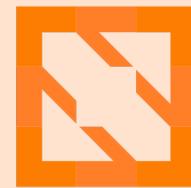




KubeCon



CloudNativeCon

---

Europe 2022

---

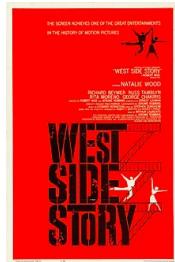
WELCOME TO VALENCIA



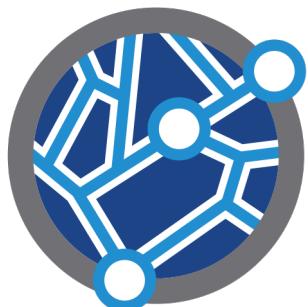
# West Side CD

Benoit Moussaud, VMware Tanzu

# West Side CD



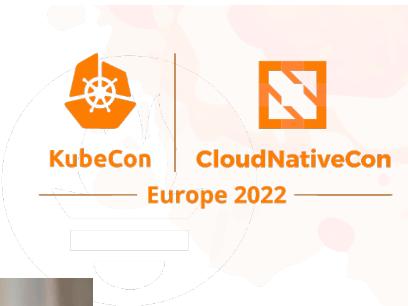
From *Orchestration* to *Choreography* to design a CI/CD process running in Kubernetes Clusters using Cartographer



**Benoit Moussaud**  
Senior Solution Engineer



VMware Tanzu



mCor  
merica 2021

# About Me

Paris,



Former: Java Developer – Architect – Consultant

Past a Decade In CI/CD + DevOps env

From XebiaLabs To  VMware Tanzu



@bmoussaud



@bmoussaud



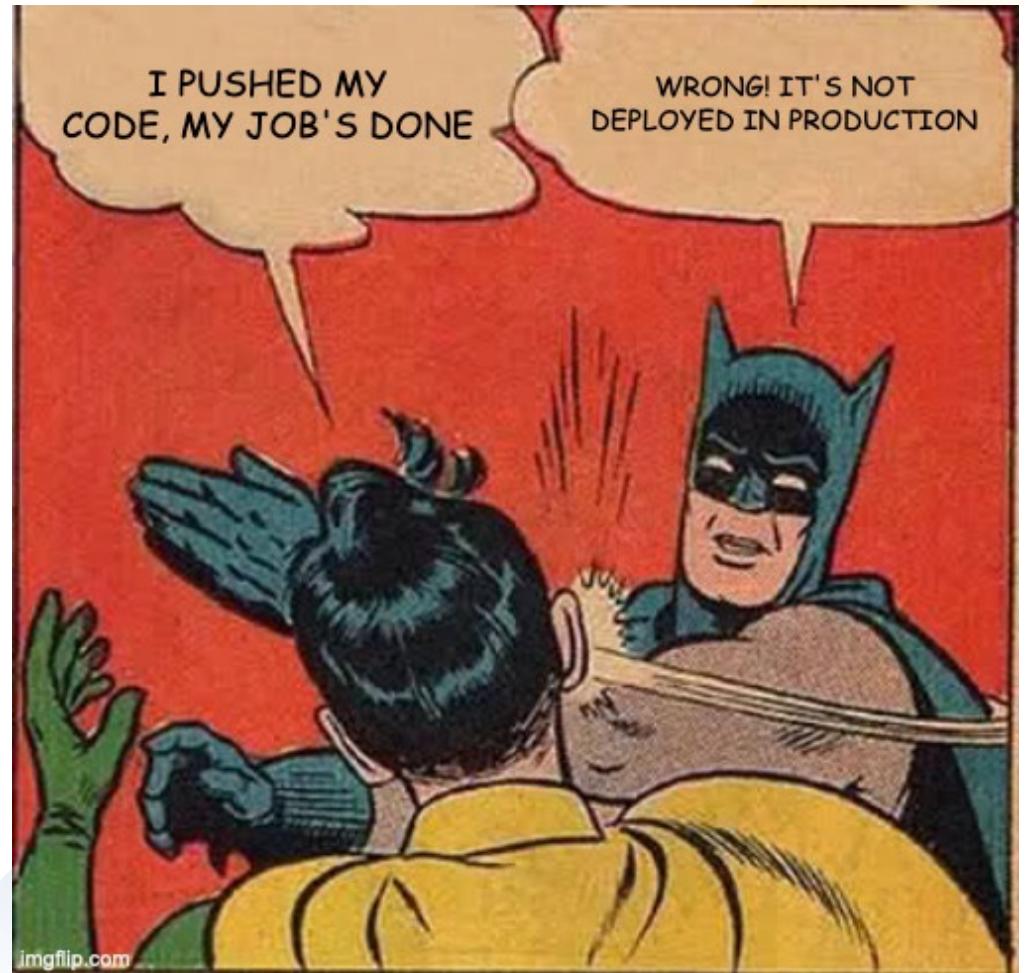
**Benoit Moussaud**  
Senior Solution Engineer



VMware Tanzu

*“An application that is not deployed in a production environment is useless”*

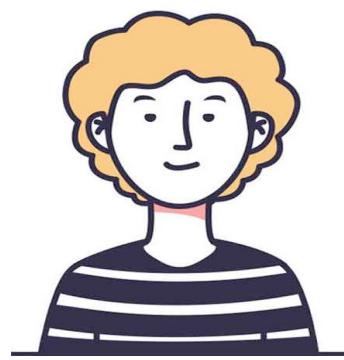
“Benoit Moussaud” ;-)



