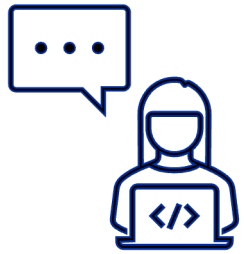
Advanced practices (3)



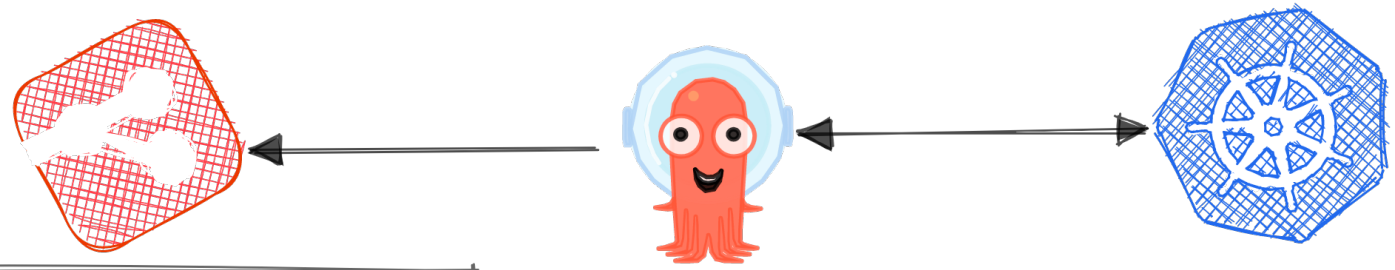
Challenges and obstacles

With great power comes great ***creativity***

Synchronization phases



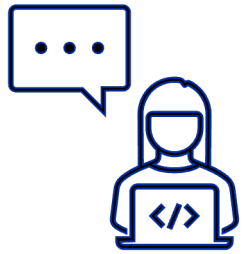
“I want to deploy my applications in a particular order.”



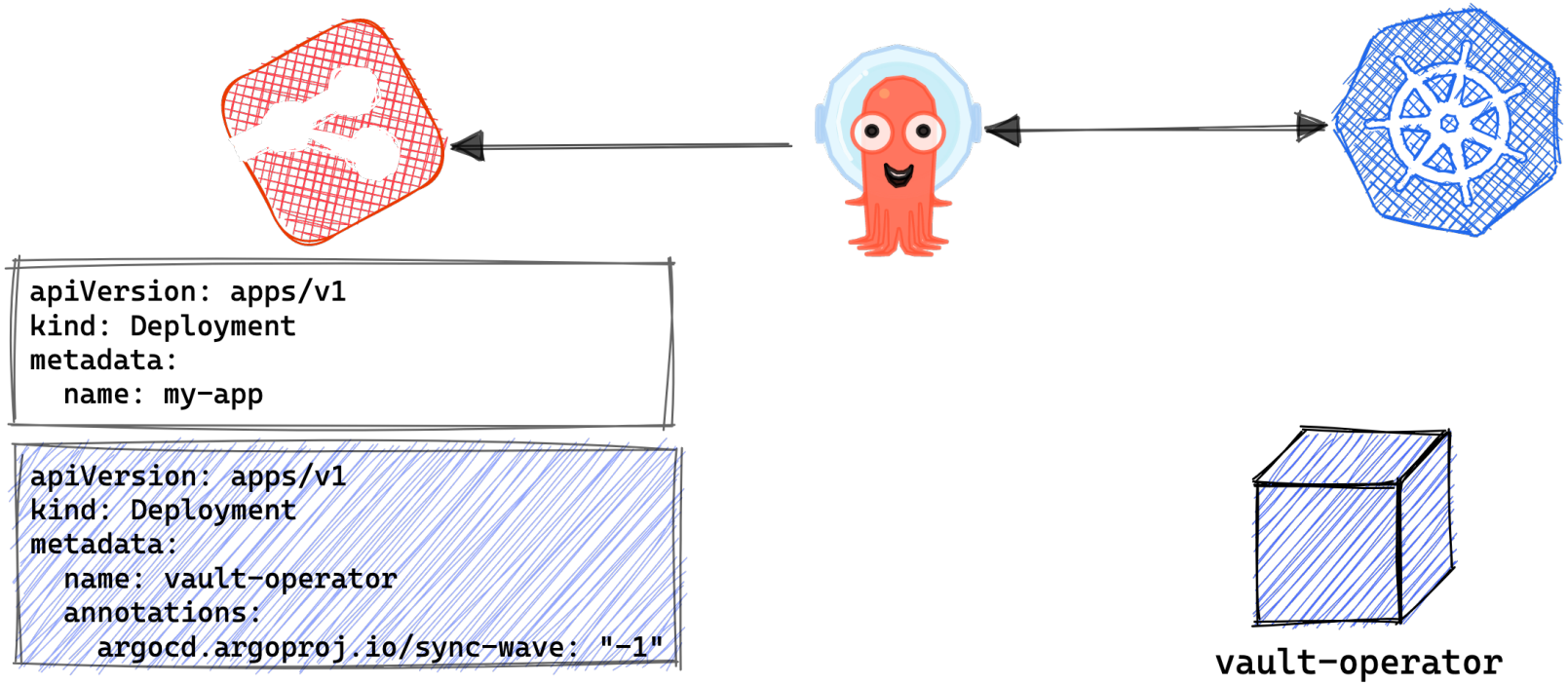
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vault-operator
  annotations:
    argocd.argoproj.io/sync-wave: "-1"
```

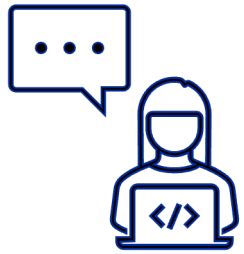
Synchronization phases



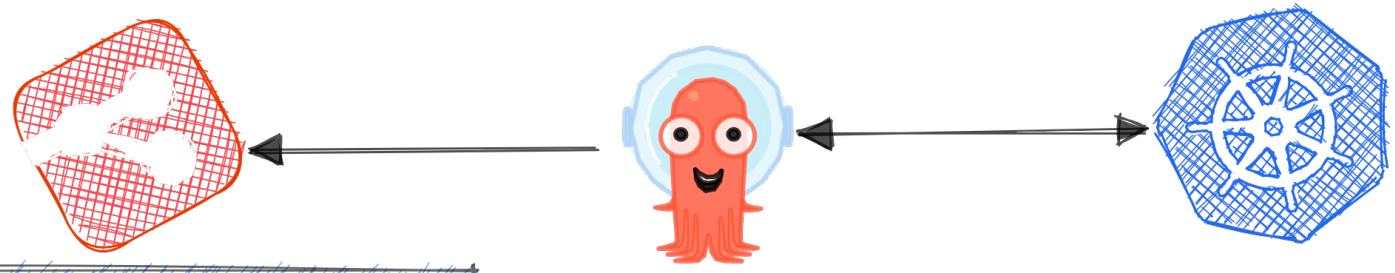
“I want to deploy my applications in a particular order.”



Synchronization phases

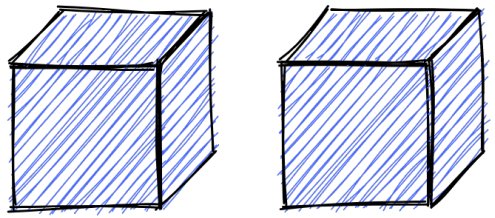


“I want to deploy my applications in a particular order.”



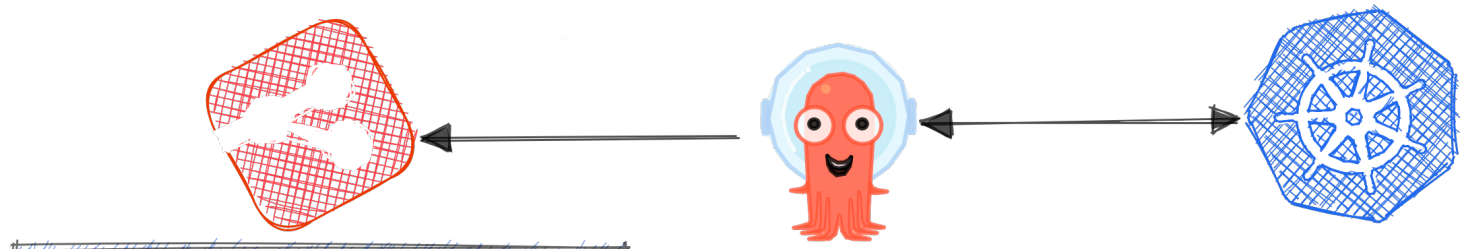
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vault-operator
  annotations:
    argocd.argoproj.io/sync-wave: "-1"
```



vault-operator my-app

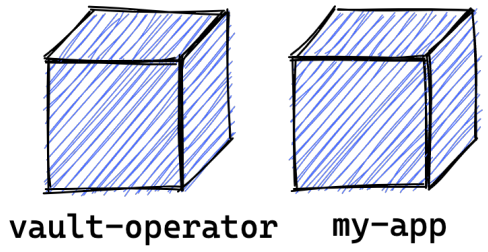
Hooks are not desired state



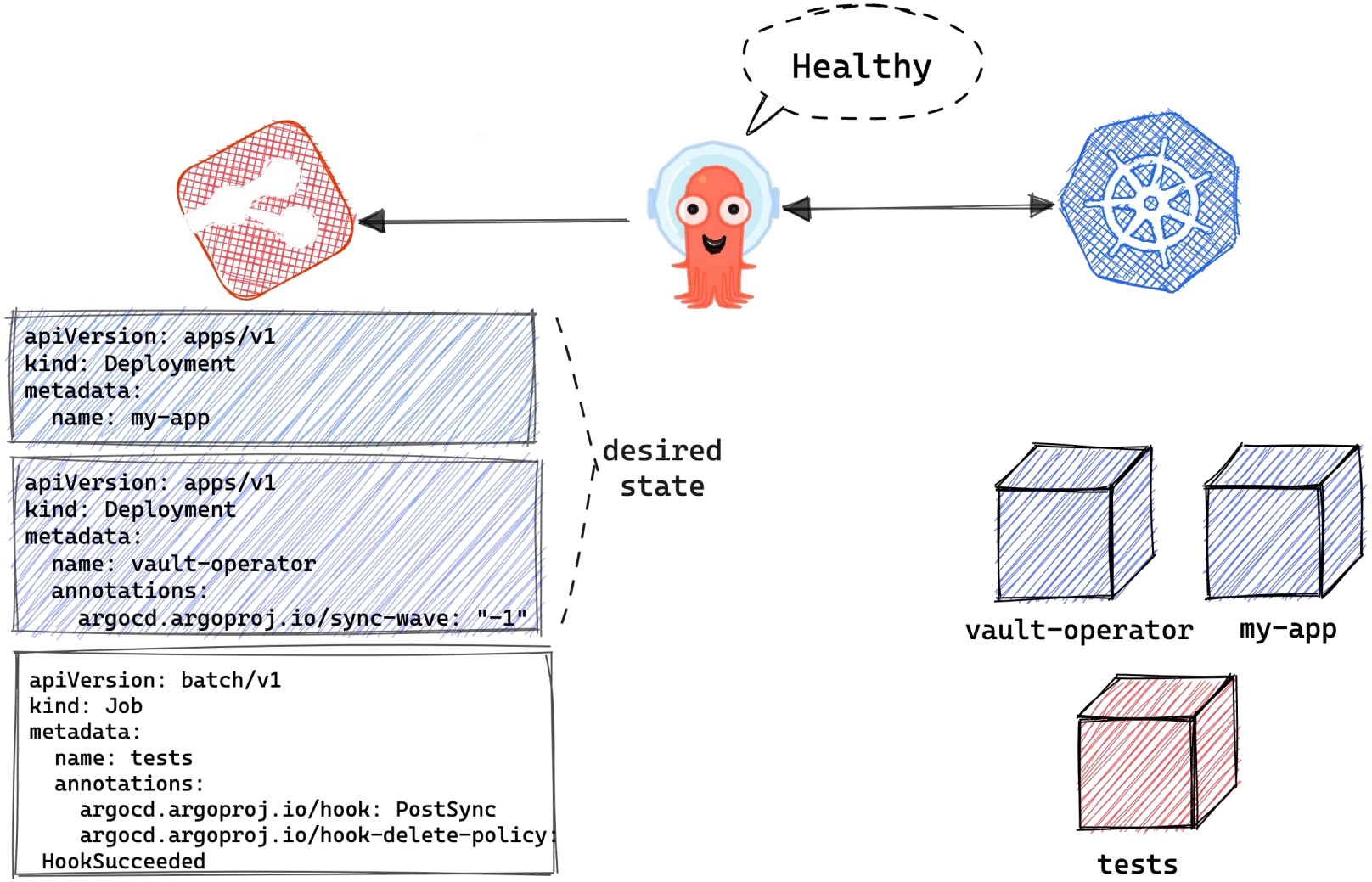
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vault-operator
  annotations:
    argocd.argoproj.io/sync-wave: "-1"
```

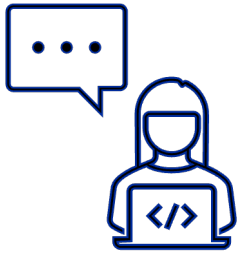
```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: tests
  annotations:
    argocd.argoproj.io/hook: PostSync
    argocd.argoproj.io/hook-delete-policy:
 HookSucceeded
```



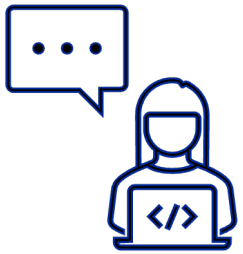
Hooks are not desired state



Running tests

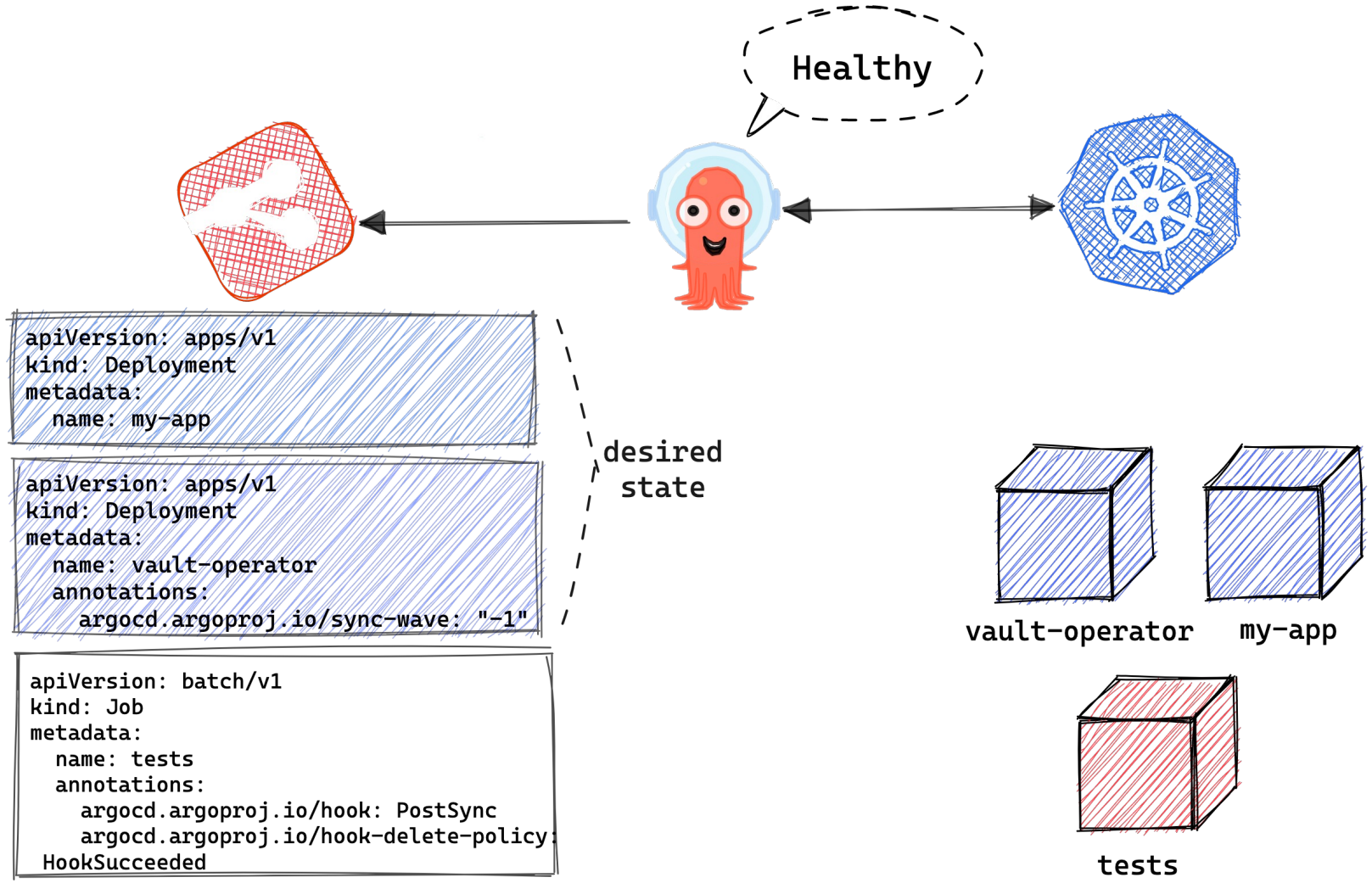


“I want to run tests post deployment..”

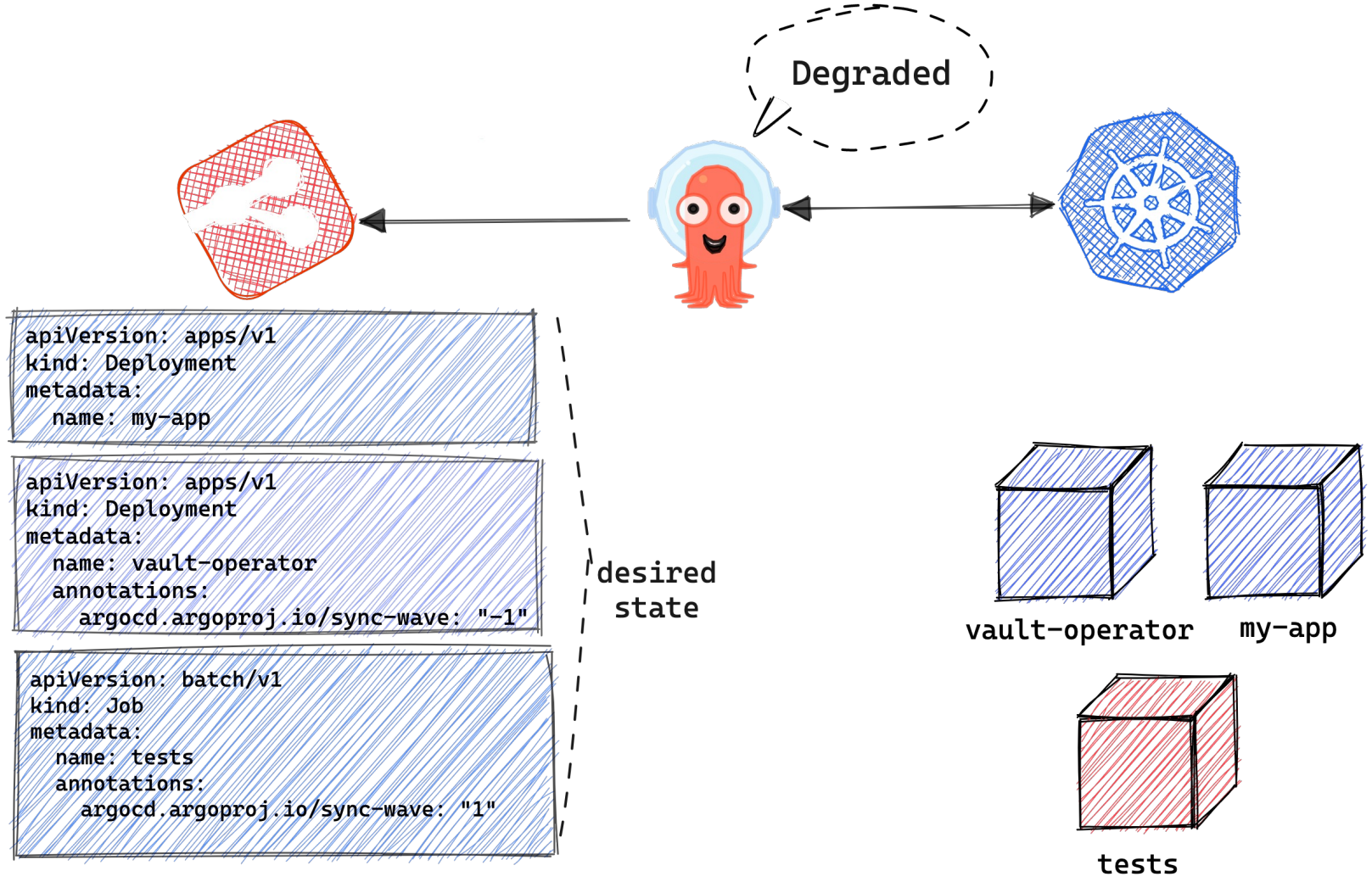


“…for dev environments and streaming applications.”

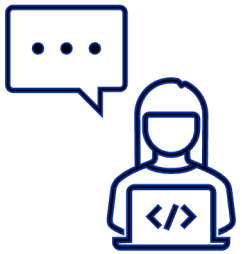
Assert health outside of Argo CD



Make tests desired state



Environment configuration



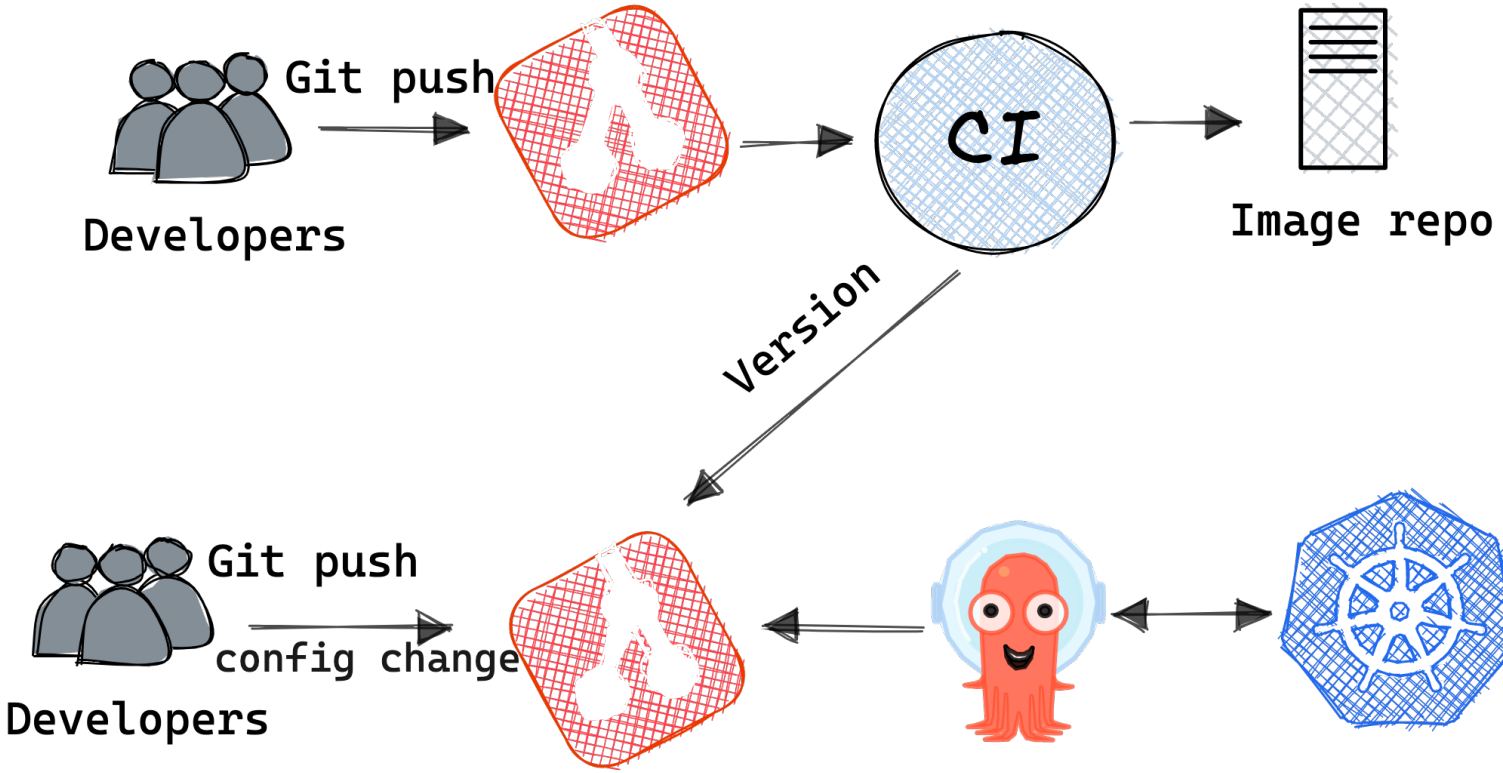
“I want to continuously deploy new changes (outside image tags).”

As a developer, I often need to change:

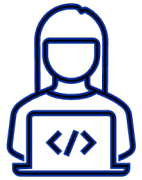
- Business configuration
- Deployment configuration

Environment configuration

“Opening 2 PRs breaks my Continuous Deployment.”

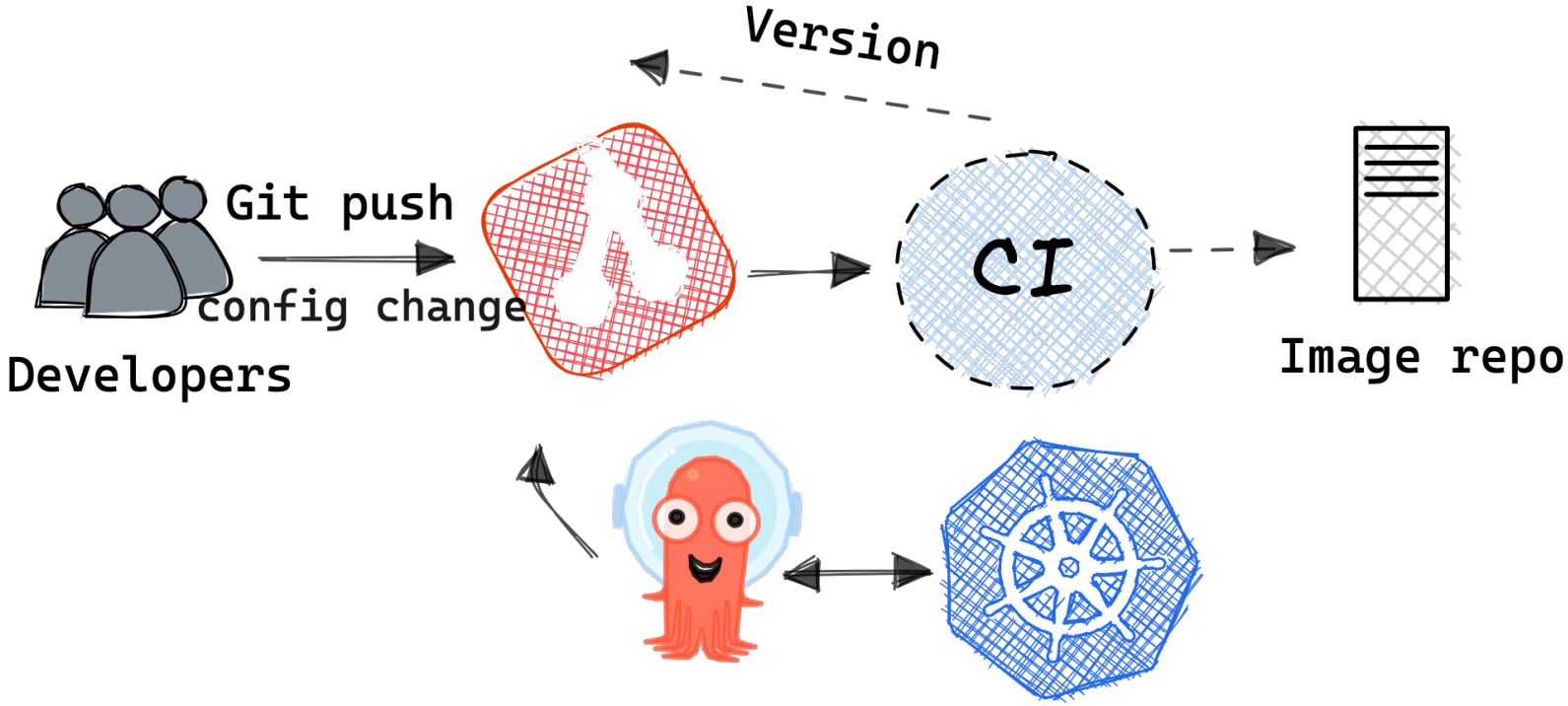


Environment configuration

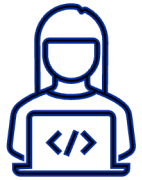


“Opening 2 PRs breaks my Continuous Deployment.”

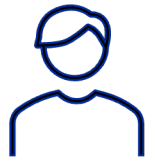
“Just do it in a single repo then.”



Environment configuration



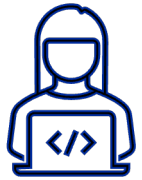
“Opening 2 PRs breaks my Continuous Deployment.”



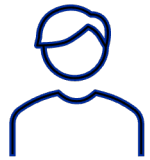
“Just do it in a single repo then.”

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

Environment configuration



“Opening 2 PRs breaks my Continuous Deployment.”

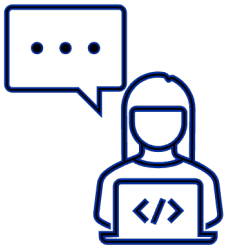


“Just do it in a single repo then.”

“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



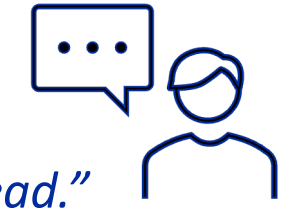
“I don’t want to duplicate my k8s manifests.”

Code & config repo