# Personas



DEV



APPOPS



SRE

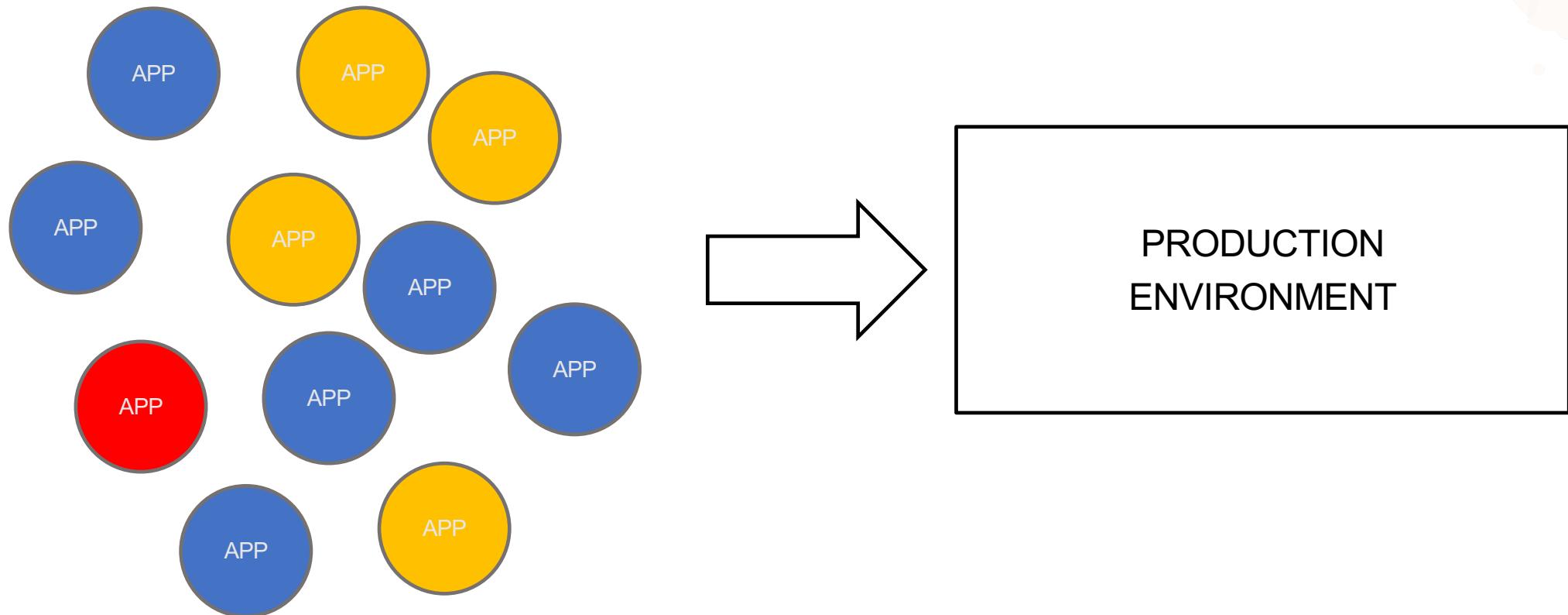


OPS

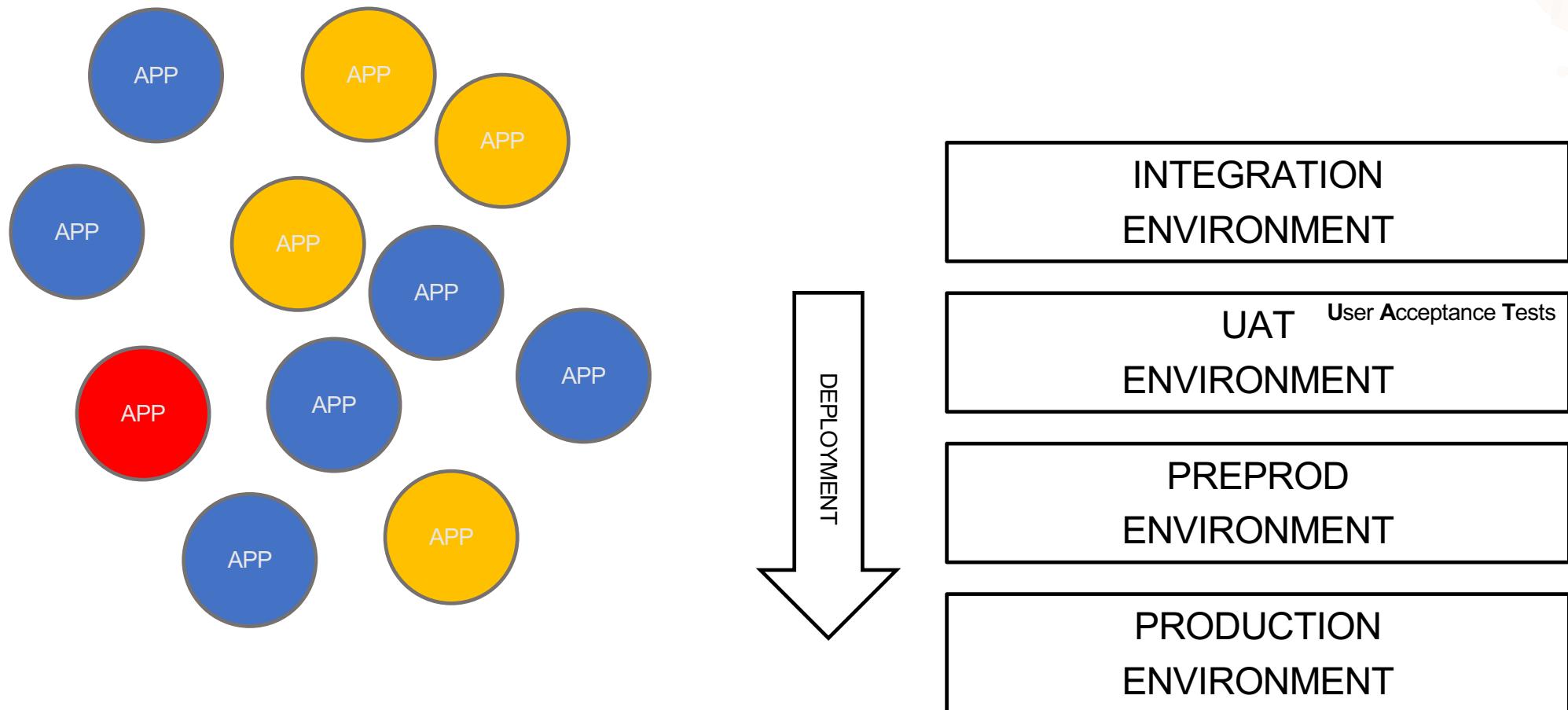
know as DevOps too

Site Reliability Engineer

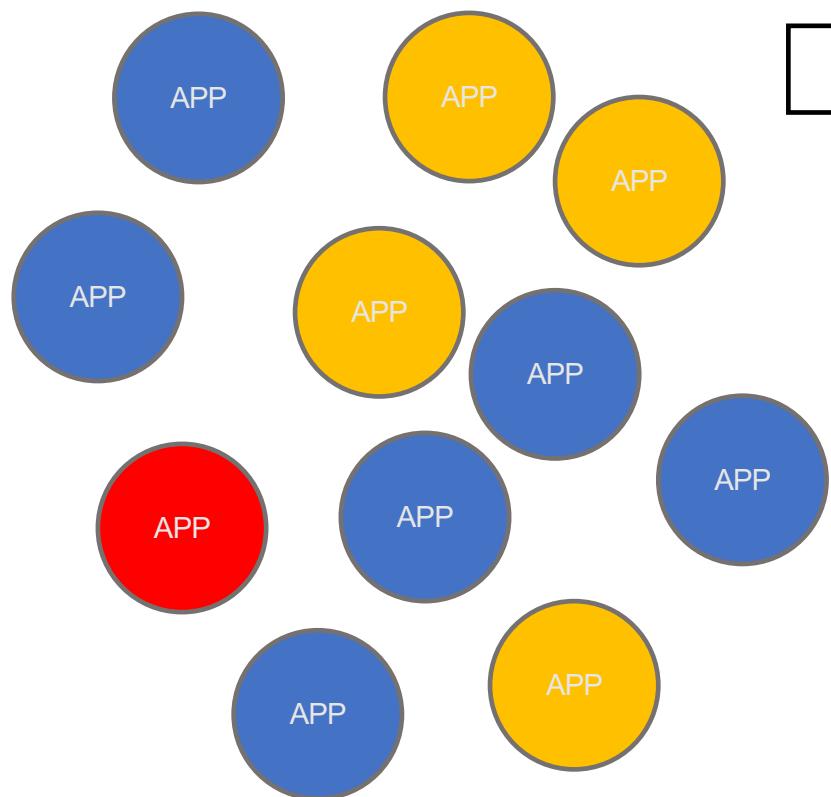
# What !



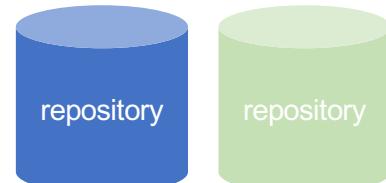
# Promote !



# Integrate !

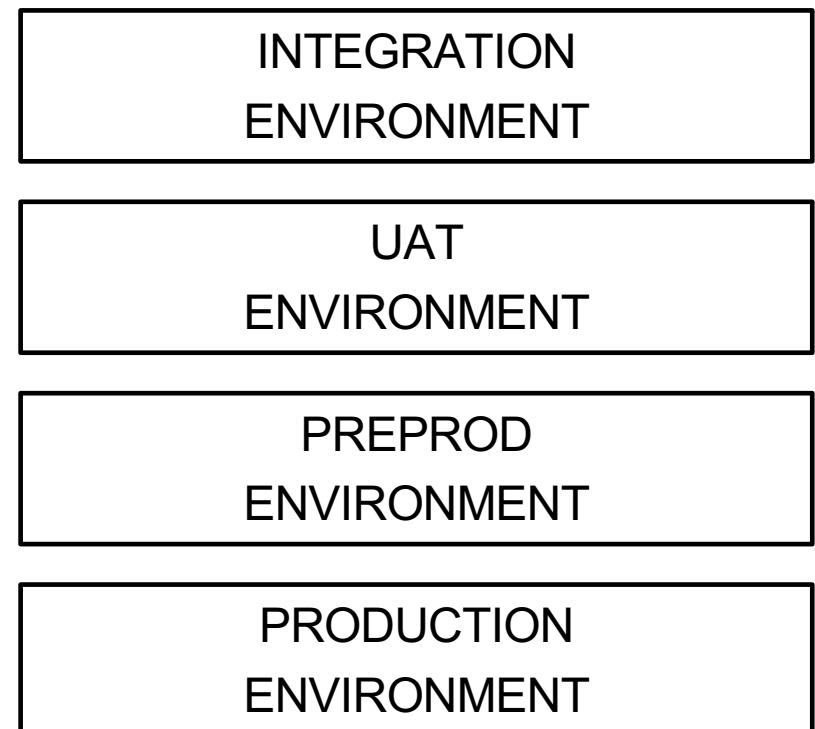


INTEGRATION



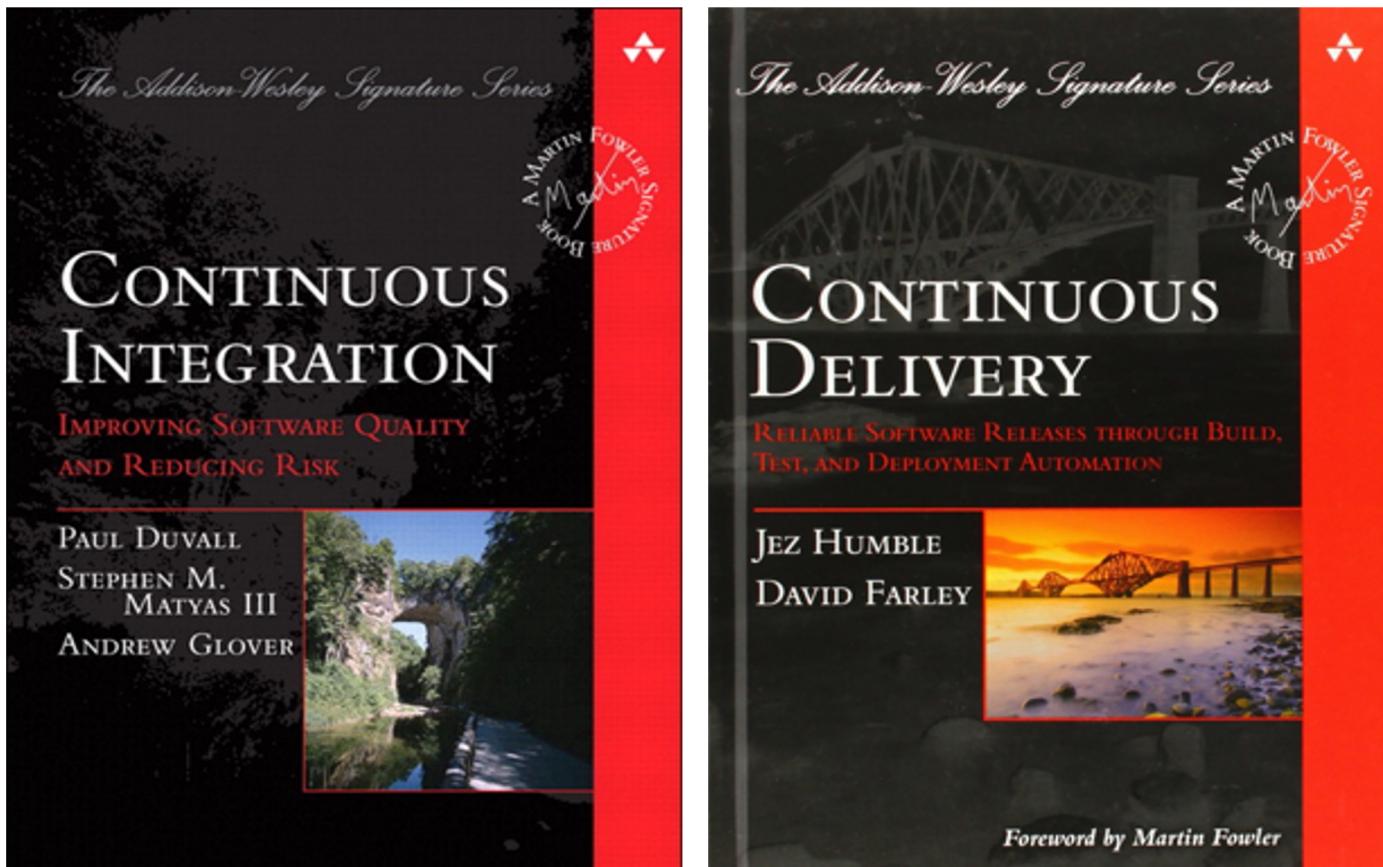
Referenceable / Versionable  
KubeCon CloudNativeCon  
Europe 2022

- Artifact
- Container Image
- Commit / Tag



DEPLOYMENT

# Books



2007

2010

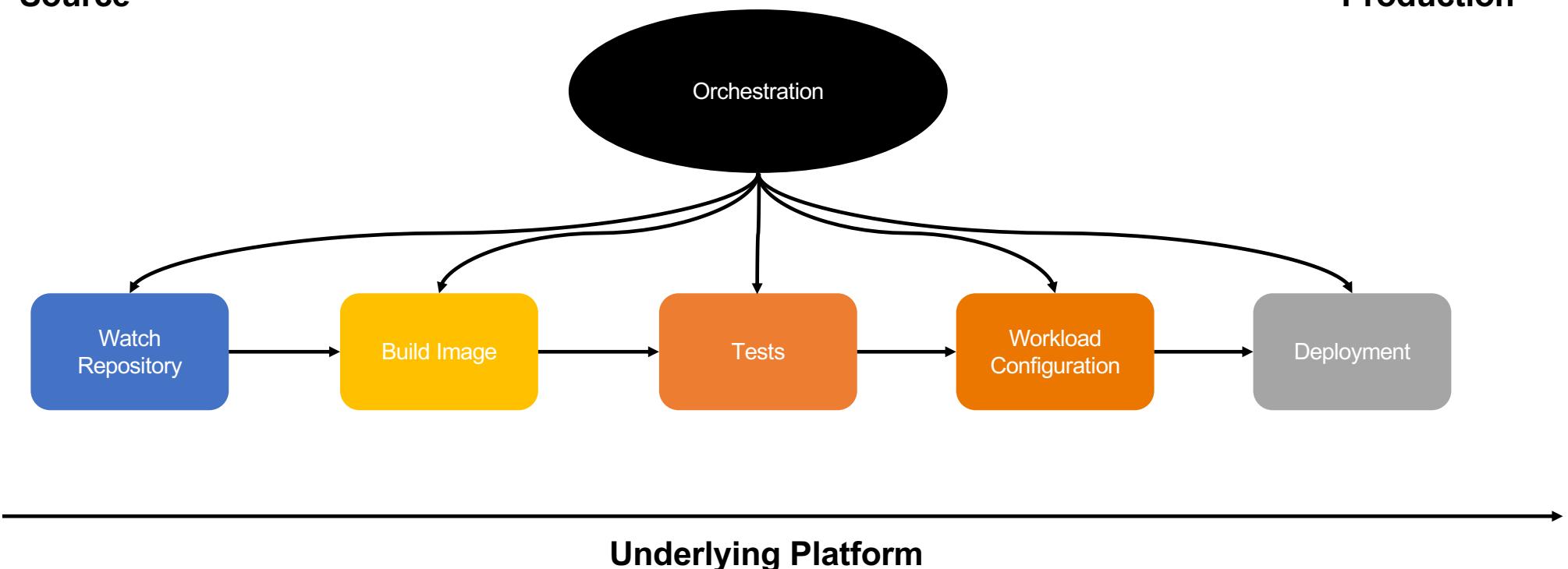
# Current Challenges in CI/CD

- ★ Natural Design
- ★ Impact on modification
- ★ Tightly-coupled
- ★ SPOF
- ★ Rigid



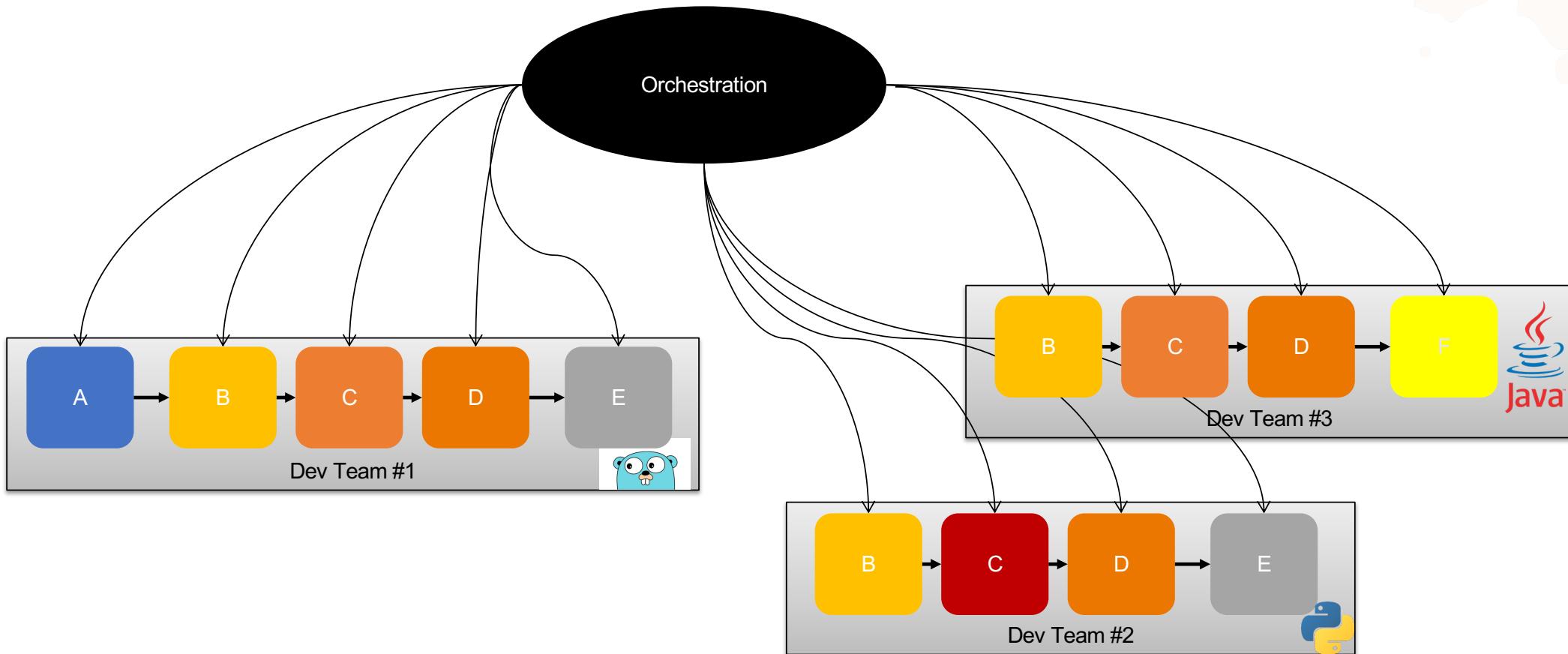
Source

Production



# Current Challenges at Scale

- Path to Production
- Separation of Concern





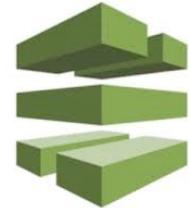
Code stream



Jenkins



Azure DevOps



Code Pipeline



GitHub Actions



Tekton

# The orchestration pattern



# The choreography pattern



KubeCon



CloudNativeCon

Europe 2022

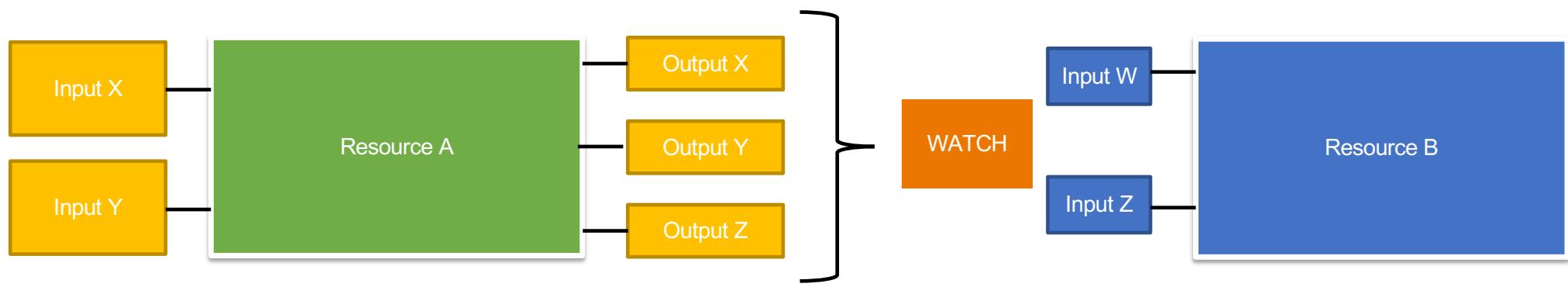


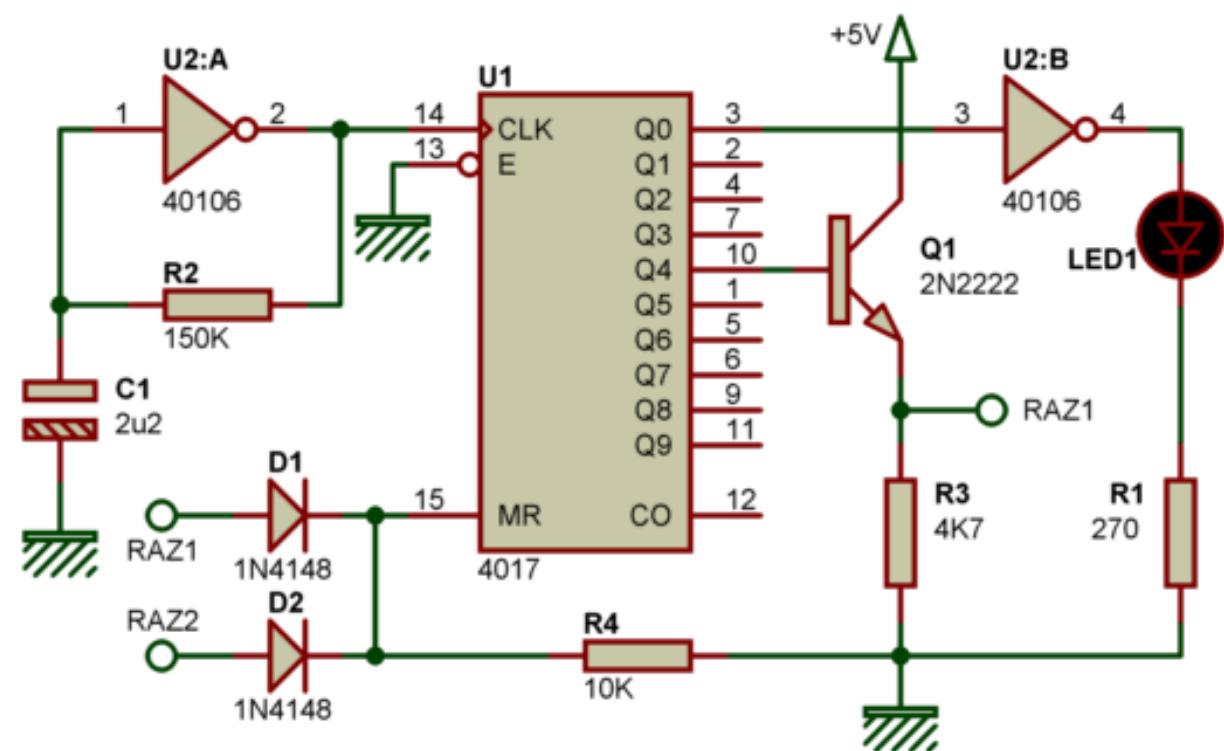
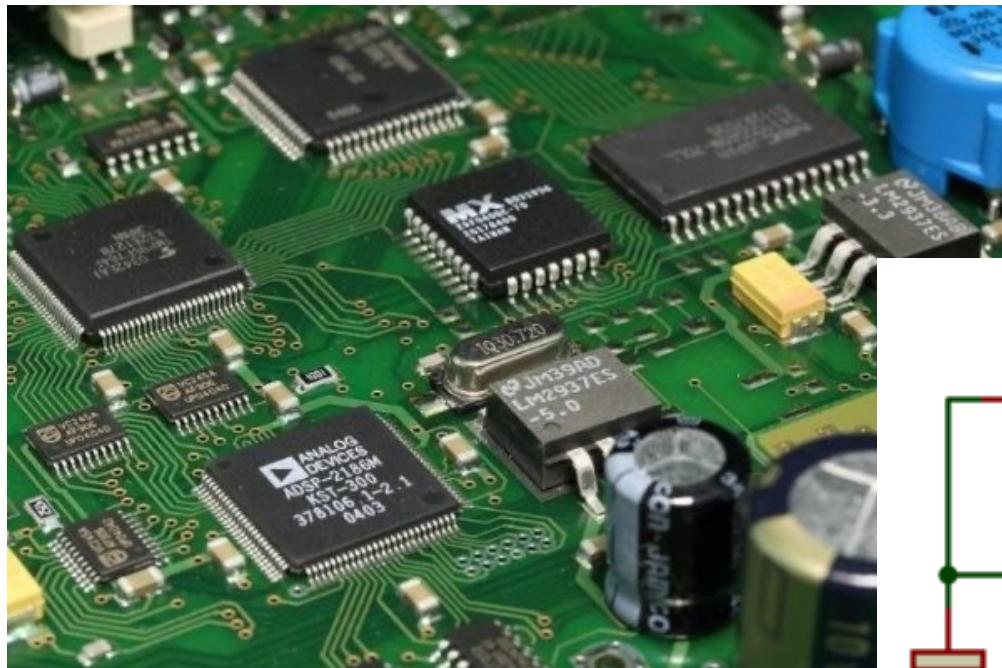
*Now you understand the name of this session ;-)*

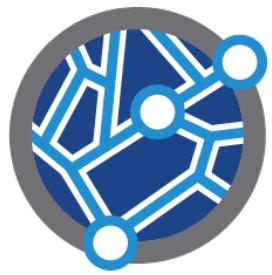
<https://www.lemagducine.fr/cinema/critiques-films/west-side-story-film-steven-spielberg-avis-10044963/>



# The choreography pattern



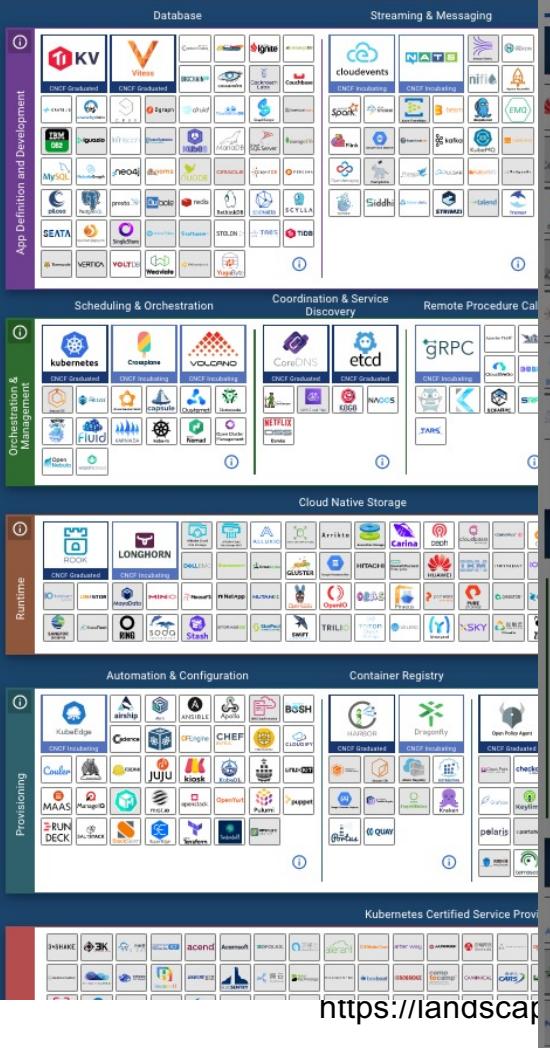




# CARTOGRAPHER

Supply Chain Choreographer

Landscape Card Mode Members Serverless Wasm



<https://landscape.cncf.io>

# Cartographer

VMware CNCF Platinum Member

App Definition and Development · Continuous Integration & Delivery

Cartographer is a Kubernetes-native Choreographer providing higher modularity and scalability for the software supply chain.