```
.
└── 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
        │   └── values.yaml
        ├── dev
        │   ├── va7
        │   │   ├── Chart.yaml
        │   │   ├── templates
        │   │   └── values.yaml
        │   └── values.yaml
        └── values.yaml
```

Environment configuration

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

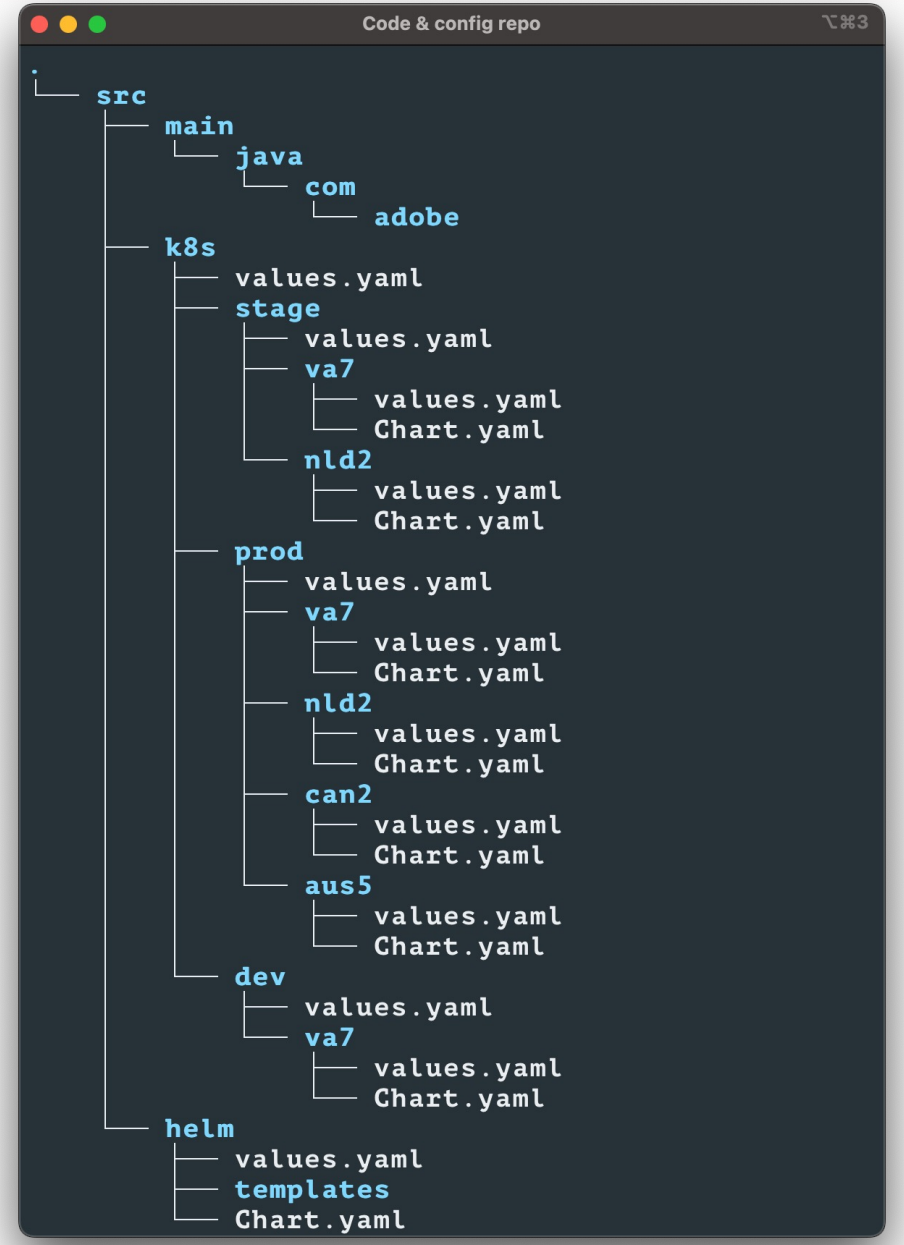


“Use an umbrella chart instead.”

Code & config repo

```
.
└── 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
        │   └── values.yaml
        ├── dev
        │   ├── va7
        │   │   ├── Chart.yaml
        │   │   ├── templates
        │   │   └── values.yaml
        │   └── values.yaml
        └── values.yaml
```

Environment configuration



Environment configuration

Umbrella chart

```yaml
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

Code & config repo ⌥⌘3

```
.
└── 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

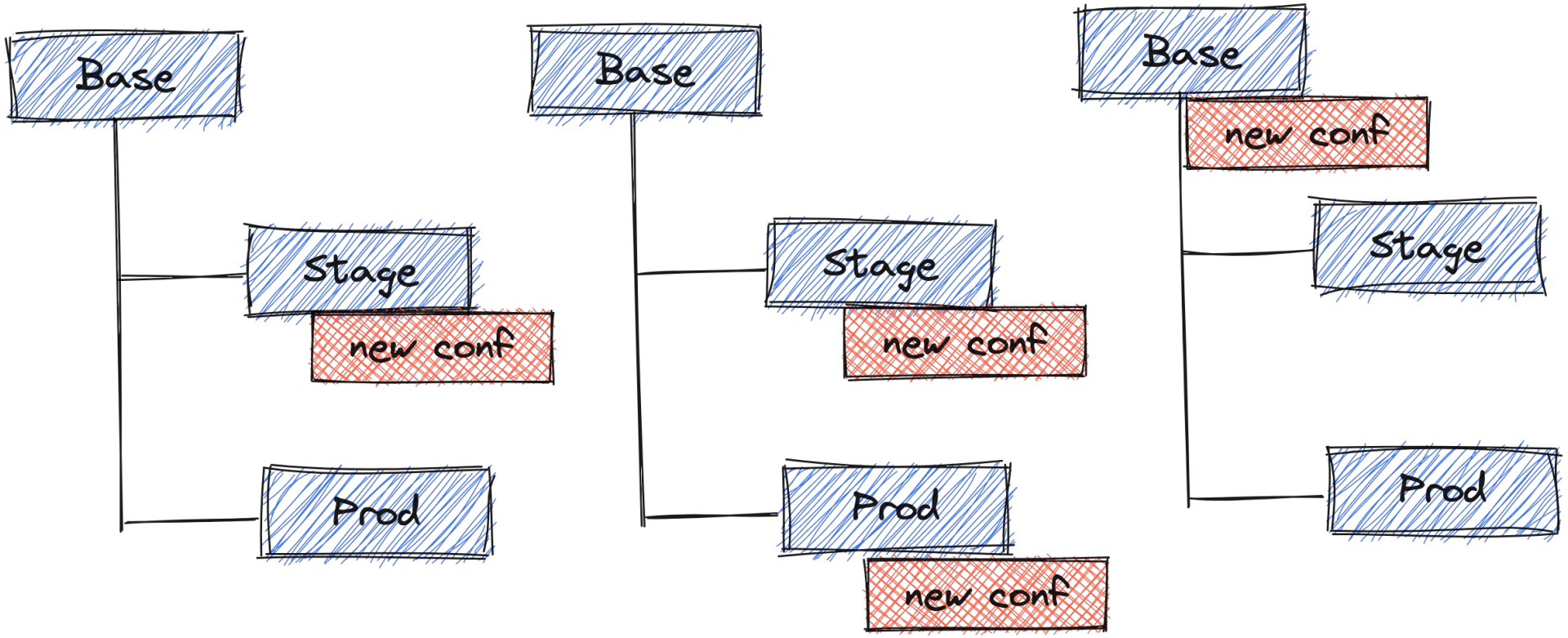
Code & config repo ⌥⌘3

```
.
└── 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
```

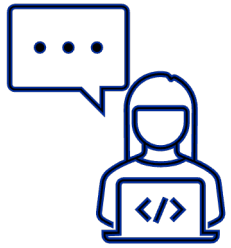
“I don’t want to deploy my change directly to prod..”

“… 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.”

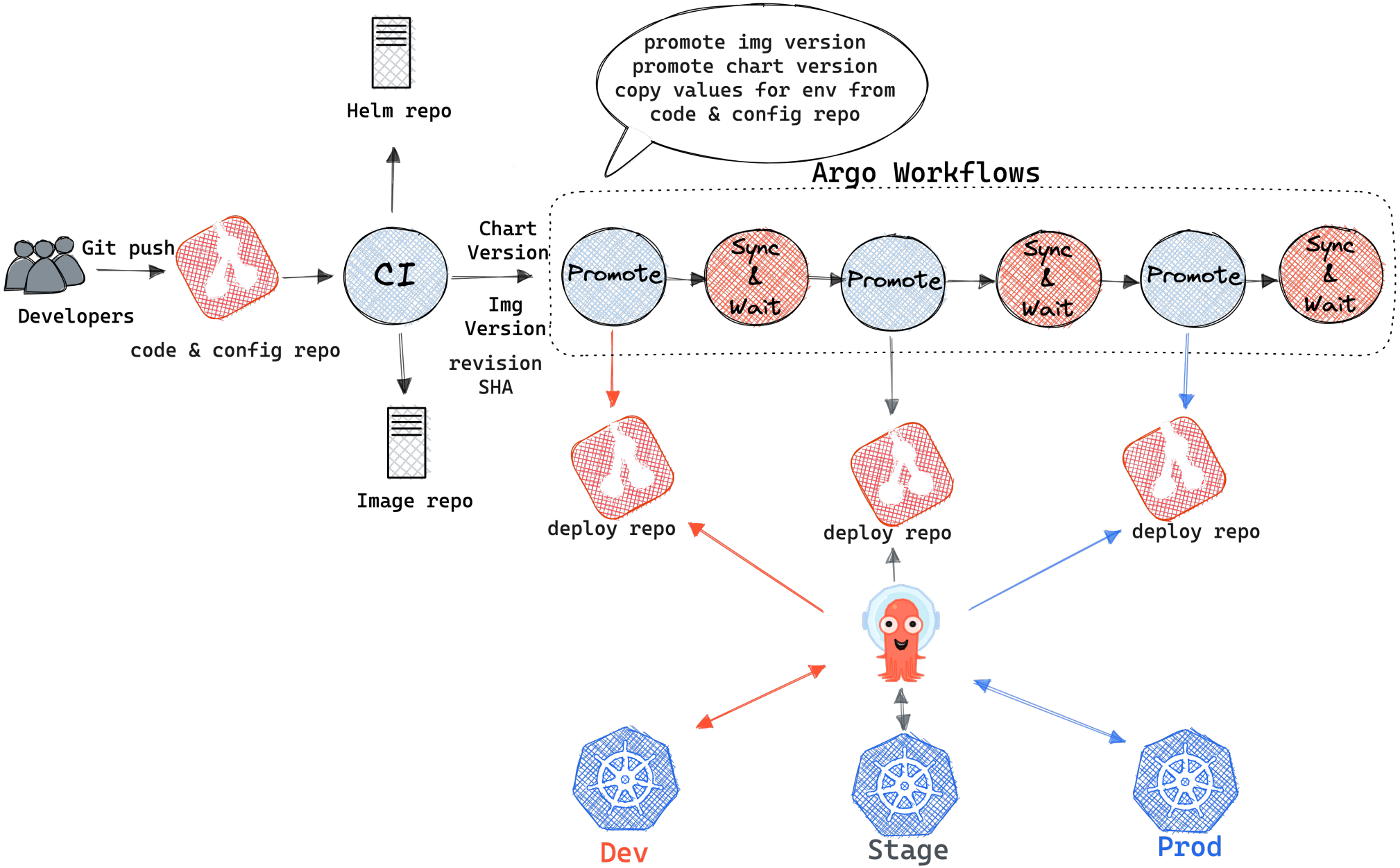
Code & config repo