Website	<a href="http://cartographer.sh">cartographer.sh</a>
Repository	<a href="https://github.com/vmware-tanzu/cartographer">github.com/vmware-tanzu/cartographer</a> 360
Crunchbase	<a href="https://crunchbase.com/organization/vmware">crunchbase.com/organization/vmware</a>
LinkedIn	<a href="https://linkedin.com/company/vmware">linkedin.com/company/vmware</a>
Twitter	@OssCartographer  Latest Tweet <span style="float: right;">this week</span>
First Commit	8 months ago  Latest Commit <span style="float: right;">this week</span>
Contributors	26  Latest Release <span style="float: right;">last week</span>
Headquarters	Palo Alto, California  Headcount <span style="float: right;">10,001-1,000,000</span>
Market Cap	\$46.4B
Slack	<a href="https://kubernetes.slack.com/archives/C02HKPSEKV1">https://kubernetes.slack.com/archives/C02HKPSEKV1</a>

**CII Best Practices** passing

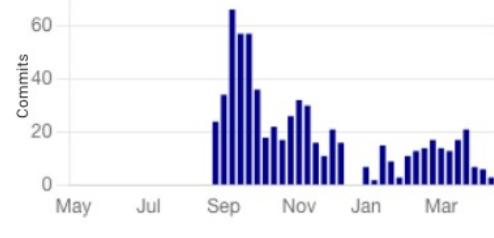
**License** Apache License 2.0

**Open Source Software**



Go 95%  
Shell 4%  
Makefile <1%  
Python <1%

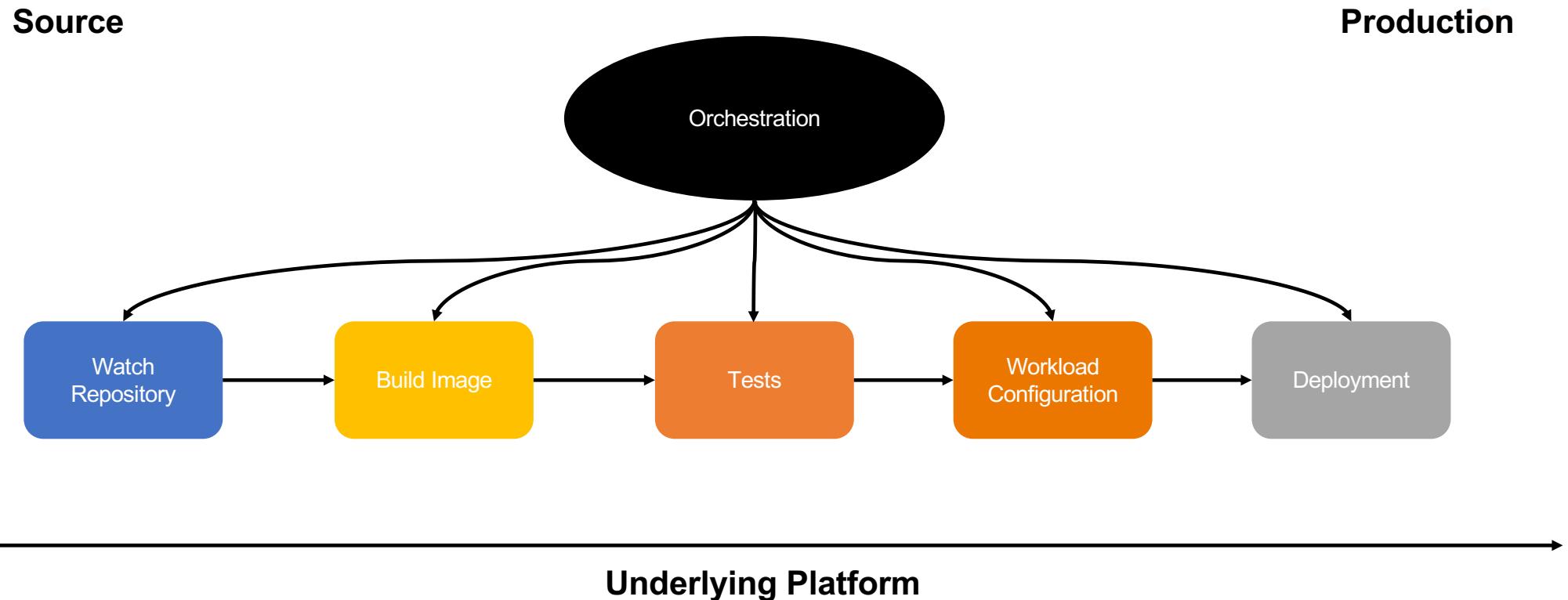
**Commits**



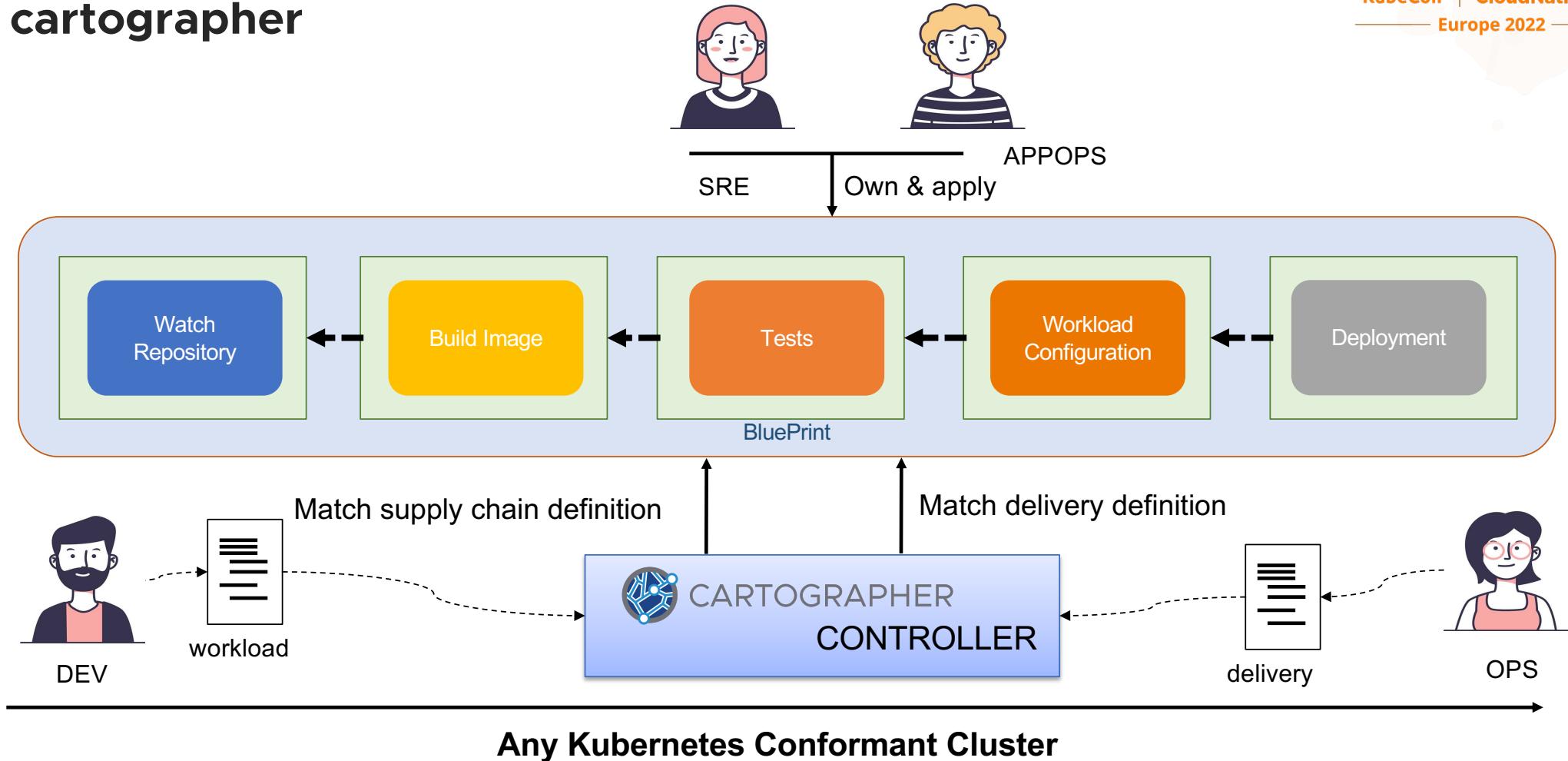
Tweets by @OssCartographer

**Cartographer**  
@OssCartographer

# Supply chain choreography with cartographer



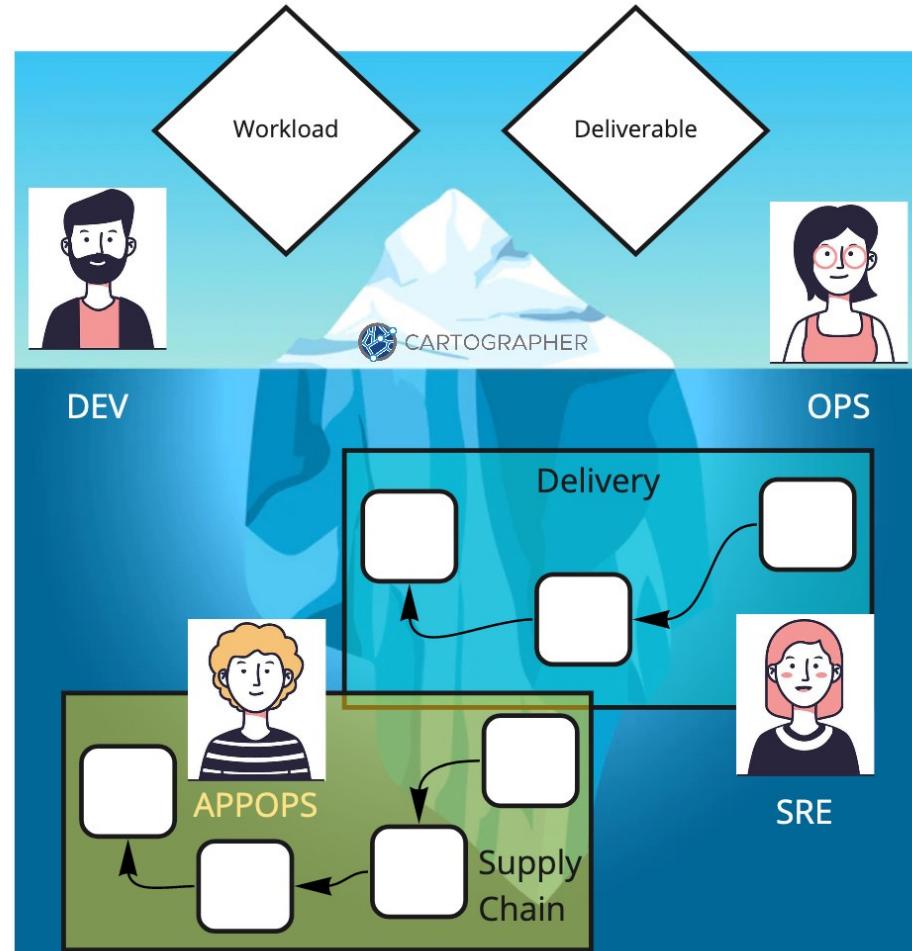
# Supply chain choreography with cartographer



# Hiding the complexity

As a Dev I need to integrate my code continuously  
 As an Ops I need to deploy my code continuously

As an AppOps, I offer Dev people Supply Chains as a service to manage continuous integration at scale.  
 As an SRE, I offer OPS people Deliveries as a service to manage continuous deployment at scale.