```
.
└── 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
```

Deploy repo

```
.
├── stage
│   ├── va7
│   │   ├── values.yaml
│   │   └── Chart.yaml
│   └── nld2
│       ├── values.yaml
│       └── Chart.yaml
├── prod
│   ├── va7
│   │   ├── values.yaml
│   │   └── Chart.yaml
│   ├── nld2
│   │   ├── values.yaml
│   │   └── Chart.yaml
│   ├── can2
│   │   ├── values.yaml
│   │   └── Chart.yaml
│   └── aus5
│       ├── values.yaml
│       └── Chart.yaml
└── dev
    └── va7
        ├── values.yaml
        └── Chart.yaml
```

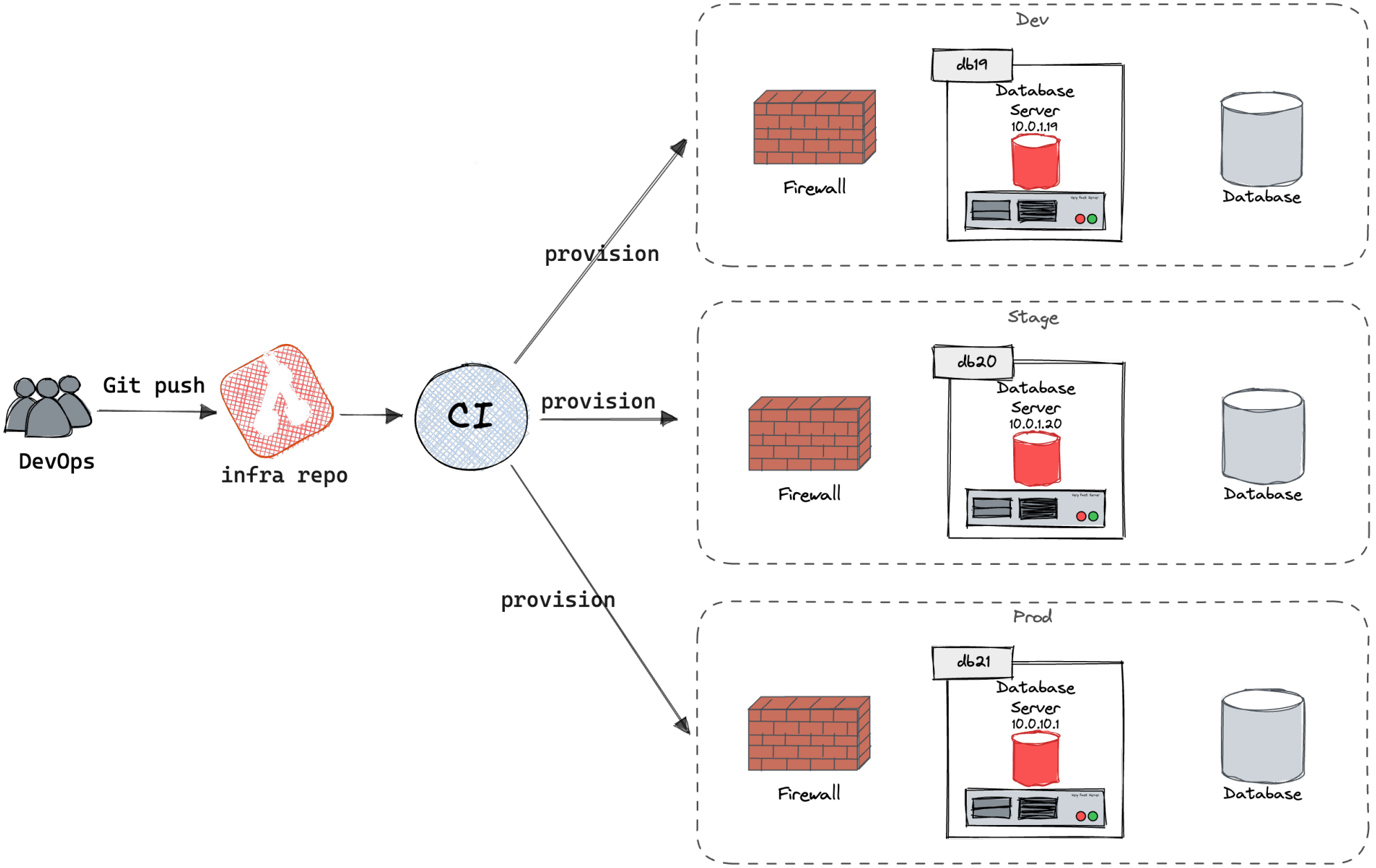
What we do


promote img version
promote chart version
copy values for env from
code & config repo
Helm repo
Argo Workflows
Git push
Developers
code & config repo
CI
Chart Version
Img Version
revision SHA
Promote
Sync & Wait
Promote
Sync & Wait
Promote
Sync & Wait
Image repo
deploy repo
deploy repo
deploy repo
Dev
Stage
Prod

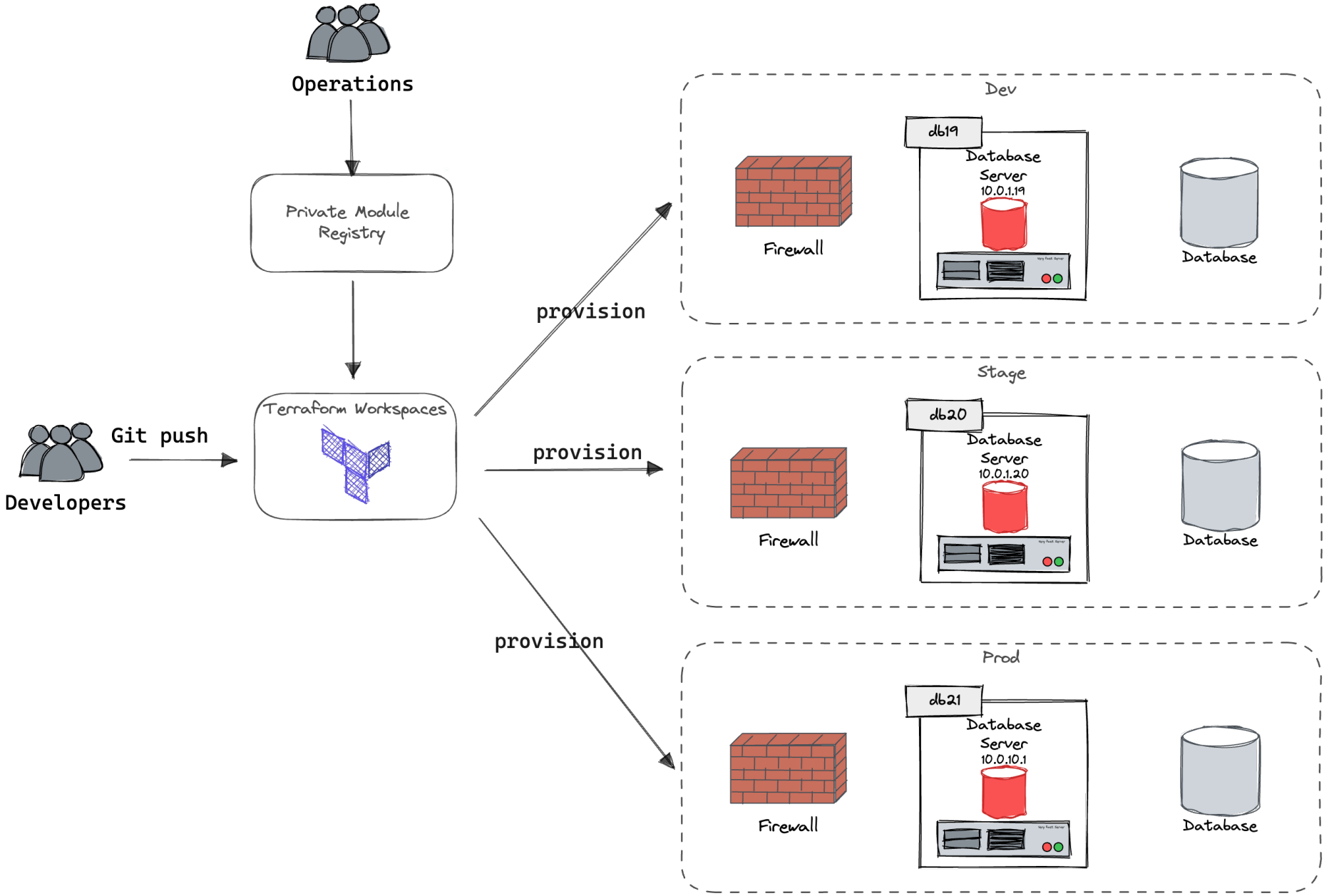
Managing apps & infrastructure

Bridging the gaps

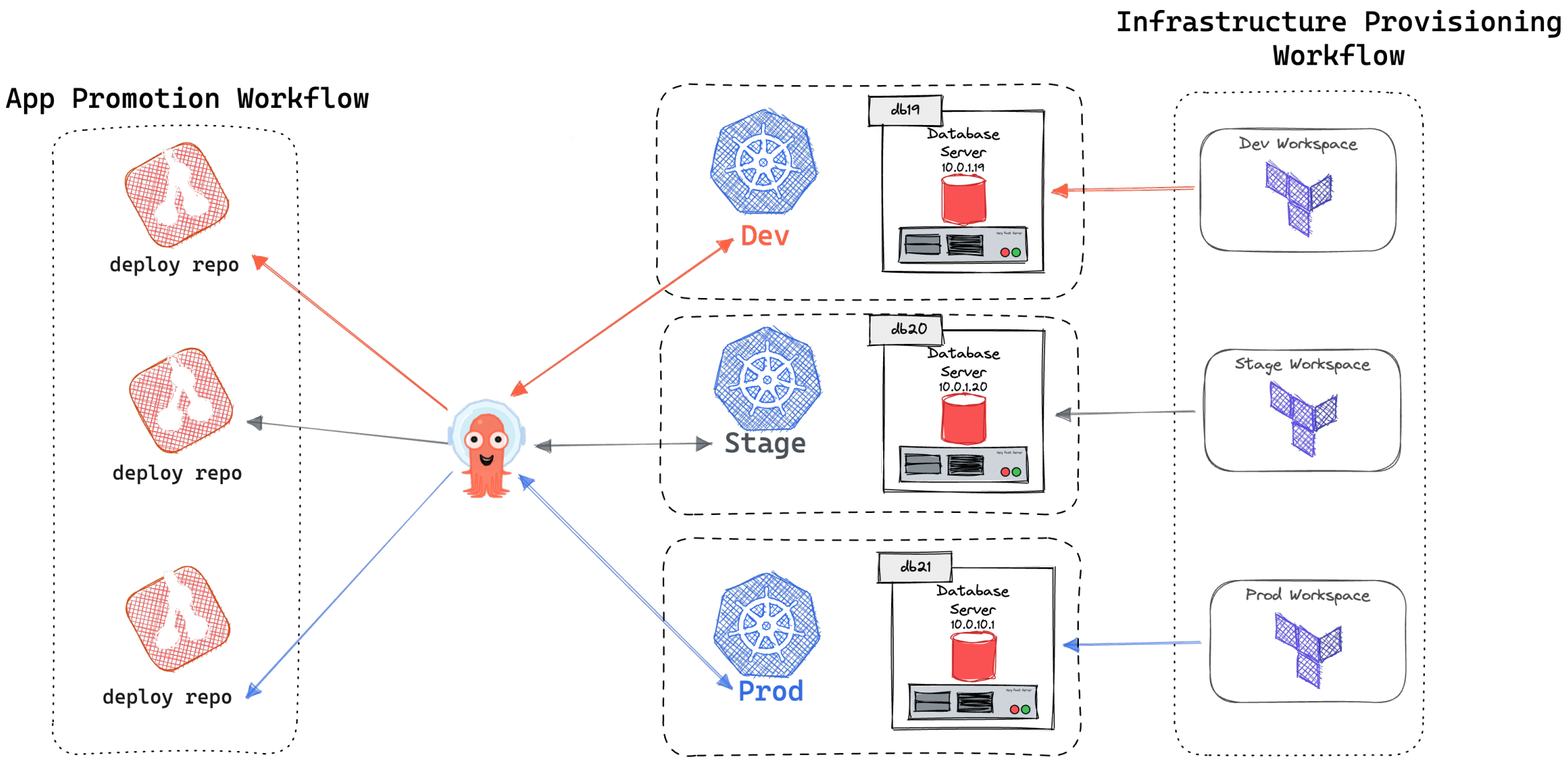
What about infrastructure?



What about infrastructure? (2)



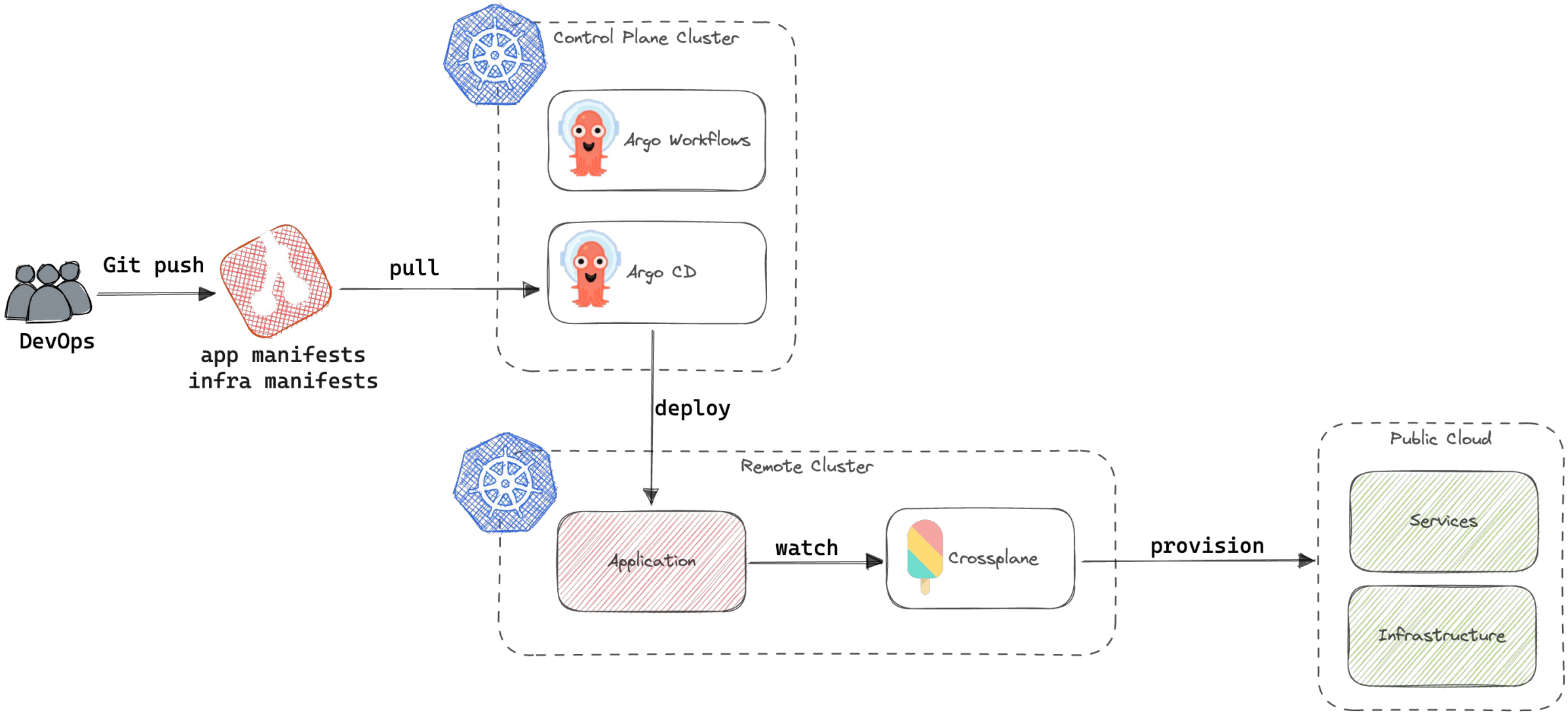
What about infrastructure? (3)



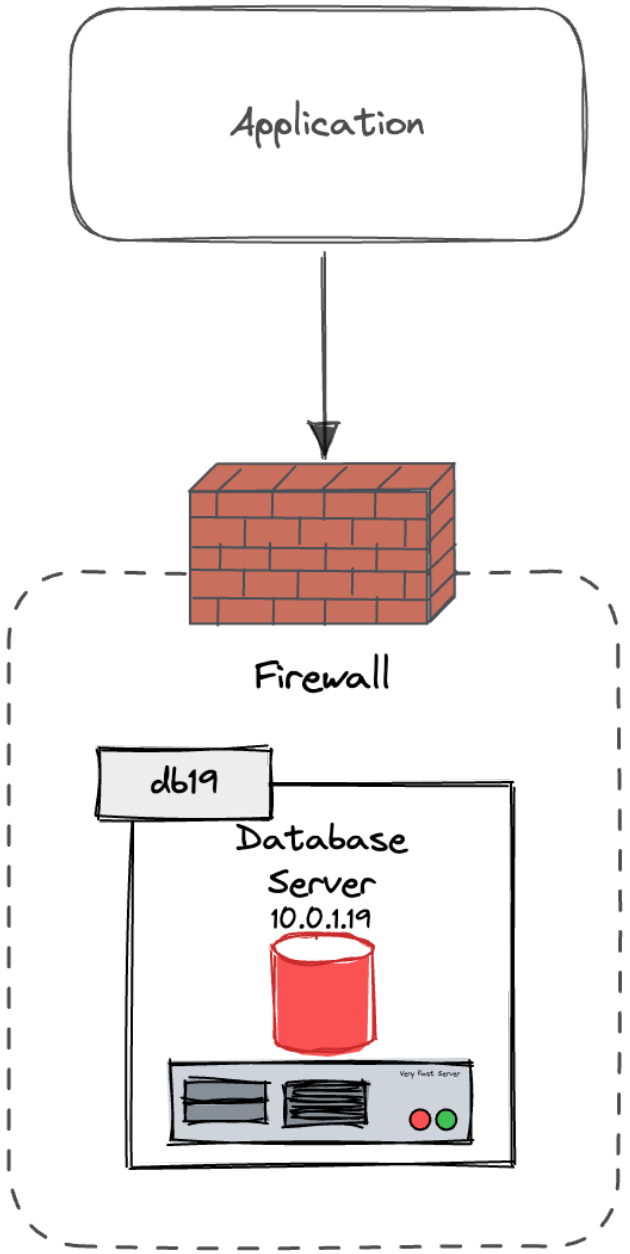
Introducing Crossplane

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

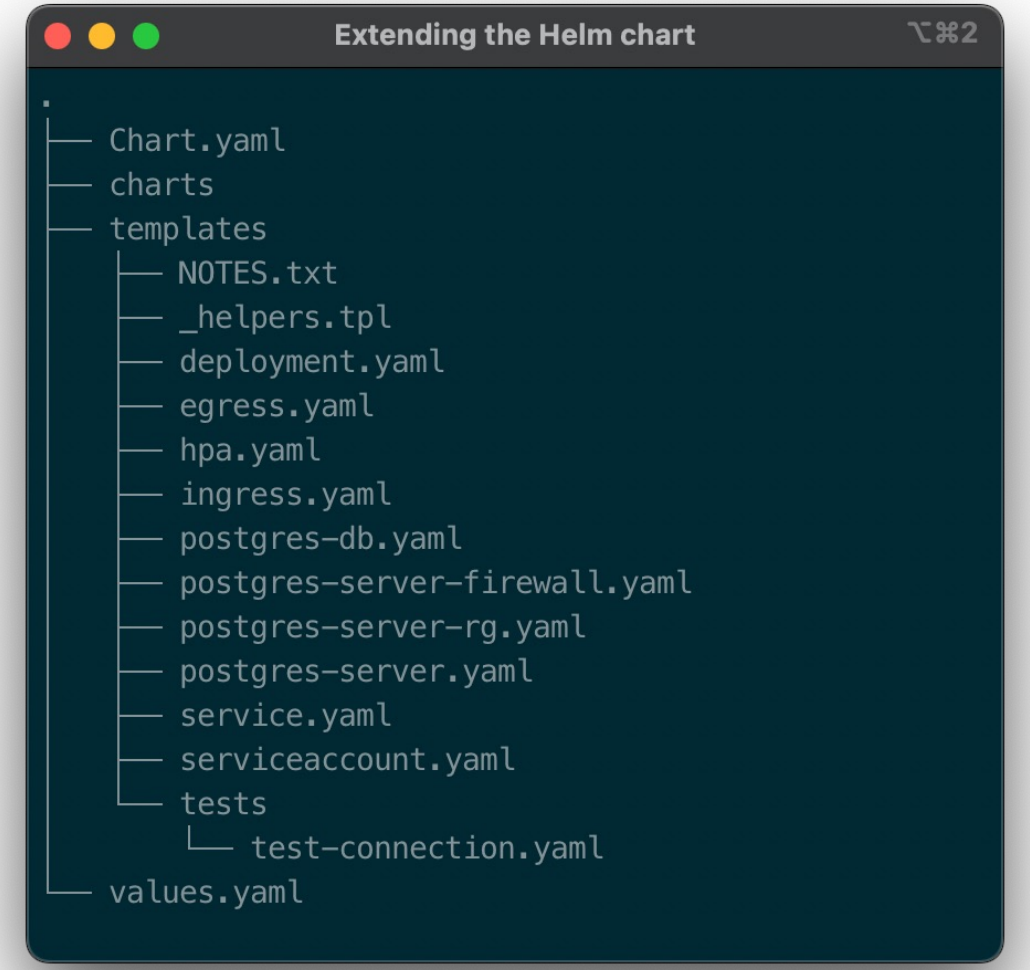
Argo CD and Crossplane



A simple example



A simple example (2)



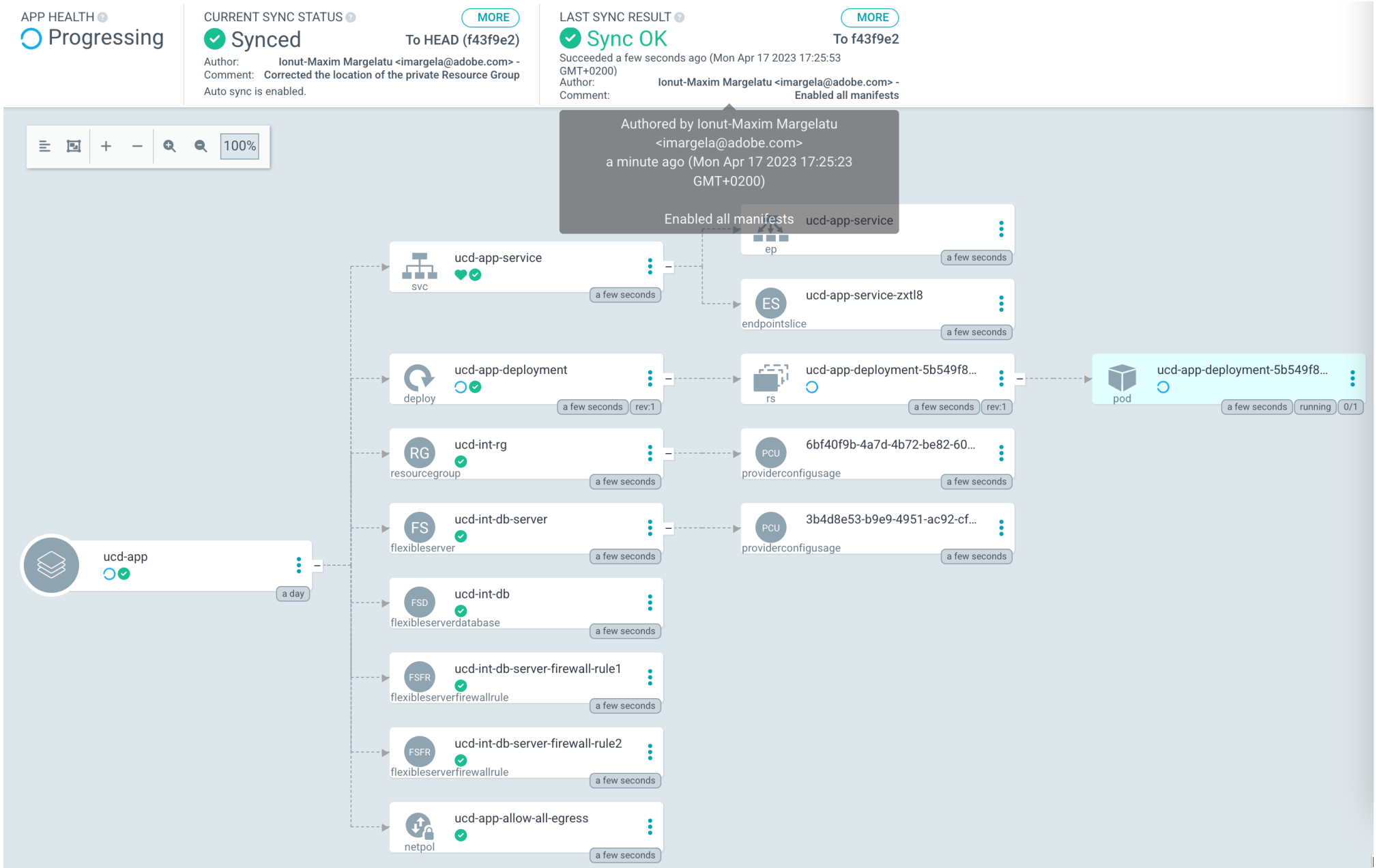
A simple example (3)