# Workload

Requested workload

Source code is here

Optional Parameters

```
apiVersion: carto.run/v1alpha1
kind: Workload
metadata:
  name: birds
  labels:
    app.tanzu.vmware.com/workload-type: micropet-service-git
spec:
  serviceAccountName: cartographer-workload-sa
  source:
    git:
      url: https://github.com/bmoussaud/micropets-app/
      ref:
        branch: master
  params:
    - name: mode
      value: "RANDOM_NUMBER"
    - name: port
      value: 8080
    - name: observability
      value: true
    - name: usedb
      value: true
```

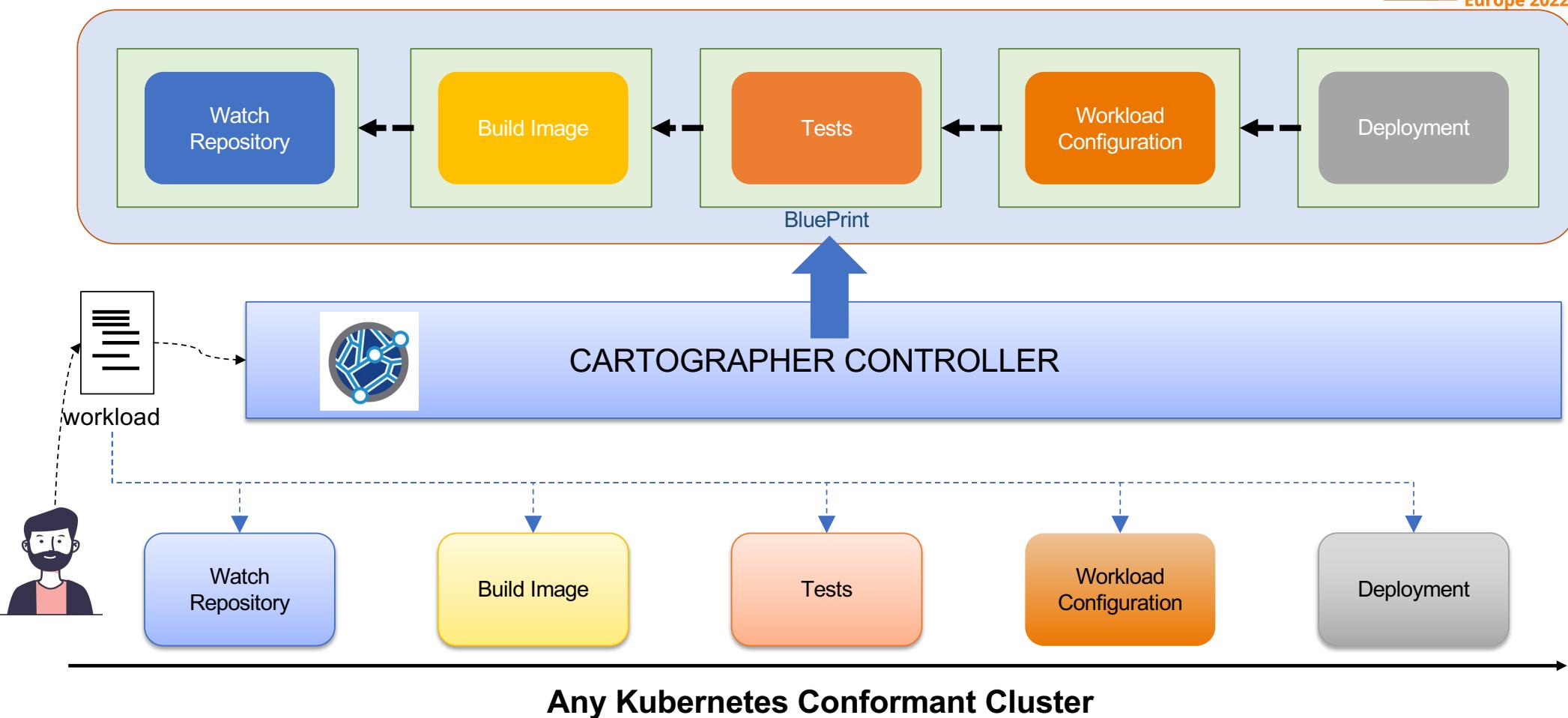


KubeCon  
Europe 2022

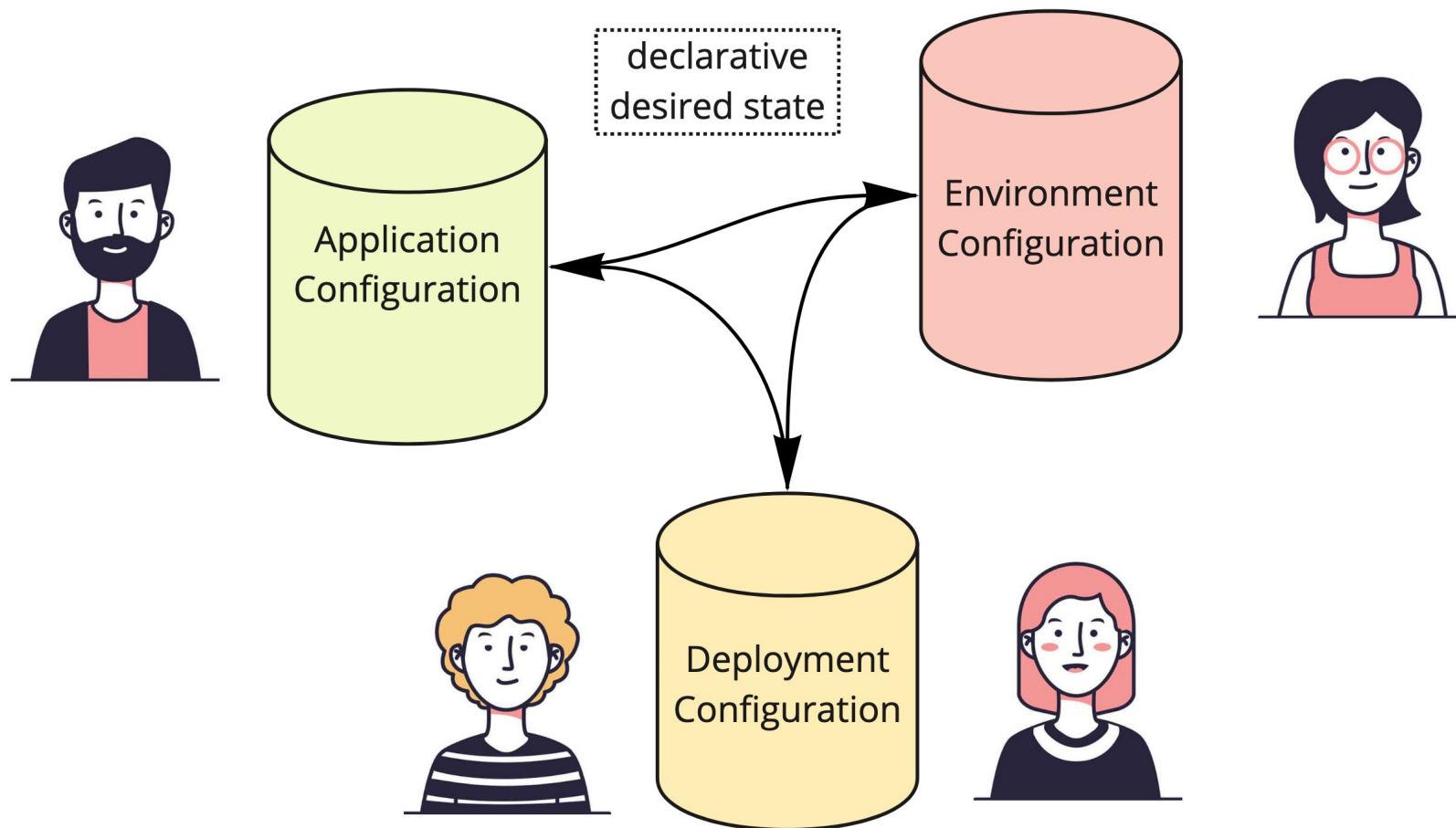


CloudNativeCon  
Europe 2022

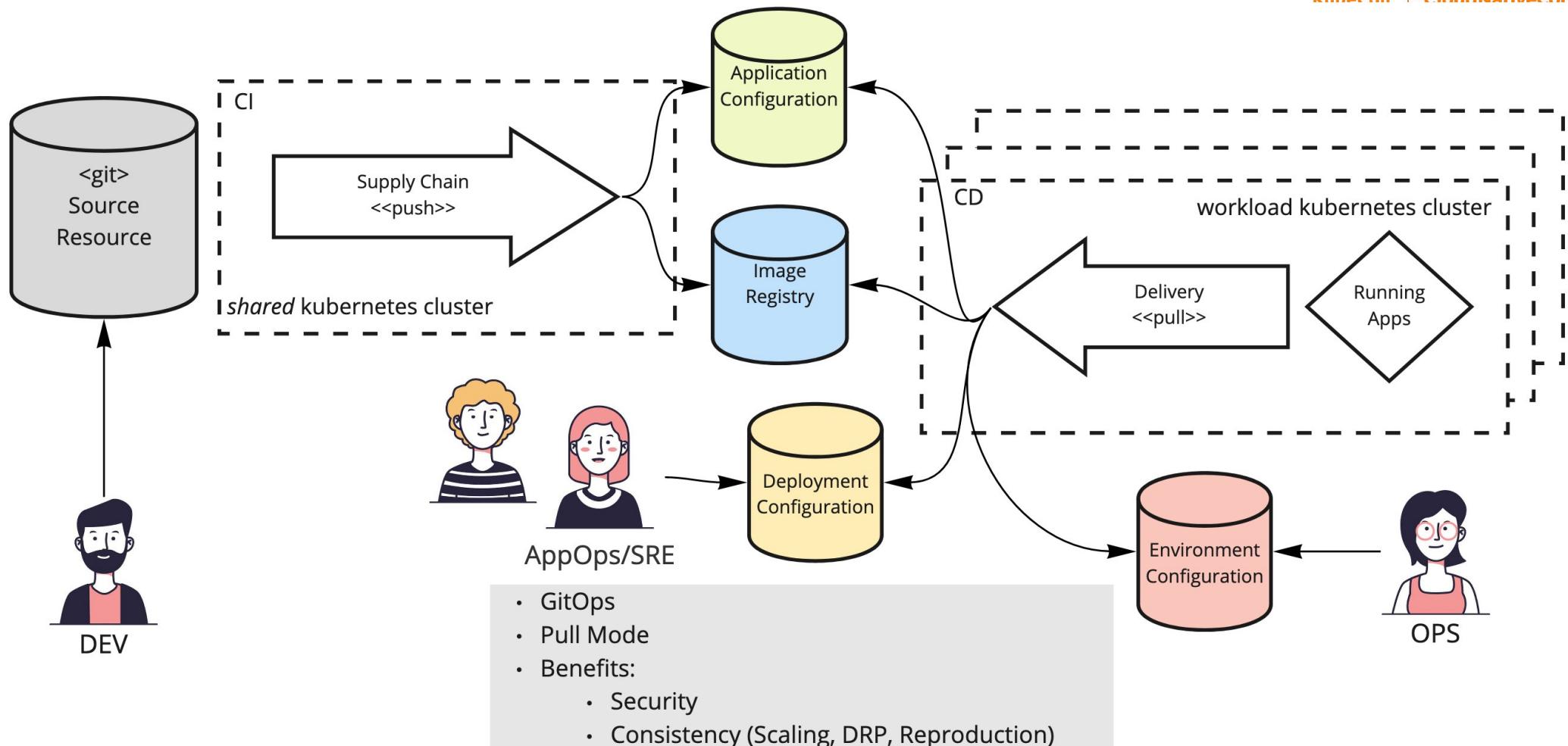
# New Workload



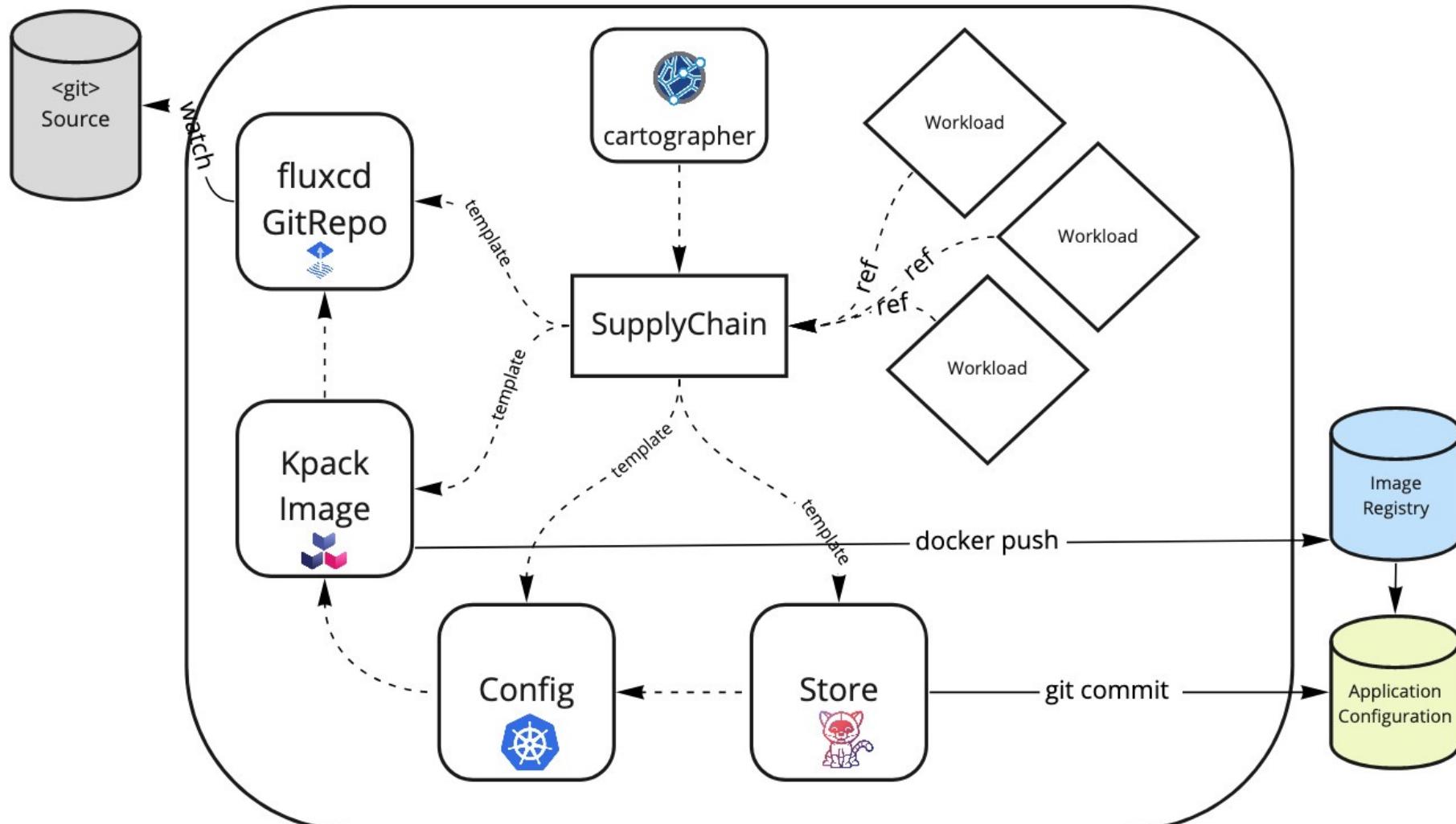
# What's a deployment ?



# CI/CD GitOps

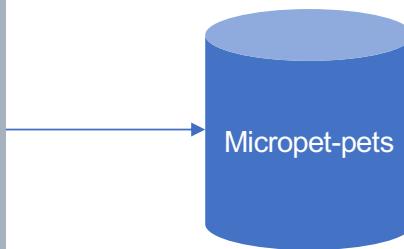


# Continuous Integration with Cartographer



```

apiVersion: source.toolkit.fluxcd.io/v1beta1
kind: GitRepository
metadata:
  name: micropet-pets
spec:
  interval: 1s
  ref:
    branch: master
  url: https://github.com/bmoussaud/micropets-app/
status:
  artifact:
    checksum: 2627a86c641a070a5ab10e6c15791bf5d412f5b7
    lastUpdateTime: "2022-05-04T16:42:40Z"
    path: gitrepository/micropets-supplychain/micropet-
pets/801430964884334ec3135f8e47779d3657c97e52.tar.gz
    revision: master/801430964884334ec3135f8e47779d3657c97e52
    url: http://source-controller.gitops-toolkit.svc.cluster.local./gitrepository/micropets-supplychain/
/micropet-pets/801430964884334ec3135f8e47779d3657c97e52.tar.gz
  
```



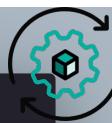
CARTOGRAPHER propagates

\$kubectl edit Image micropet-pets

CARTOGRAPHER is wire between the 2 resources

```

apiVersion: kpack.io/v1alpha2
kind: Image
metadata:
  name: micropet-pets
spec:
  build:
    env:
      - name: BP_IMAGE_LABELS
        value: watermark.project=micropet
    builder:
      kind: ClusterBuilder
      name: micropet-builder
      serviceAccountName: micropet-service-account
    source:
      blob:
        url: http://source-controller.gitops-toolkit.svc.cluster.local./gitrepository/micropets-
supplychain/micropet-pets/801430964884334ec3135f8e47779d3657c97e52.tar.gz
        subPath: pets
        successBuildHistoryLimit: 2
        tag: harbor.mytanu.xyz/library/micropet-tap-pets
  
```



# Workload in action



```
apiVersion: carto.run/v1alpha1
kind: Workload
metadata:
  name: birds
  labels:
    app.tanzu.vmware.com/workload-type: micropet-service-git
spec:
```

```
  serviceAccountName: cartographer-workload-sa
```

```
  source:
```

```
    git:          → kubectl tree workloads.carto.run birds
```

```
      url: https://NAMESPACE
```

```
      ref:      micropets-supplychain
```

```
      branch:  main/micropets-supplychain