```yaml
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"
```

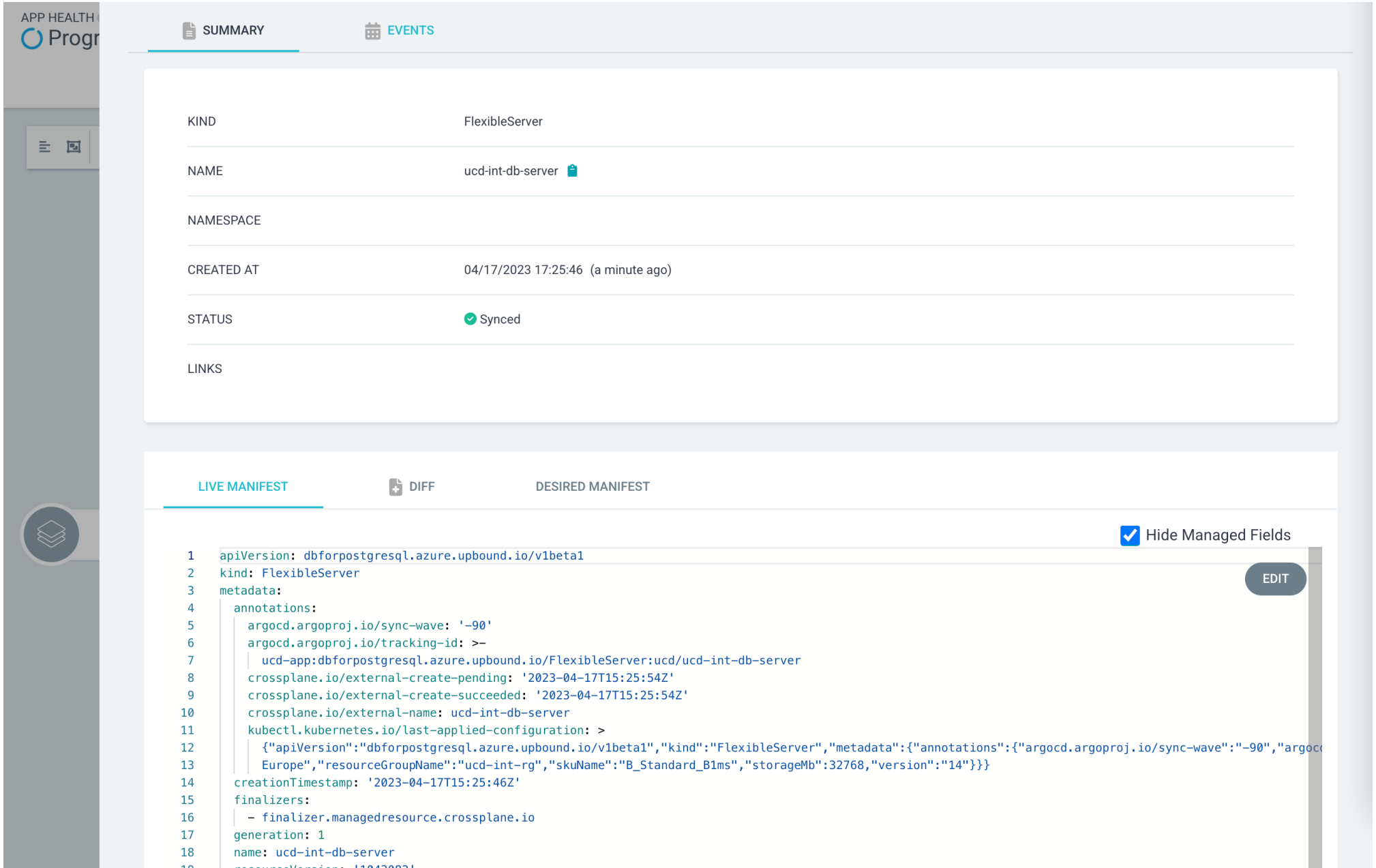
Ordering via sync waves