```

```
  params:      micropets-supplychain
```

```
  - name: mode   micropets-supplychain
```

```
    value: "RANDOM"
```

```
  - name: port   micropets-supplychain
```

```
    value: 8080
```

```
  - name: observv  micropets-supplychain
```

```
    value: true
```

```
  - name: usedb  micropets-supplychain
```

```
    value: true
```

```
      NAME
```

```
      Workload/birds
```

```
      ConfigMap/birds
```

```
      GitRepository/micropet-birds
```

```
      Image/micropet-birds
```

```
      Build/micropet-birds-build-6
```

```
        Pod/micropet-birds-build-6-build-pod
```

```
      Build/micropet-birds-build-7
```

```
        Pod/micropet-birds-build-7-build-pod
```

```
      PersistentVolumeClaim/micropet-birds-cache
```

```
      SourceResolver/micropet-birds-source
```

```
      Runnable/birds-git-writer
```

```
      TaskRun/birds-git-writer-2mmh5
```

```
      TaskRun/birds-git-writer-kkj6z
```

```
        Pod/birds-git-writer-kkj6z-pod
```

```
      TaskRun/birds-git-writer-v8k6j
```

```
      READY
```

```
      True
```

```
      REASON
```

```
      Ready
```

```
      AGE
```

```
      20h
```

```
      19h
```

```
      20h
```

```
      20h
```

```
      36m
```

```
      36m
```

```
      19m
```

```
      19m
```

```
      20h
```

```
      20h
```

```
      19h
```

```
      18h
```

```
      18m
```

```
      18m
```

```
      12h
```



DEV



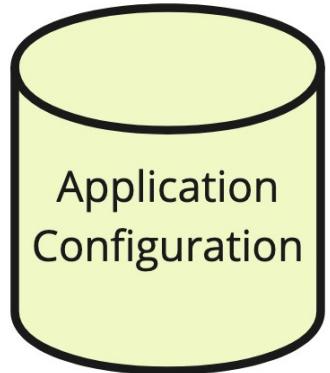
KubeCon



CloudNativeCon

Europe 2022

# Dev-Managed Configuration



```
● ● ●

applications
├── loader
│   ├── 001
│   │   └── resources.yaml
│   ├── 004
│   │   └── vegeta
│   │       └── resources.yaml
│   ├── 005
│   │   └── vegeta
│   │       └── resources.yaml
└── micropets
    ├── 20220329
    │   ├── cats
    │   │   └── values.yaml
    │   ├── dogs
    │   │   └── values.yaml
    │   ├── fishes
    │   │   └── values.yaml
    │   ├── gui
    │   │   └── values.yaml
    │   ├── pets
    │   │   └── values.yaml
    └── current
        ├── birds
        │   └── values.yaml
        ├── cats
        │   └── values.yaml
        ├── dogs
        │   └── values.yaml
        ├── fishes
        │   └── values.yaml
        ├── gui
        │   └── values.yaml
        └── pets
            └── values.yaml
```



DEV



KubeCon



CloudNativeCon

Europe 2022

# Dev-Managed Configuration



KubeCon

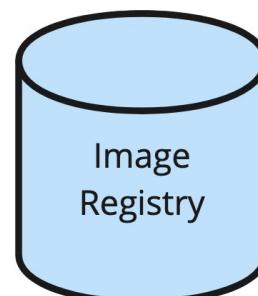
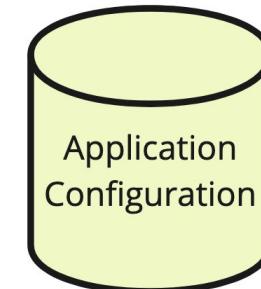


CloudNativeCon

Europe 2022

```
● ● ●  
#@data/values  
---  
application:  
  name: micro-pet  
service:  
  name: dogs  
  port: 7003  
  image: harbor.mytanzu.xyz/library/micropet-tap-dogs@sha256:305a204a48210c5fc87539522f20  
  config_path: /config  
  config_filename: pets_config.json  
  usedb: false
```