```yaml
apiVersion: dbforpostgresql.azure.upbound.io/v1beta1
kind: FlexibleServer
metadata:
  name: {{ .Values.application.dbServerName | quote }}
  annotations:
    argocd.argoproj.io/sync-wave: "-90"
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
```

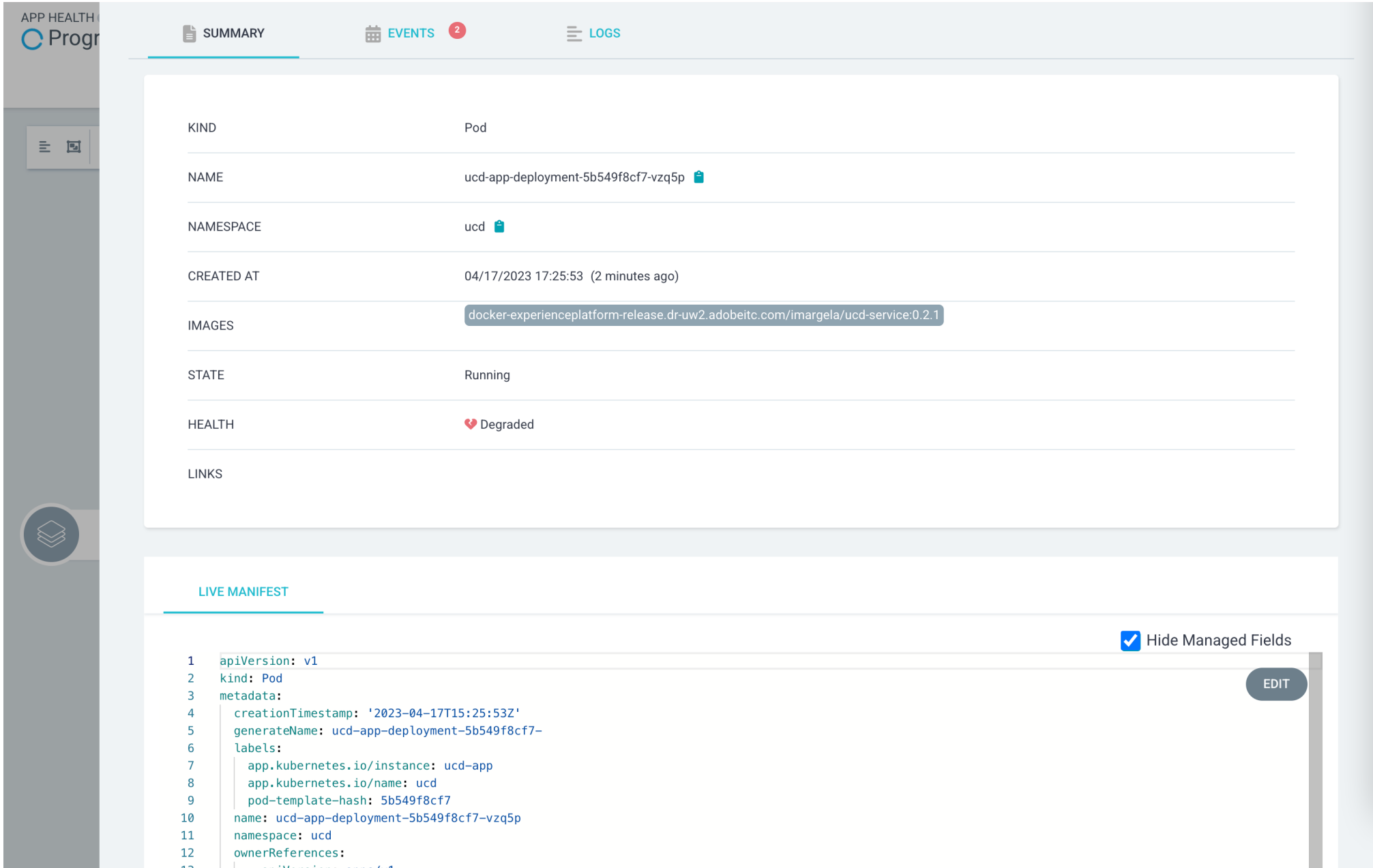
Deploying application and infrastructure



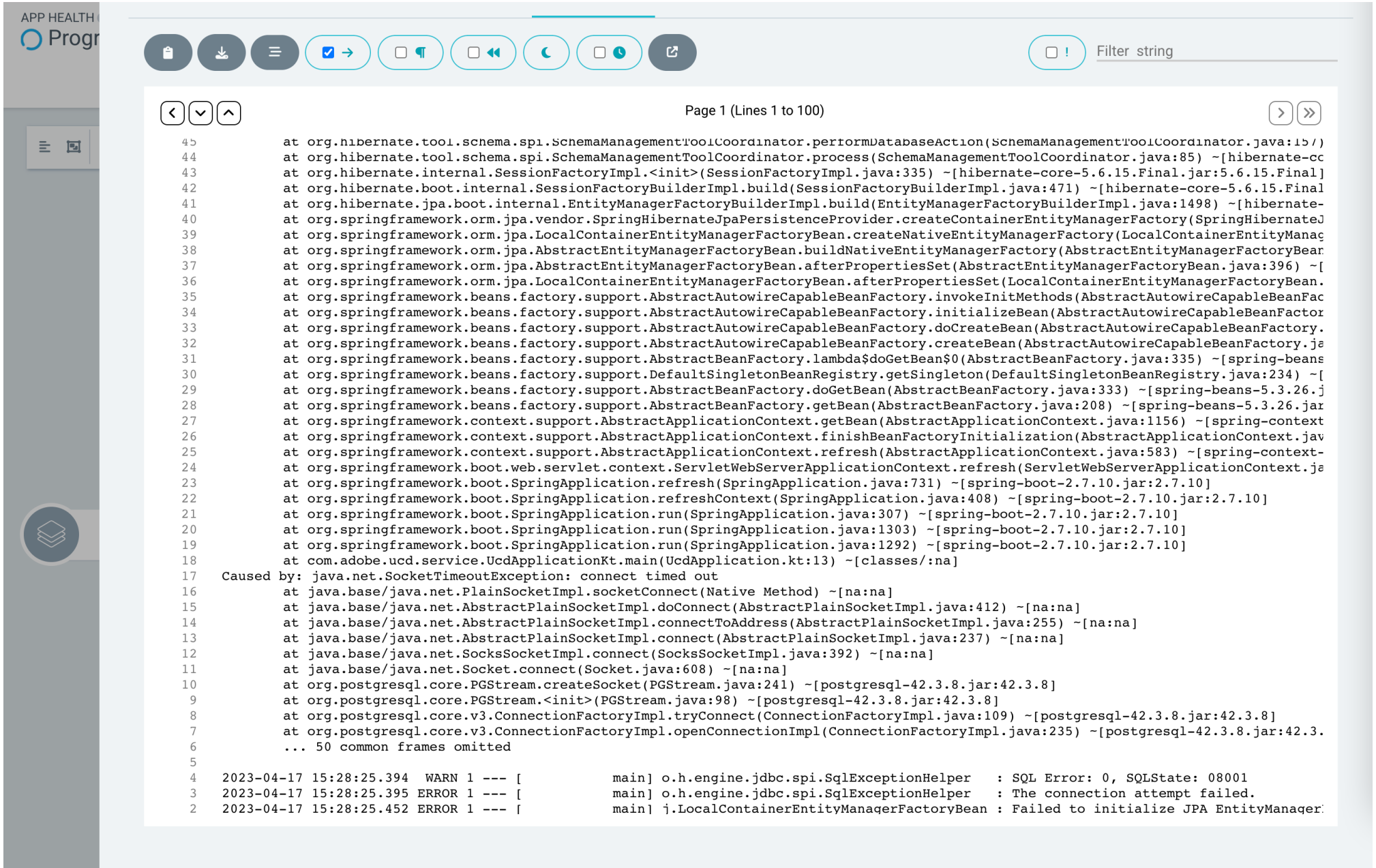
Deploying application and infrastructure



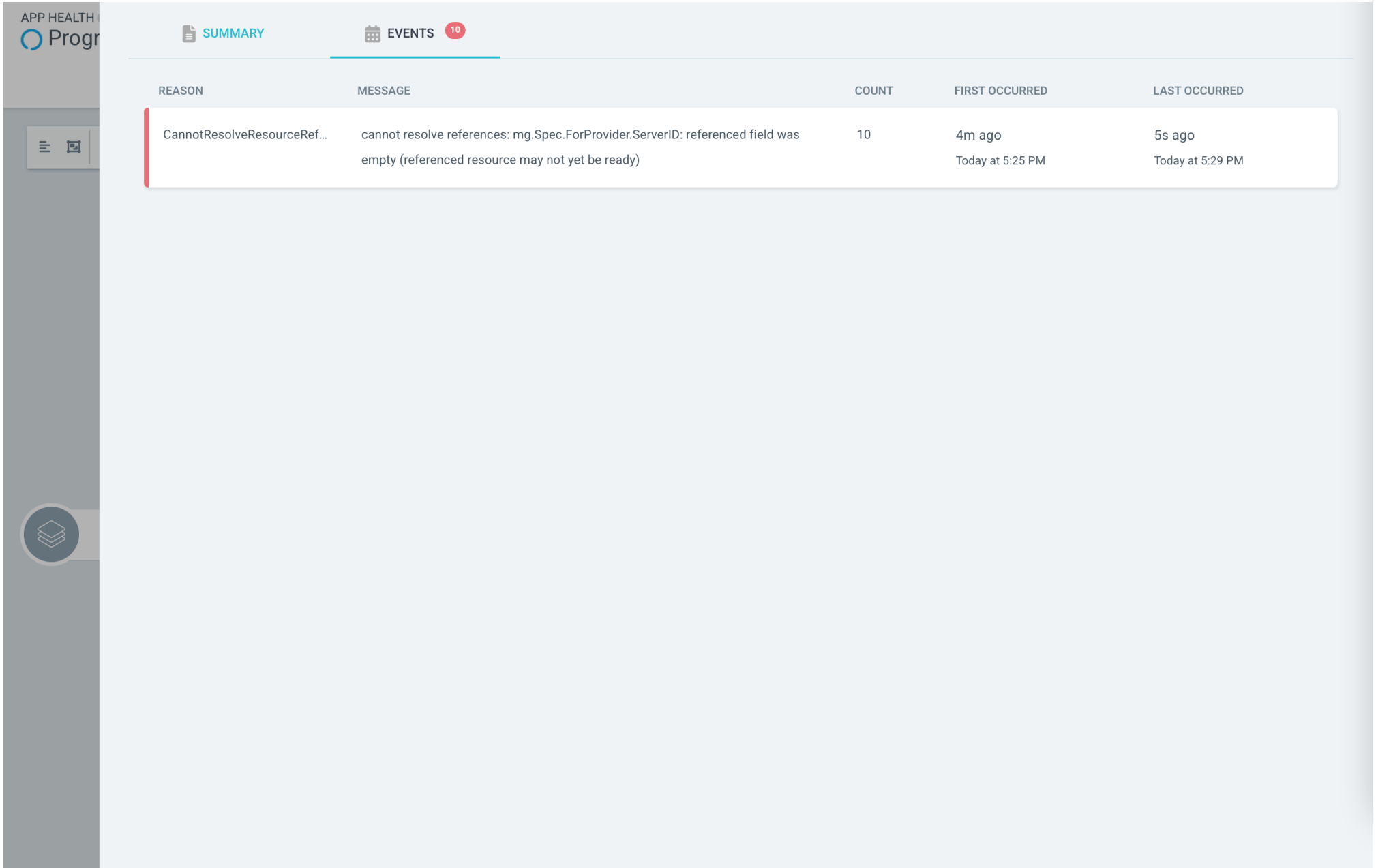
Deploying application and infrastructure



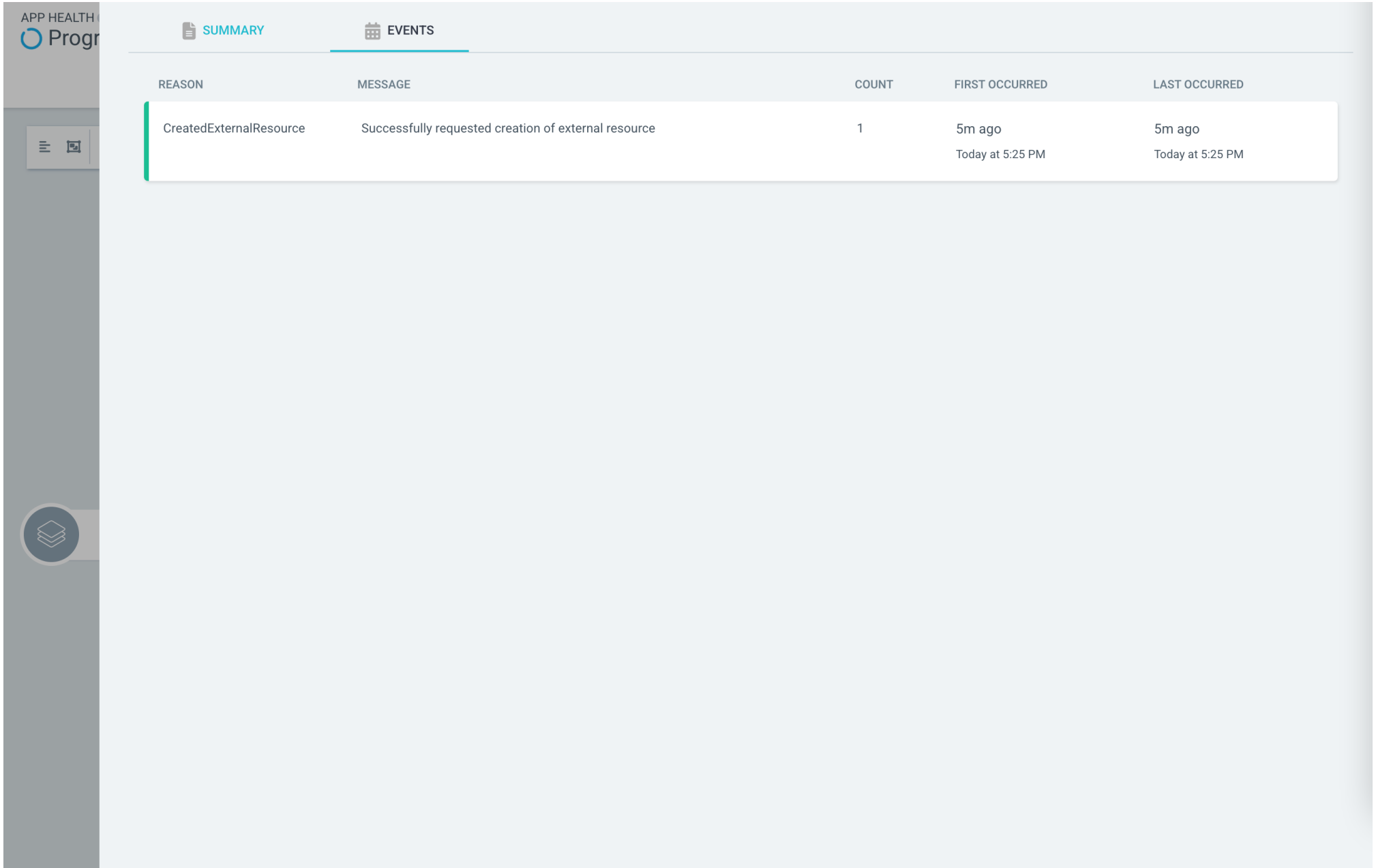
Deploying application and infrastructure



Deploying application and infrastructure



Deploying application and infrastructure



Crossplane resource state in Argo CD

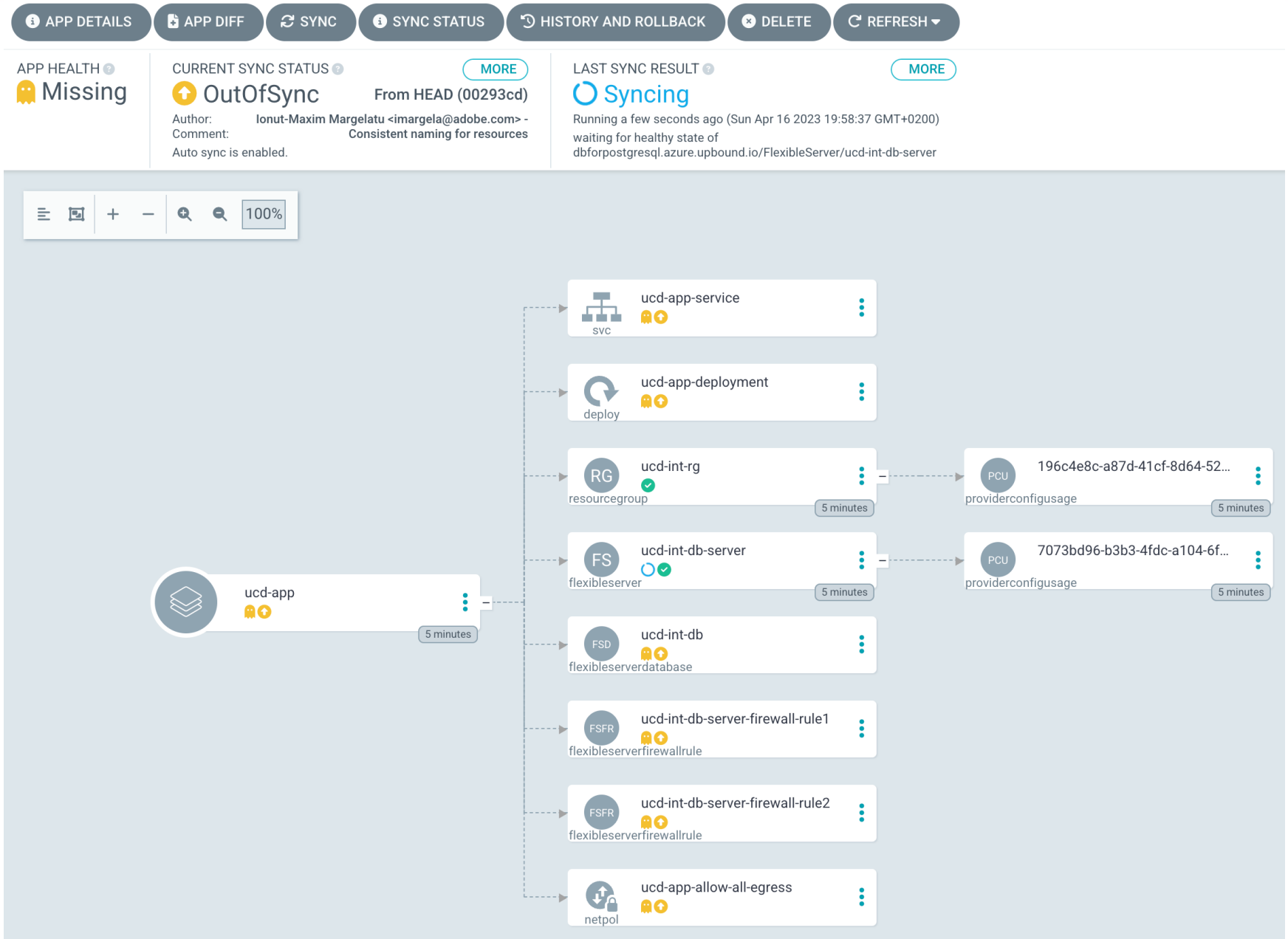
By default, Argo uses the state of the Crossplane resource,

NOT the actual state of the provisioned infrastructure.

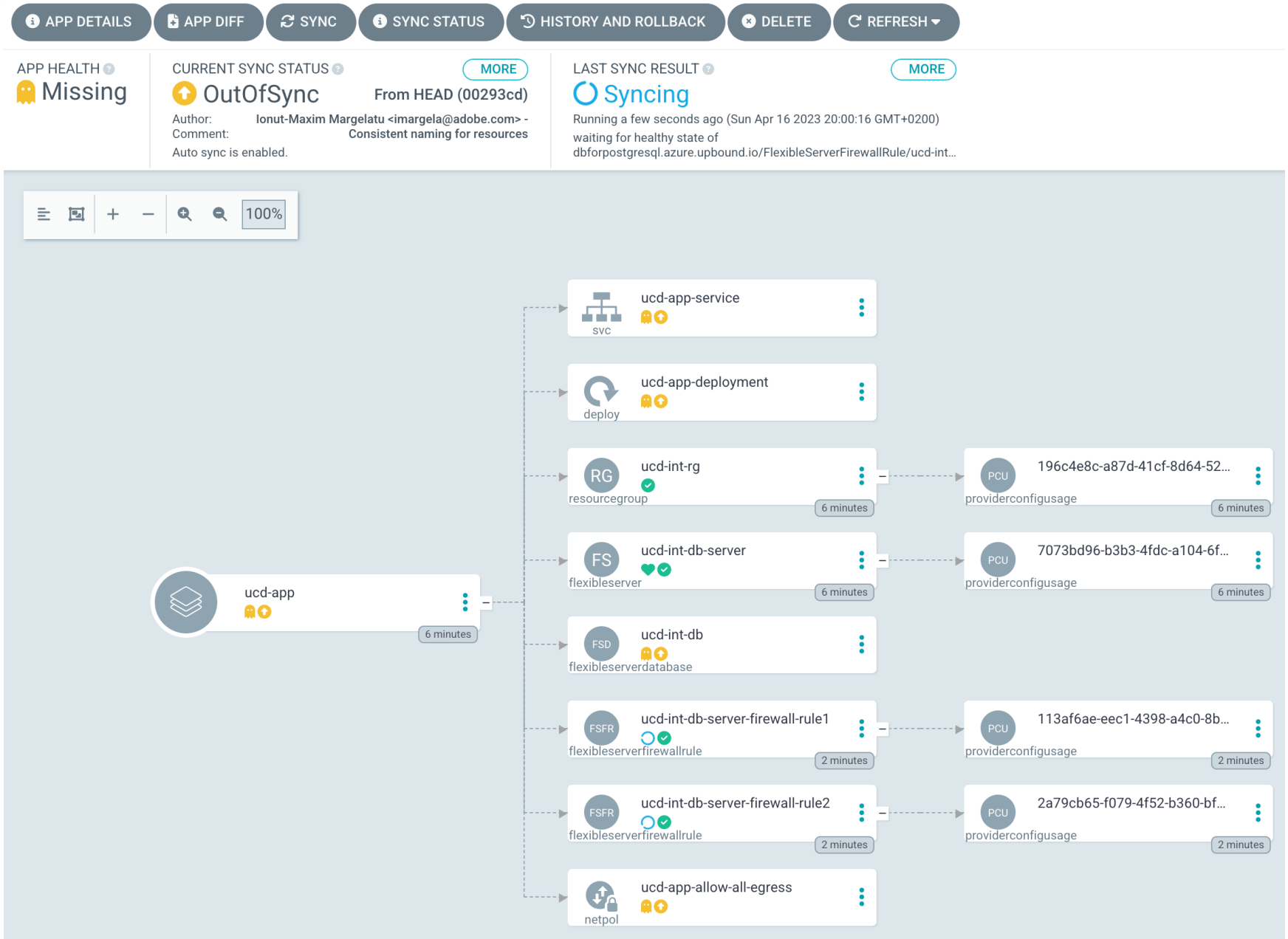
Custom resource health in Argo CD


```
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
```

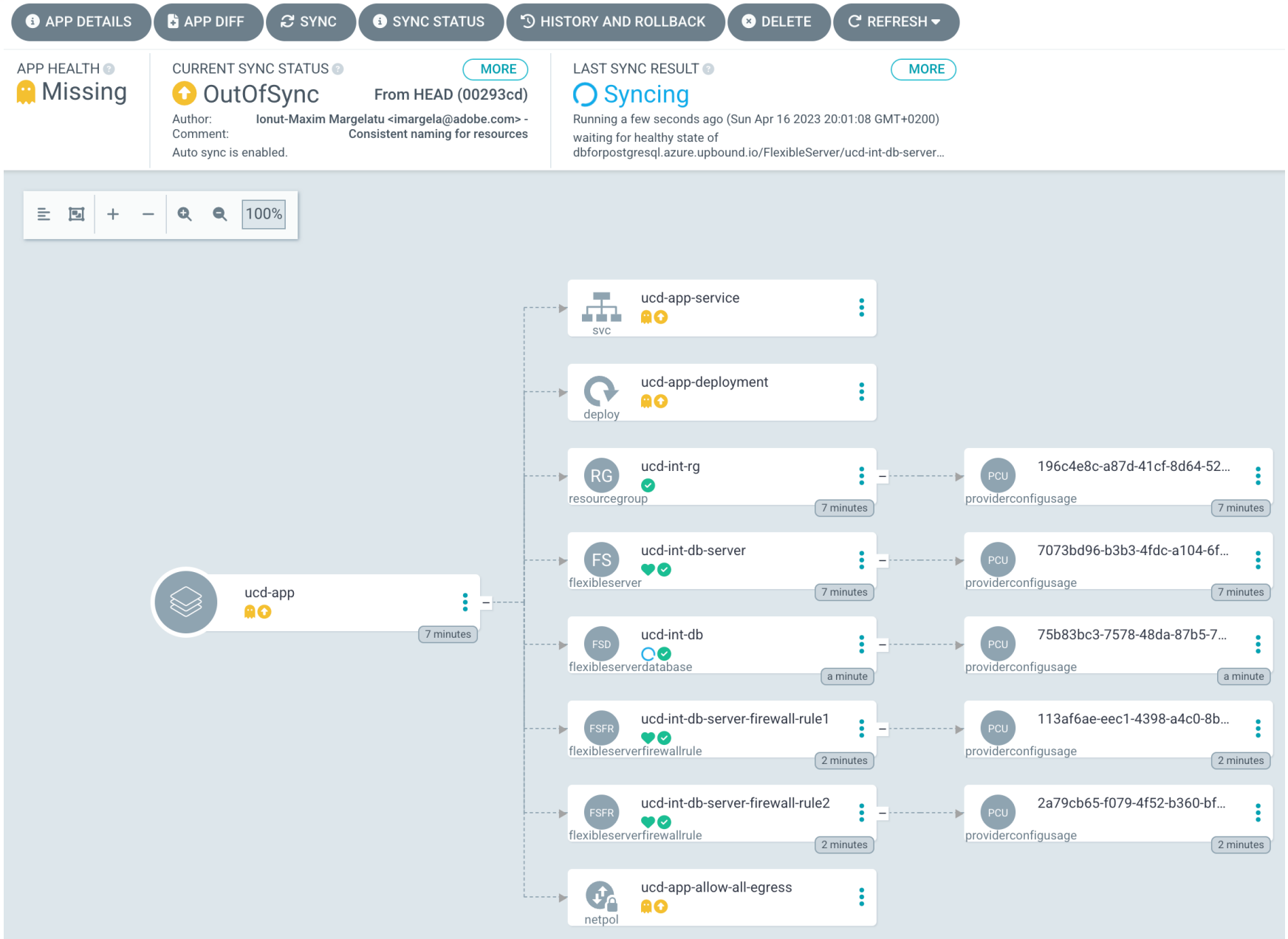
Synchronized deployment



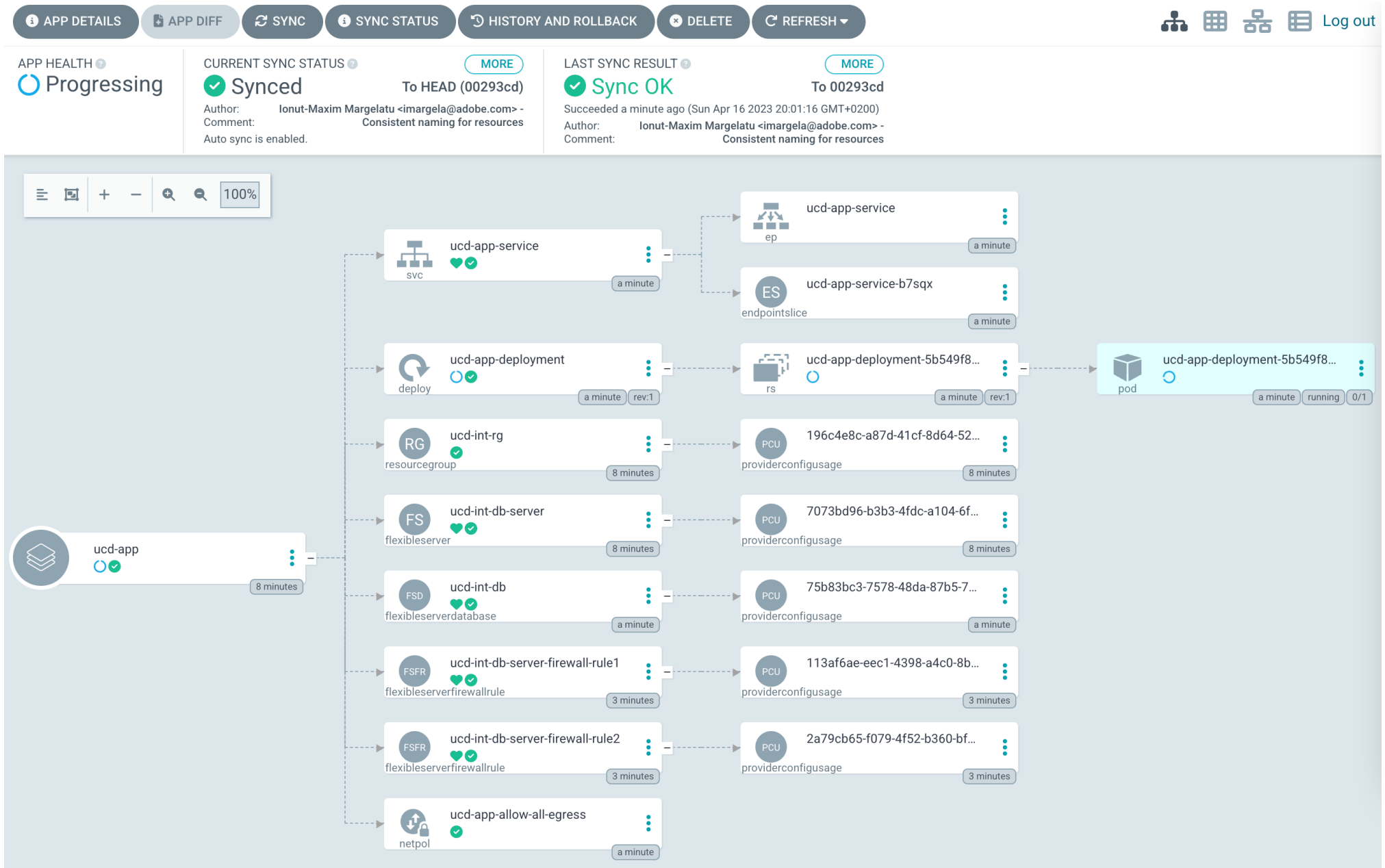
Synchronized deployment



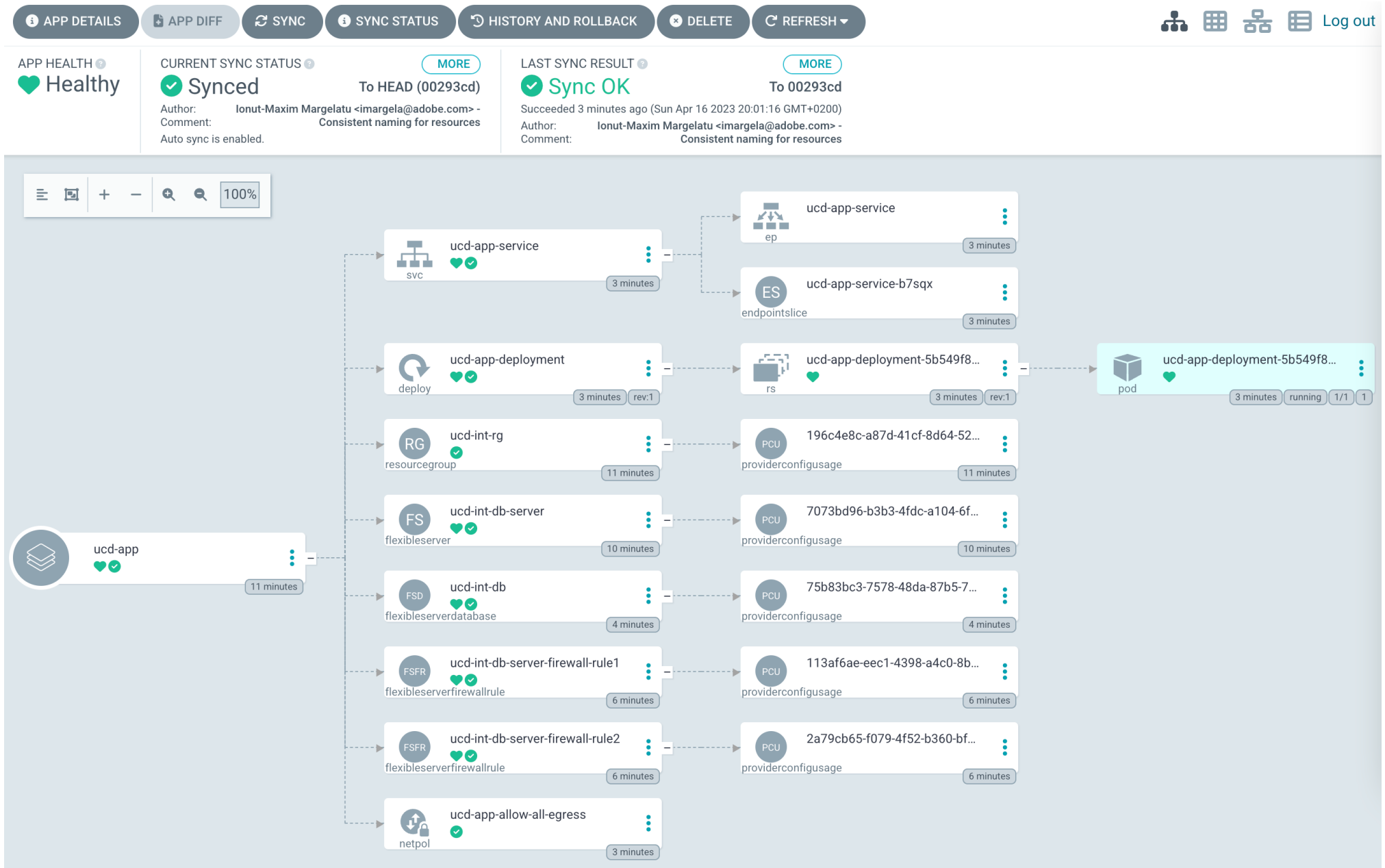
Synchronized deployment



Synchronized deployment



Synchronized deployment



Infrastructure up & running

KubeCon CloudNativeCon Europe 2023