DEV



micropet-tap-dogs

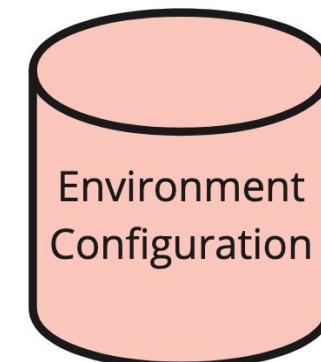
Info Artifacts

SCAN ACTIONS

	Artifacts	Pull Command	Tags	Size	Vulnerabilities	Annotations	Last Scan
<input type="checkbox"/>	sha256:cf325e89	<input type="button"/>	b2.2022	47.96MiB	<span>Not Scanned</span>		
<input type="checkbox"/>	sha256:c2339d8b	<input type="button"/>	b22.2022	48.21MiB	<span>Not Scanned</span>		
<input type="checkbox"/>	sha256:c0bfde9d	<input type="button"/>	b21.2022	47.95MiB	<span>Not Scanned</span>		
<input type="checkbox"/>	sha256:2b6030a5	<input type="button"/>	b20.2022	47.95MiB	<span>Not Scanned</span>		
<input type="checkbox"/>	sha256:79b82be4	<input type="button"/>	b19.2022	55.15MiB	<span>Not Scanned</span>		
<input type="checkbox"/>	sha256:329f4c6a	<input type="button"/>	b18.2022	47.95MiB	<span>Not Scanned</span>		

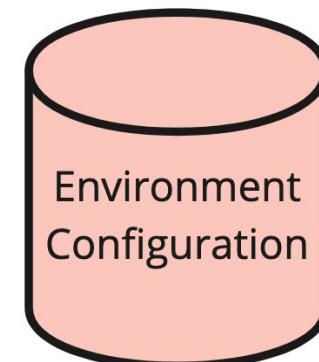
# Ops-Managed Configuration

```
environments
  aws
    aws-front
      └── environment-values.yaml
    aws-north
      └── environment-values.yaml
    tkg-europe-3
      └── environment-values.yaml
    tkg-uat-1
      └── environment-values.yaml
  azure
    aks-east-2
      └── environment-values.yaml
    bmoussaud-tap-demo
      └── environment-values.yaml
  vsphere
    c1
      └── environment-values.yaml
    c2
      └── environment-values.yaml
```



# Ops-Managed Configuration

```
#@data/values
---
# environment-values.yaml
service:
  backend:
    frequencyError: 10
  gui:
    color: green
    broker_service: http://micropet.north.mytanzu.xyz
environment:
  name: aws/north
  ingress: contour
  domain:
    internal: .micropets-supplychain.svc.cluster.local
    exposed: micropet.north.mytanzu.xyz
```



# SRE-Managed Configuration



KubeCon



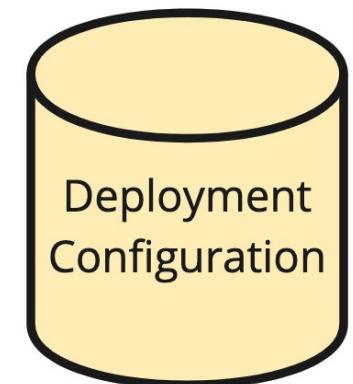
CloudNativeCon

Europe 2022

- SERVICE : generate K8S Resources: Deployment Service Ingress...
- KSERVICE: generate KNative Resource: Service
- NONE : Only Replace values

```
✓ kapp
  ✓ kservice
    ! kapp-config.yml
    ! resources.yaml
  ✓ none
    ≈ .keep
  ✓ service
    ! config-backend.yaml
    ! config-gui.yaml
    ≈ helpers.star
    ! istio.yaml
    ! resources.yaml
  > values
  > knative
  > kpack
  > repository
  ✓ scripts
```

```
30   matchLabels:
31     service: #@ data.values.service.name
32     k8s-app: #@ data.values.application.name
33     app.kubernetes.io/component: #@ data.values.service.name
34     app.kubernetes.io/part-of: #@ data.values.application.name
35   template:
36     metadata:
37       labels:
38         service: #@ data.values.service.name
39         k8s-app: #@ data.values.application.name
40         app.kubernetes.io/component: #@ data.values.service.name
41         app.kubernetes.io/part-of: #@ data.values.application.name
42   spec:
43     volumes:
44       - name: config-volume
45         configMap:
46           name: #@ data.values.service.name + "-config"
47           .....
```

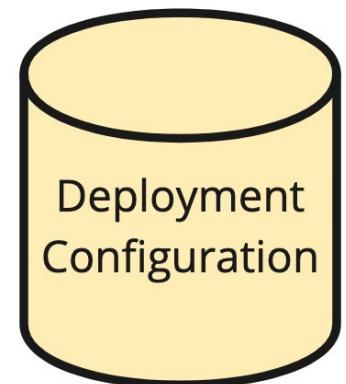


# SRE-Managed Configuration

```
#@ load("@ytt:data", "data")
#@yaml/text-templated-strings
#@ if/end data.values.service.name != "gui":
---
apiVersion: v1
kind: ConfigMap
metadata:
  annotations:
    kapp.k14s.io/versioned : ""
  name: #@ data.values.service.name + "-config"
  labels:
    app.kubernetes.io/component: #@ data.values.service.name
    app.kubernetes.io/part-of: #@ data.values.application.name
data:
  #@ if data.values.service.backend.observability:
  (@= data.values.service.config_filename @): |-
  {
    "service": {
      "port": ":(@= str(data.values.service.port) @)",
      "listen": "true",
      "mode": "(@= data.values.service.backend.mode @)",
      "frequencyError": (@= str(data.values.service.backend.frequencyError) @),
      "delay": {
        "period": 100,
        "amplitude": 0.3
      },
      "from": "(@= data.values.environment.name @)"
    },
    "observability": {
      "enable": true,
      "application": "micropets",
      "service": "(@= str(data.values.service.name) @)",
      "cluster": "(@= str(data.values.environment.name) @)"
    }
  }
  @ end
---
```



SRE



# Carvel KAppCtrl



- Carvel: one tool, a single responsibility
- **YTT**: to template and patch YAML files.

<https://carvel.dev/ytt/>

- **KAPP**: Deploy and manage groups of Kubernetes resources as "applications". Apply changes safely and predictably, watching resources as they converge.

<https://carvel.dev/kapp/>

- KAppCtrl = YTT + KAPP

```
● ● ●

apiVersion: kappctrl.k14s.io/v1alpha1
kind: App
metadata:
  name: dogs
spec:
  deploy:
    - kapp:
        rawOptions:
        - --diff-changes=true
    fetch:
      - http:
          subPath: kapp/service
          url: http://source-controller.gitops-toolkit.svc.cluster.local./gitrepository/micropets-supplychain
/deployment-dogs/942424e653ad4cba1c4bf45a526759db737fce66.tar.gz
        - http:
            subPath: applications/micropets/current/dogs
            url: http://source-controller.gitops-toolkit.svc.cluster.local./gitrepository/micropets-supplychain
/dogs-app/7bda569a64feca68f4cc85bbcc656f62441fc8d1.tar.gz
        - http:
            subPath: environments/aws/aws-front
            url: http://source-controller.gitops-toolkit.svc.cluster.local./gitrepository/micropets-supplychain
/dogs-env/7bda569a64feca68f4cc85bbcc656f62441fc8d1.tar.gz
      serviceAccountName: cartographer-deliverable-sa
      template:
        - ytt:
            ignoreUnknownComments: true
```

# Smoke Test

- Check the e2e availability of the service
- It's not because all the resources are deployed the service is up and running
- Personal Side Project – work in progress

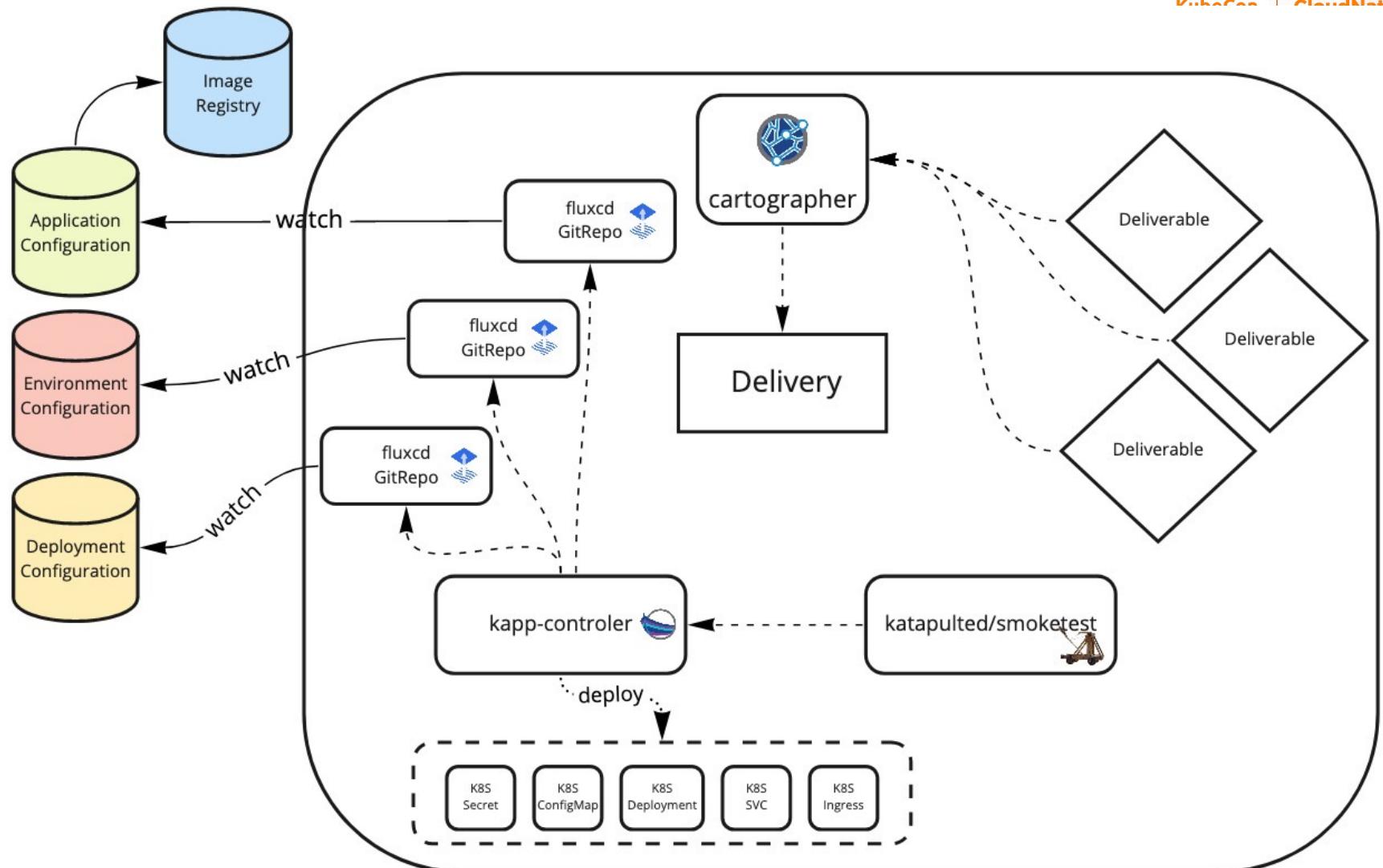


PromCon  
Latin America 2021

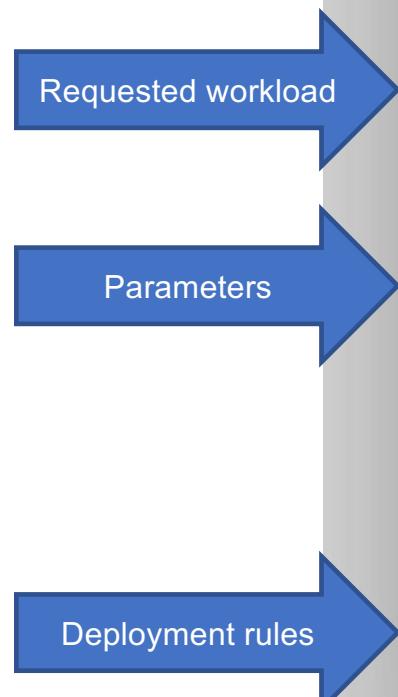


```
apiVersion: katapult.org/v1alpha1
kind: SmokeTest
metadata:
  name: dogs-smoke-test
spec:
  expectedResponseText: Total
  maxRetries: 10
  retryIntervalSeconds: 5
  showPageContent: true
  startDelaySeconds: 30
  url: http://front.mytanzu.xyz/dogs/v1/data?main/7bda569a64feca68f4cc85bbcc656f62441fc8d1
```

# Continuous Deployment with Cartographer



# Deliverable



```
apiVersion: carto.run/v1alpha1
kind: Deliverable
metadata:
  name: dogs
  labels:
    app.tanzu.vmware.com/workload-type: micropets-delivery-smoke
spec:
  params:
  - name: environment
    value: aws/aws-front
  - name: application
    value: micropets/current
  serviceAccountName: cartographer-deliverable-sa
  source:
    git:
      ref:
        branch: main
        url: http://github.com/bmoussaud/micropets-app-gitops.git
```



## Deliverable in action

```
apiVersion: carto.run/v1alpha1
kind: Deliverable
metadata:
  name: dogs
  labels:
    app.tanzu.vmware.com/workload-type: micropets-delivery-smoke
spec:
  params:
    - name: environment
      value: aws/aws-front
    - name: application
      value: micropets/current
```

NAMESPACE	NAME	READY	REASON	AGE
micropets-supplychain	Deliverable/ <b>dogs</b>	True	Ready	63s
micropets-supplychain	App/ <b>dogs</b>	-	-	61s
micropets-supplychain	GitRepository/ <b>deployment-dogs</b>	True	GitOperationSucceed	56s
micropets-supplychain	GitRepository/ <b>dogs-app</b>	True	GitOperationSucceed	56s
micropets-supplychain	GitRepository/ <b>dogs-env</b>	True	GitOperationSucceed	56s
micropets-supplychain	SmokeTest/ <b>dogs-smoke-test</b>	True	'http://front.mytanzu.xyz/dogs/v1/data' is available	61s
micropets-supplychain	Job/ <b>job-smoke-test-dogs-smoke-test</b>	-	-	60s
micropets-supplychain	Pod/ <b>job-smoke-test-dogs-smoke-test-blwxr</b>	False	PodCompleted	60s

# Wrap Up!

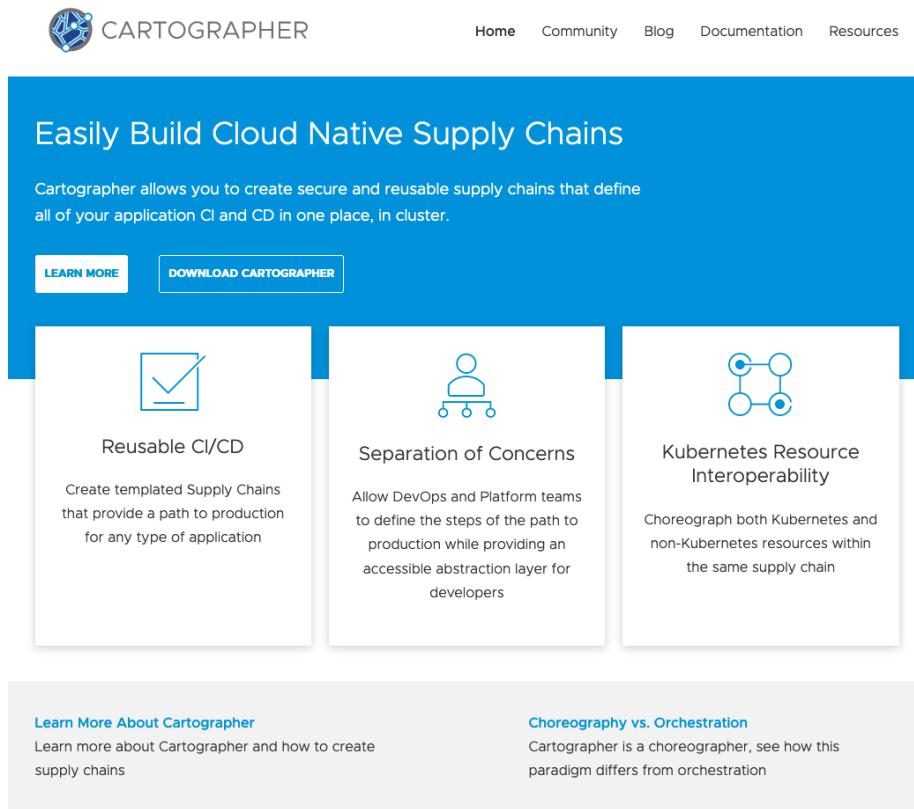
## Choreography with Cartographer

- An Opinionated Path to Production
- Hide the complexity
- Standard Contract
- Separation of concerns

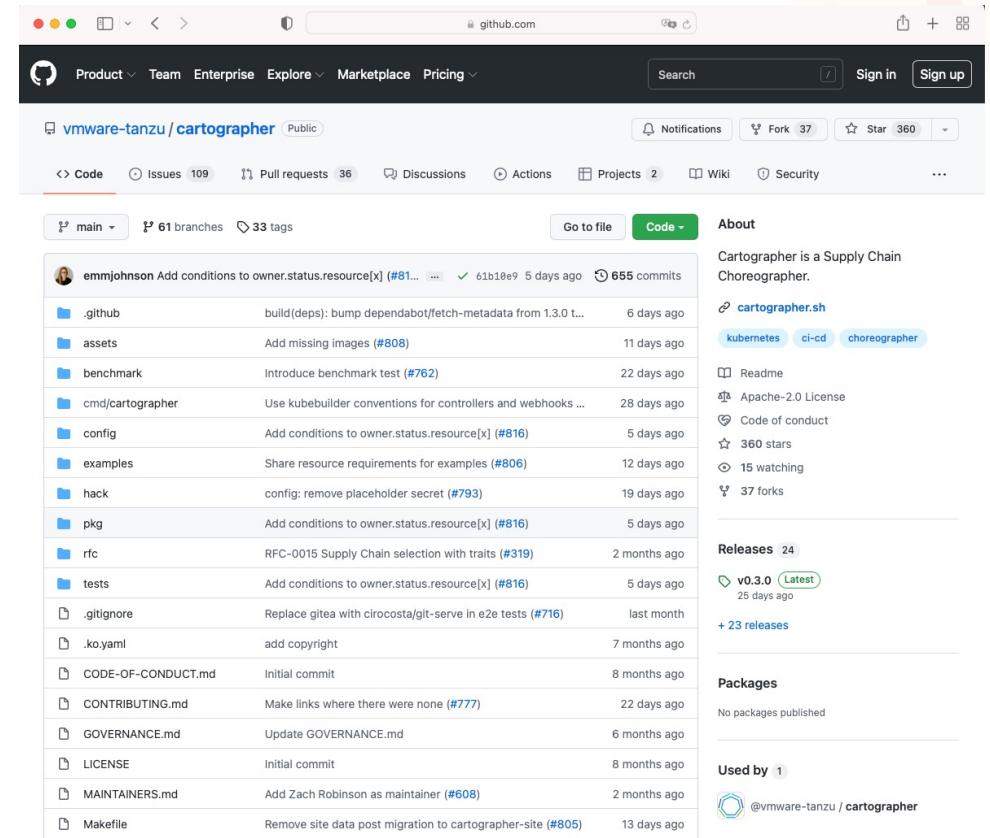


CARTOGRAPHER

# <http://cartographer.sh>



The screenshot shows the Cartographer homepage. At the top, there's a navigation bar with links for Home, Community, Blog, Documentation, and Resources. Below the navigation is a large blue header section with the title "Easily Build Cloud Native Supply Chains". A sub-section below it says "Cartographer allows you to create secure and reusable supply chains that define all of your application CI and CD in one place, in cluster." There are two buttons: "LEARN MORE" and "DOWNLOAD CARTOGRAPHER". Below these are three main features: "Reusable CI/CD" (with a checkmark icon), "Separation of Concerns" (with a person icon), and "Kubernetes Resource Interoperability" (with a network icon). Each feature has a brief description. At the bottom, there are sections for "Learn More About Cartographer" and "Choreography vs. Orchestration".



The screenshot shows the GitHub repository page for "vmware-tanzu / cartographer". The repository is public and has 109 issues, 36 pull requests, and 2 projects. The "Code" tab is selected, showing the main branch with 61 branches and 33 tags. The codebase includes files like .github, assets, benchmark, cmd/cartographer, config, examples, hack, pkg, rfc, tests, .gitignore, .ko.yaml, CODE-OF-CONDUCT.md, CONTRIBUTING.md, GOVERNANCE.md, LICENSE, MAINTAINERS.md, and Makefile. The repository has 360 stars, 15 watching, and 37 forks. It also has 24 releases, with the latest being v0.3.0. The "About" section describes Cartographer as a Supply Chain Choreographer and lists tags like kubernetes, ci-cd, choreographer. The "Used by" section shows one user, @vmware-tanzu.

## Resources

- Micropets: the Pet Application based on microservices  
 <https://github.com/bmoussaud/micropets-app>
- Micropets-gitops: the description of the applications & the environments  
 <https://github.com/bmoussaud/micropets-app-gitops>
- Micropets-app-operator provided the supply chains, the deliveries and the deployment rules  
 <https://github.com/bmoussaud/micropets-app-operator>
- Smoke-Test operator: the smoke test CRD (Work in Progress)  
 <https://github.com/bmoussaud/smoketest-operator>

# Give a try now



VMware Tanzu™  
Community Edition



Benoit Moussaud @bmoussaud · May 11

Having in a few minutes a [@VMwareTCE](#) Kubernetes Cluster running locally on my laptop with the new DockerDesktop is amazing! All the packages are ready to be installed. I think the 1st one will [@OssCartographer](#) ;-)

The screenshot shows the Docker desktop interface. On the left, there's a sidebar with a "VMware Tanzu Community Edition" section indicating "Cluster is running". Below it, under "LOGS" and "KUBECONFIG", there are several log entries related to the cluster setup. On the right, a detailed list of packages is shown with their descriptions:

DISPLAY-NAME	SHORT-DESCRIPTION
app-toolkit	App Toolkit package for TCE Kubernetes-native toolkit to support application lifecycle
cartographer	Reusable Cartographer blueprints
cert-injection-webhook	The Cert Injection webhook injects CA certificates and proxy environment variables into pods
cert-manager	Certificate management
contour	An Ingress controller
external-dns	This package provides DNS synchronization functionality
fluent-bit	Fluent Bit is a fast Log Processor and Forwarder
flux-source-controller	The source-controller is a Kubernetes operator, specialised in artifacts acquisition from external sources such as GitHub, GitLab, Artifactory, and more
gatekeeper	Policy management
grafana	Visualization and analytics software
harbor	OCI Registry
flux-helm-controller	The Helm Controller is a Kubernetes operator, allowing one to declaratively manage Helm chart releases with Kubernetes
httpproxy	HTTP proxy
informer-servic	Knative Serving builds on Kubernetes to support deploying and serving of applications and functions as serverless services
pack	Dependencies
pack-build	Builds application source code into OCI compliant images using Cloud Native Buildpacks
flux-autodiscover-controller	Knative controller is one of the components in GitOps toolkit
localpath-storage	This package provides local path node storage and primarily supports RBD Access mode
netlabel-cni	This package provides the ability for enabling attaching multiple network interfaces to pods in Kubernetes
prometheus	A time series database for your metrics
velero	Disaster recovery capabilities
whereabouts	A CNCF ISV plugin that assigns IP addresses cluster-wide

Comment 7 Likes 20 Share

<https://tanzucommunityedition.io/posts/tanzu-community-edition-011-adds-cartographer-supply-chain-choreography/>

Thank you and Merci

Q & A