```
~ 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
```

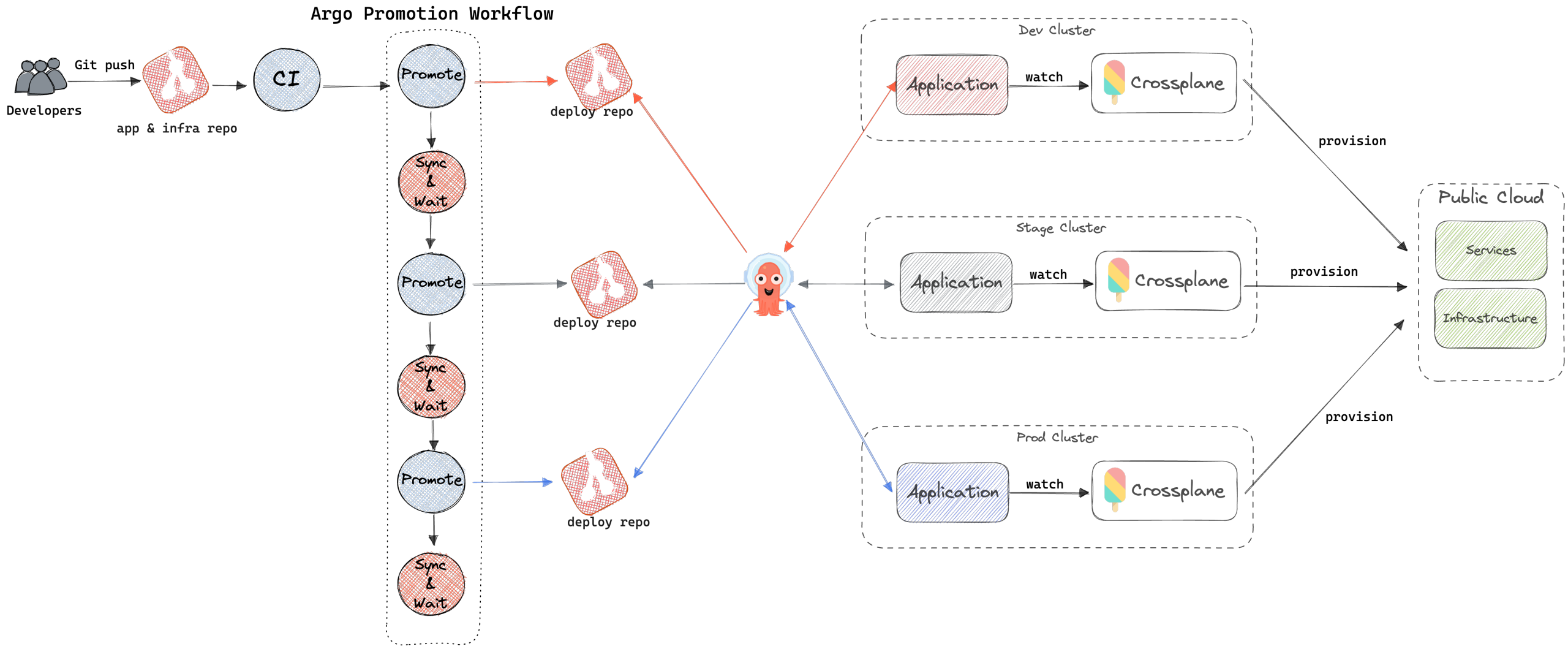
Application running


```
~ 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