

Application Delivery to Kubernetes

A 101 to a fast-evolving ecosystem

Max Körbächer

Introduction

Max Körbächer

Co-Founder & Manager Cloud Native Engineer, focusing on:

- Platform Engineering
- Application Delivery
- Cloud Native Advisory

Part of the Kubernetes Release Team & Release Engineering Team



CLOUD NATIVE

You are not cloud native just because you run on an CSP

- Scalable applications
- Running in **dynamic environments**
- Based containers or functions
- Utilizing **declarative APIs**
- Structured in (micro) services



**“The cloud isn’t a place,
it’s a way of doing IT” –
Michael Dell**

WHAT I MEAN

When I talk about cloud native and platforms, I think of



Platforms

We build platforms based on Kubernetes where application runs on.



Application Delivery

We implement tools to support the development and delivery of applications to the platform.



Automation

Everything is automated, tested and proved for reliability and zero-downtimes.



Declarative

We utilize APIs and declarative manifests to provision infrastructure, platform and delivery.



The Foundation

We build a trustful foundation for valuable solutions. This has to be reliable, secure and supporting the requirements of the applications.



Common patterns & problems with Platforms

Platforms abstracts
infrastructure complexities away.

BUT they create new unknown,
custom complexity:

- New responsibilities
- 100 options for one problem
- Single vs Multi Clusters
- CICD, GitOps or better something else
- How to build the application?
- How to ensure security, compliance and governance?

How to deliver software for K8s



The fairy tale of CI/CD

23 CI/CD Tools | journey of quality
journeyofquality.com

5 Reasons You Need to Consider...
walmartlabs.io

All you need to know about CI/CD...
dev.to

CI/CD - Behnam Ebrahimpour
ebrahimpour-b.ir

Getting Started in CI/CD for Beginners...
dev.to

Please stop guessing what DevOps...
blog.kintoohub.com

What Is CI/CD? What's Important ...
stackify.com

Assess Vulnerabilities, Misconfigurations...
blog.qualys.com

Tempest Blog: Our Process Driven...
tempest.house

Magenta Cloud CI/CD
magenta-cloud.pan-net.cloud

CI/CD คืออะไร? ช่วยทำให้ Developer...
blog.cloudhm.co.th

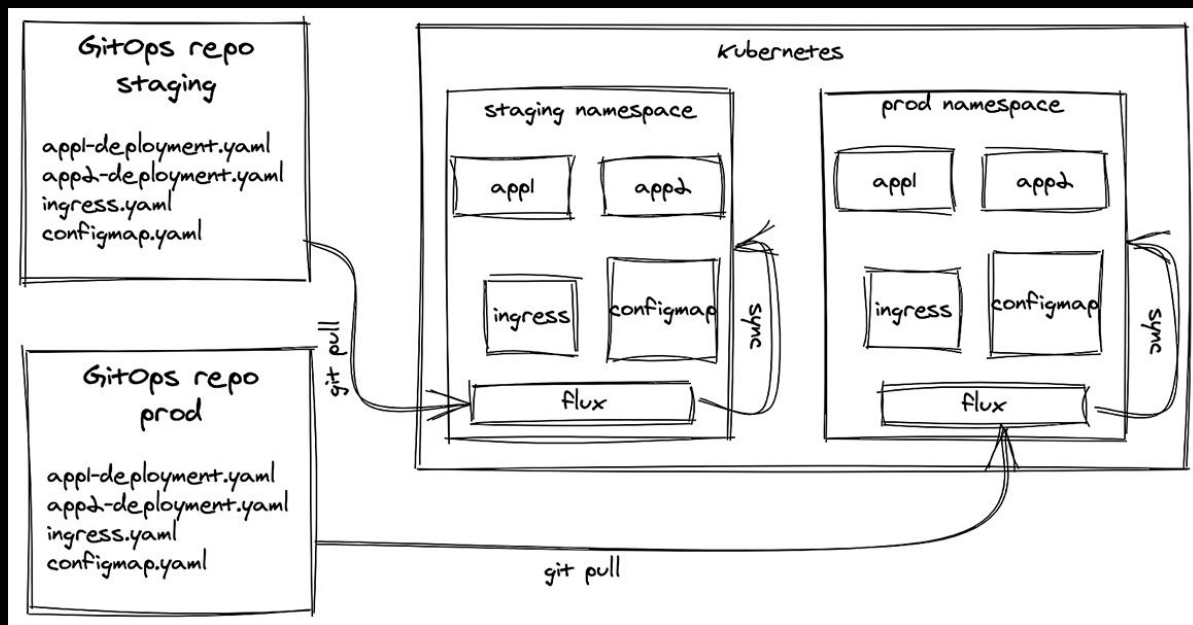
Ci Cd Clipart (#5198028) - PinCli...

- Specially custom build pipelines
- “Hand Made”
- Yet another script

That's not cloud native!



The fairy tale of ~~CI/CD~~ GitOps



GitOps only gives the answer to 50% of the story.

That's awesome, but not a 100% solution!

**None of it is a
perfect solution**



PERFECT

Where to find the right tools?

CNCF Cloud Native Landscape
2021-06-04T05:29:06Z de3c1f20

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

The landscape is organized into several functional categories:

- App Definition and Development:** Database, Streaming & Messaging, Application Definition & Image Build, Continuous Integration & Delivery.
- Orchestration & Management:** Scheduling & Orchestration, Coordination & Service Discovery, Remote Procedure Call, Service Proxy, API Gateway, Service Mesh.
- Runtime:** Cloud Native Storage, Container Runtime, Cloud Native Network.
- Provisioning:** Automation & Configuration, Container Registry, Security & Compliance, Key Management.
- Special:** Kubernetes Certified Service Provider, Kubernetes Training Partner.

Additional categories on the right side include:

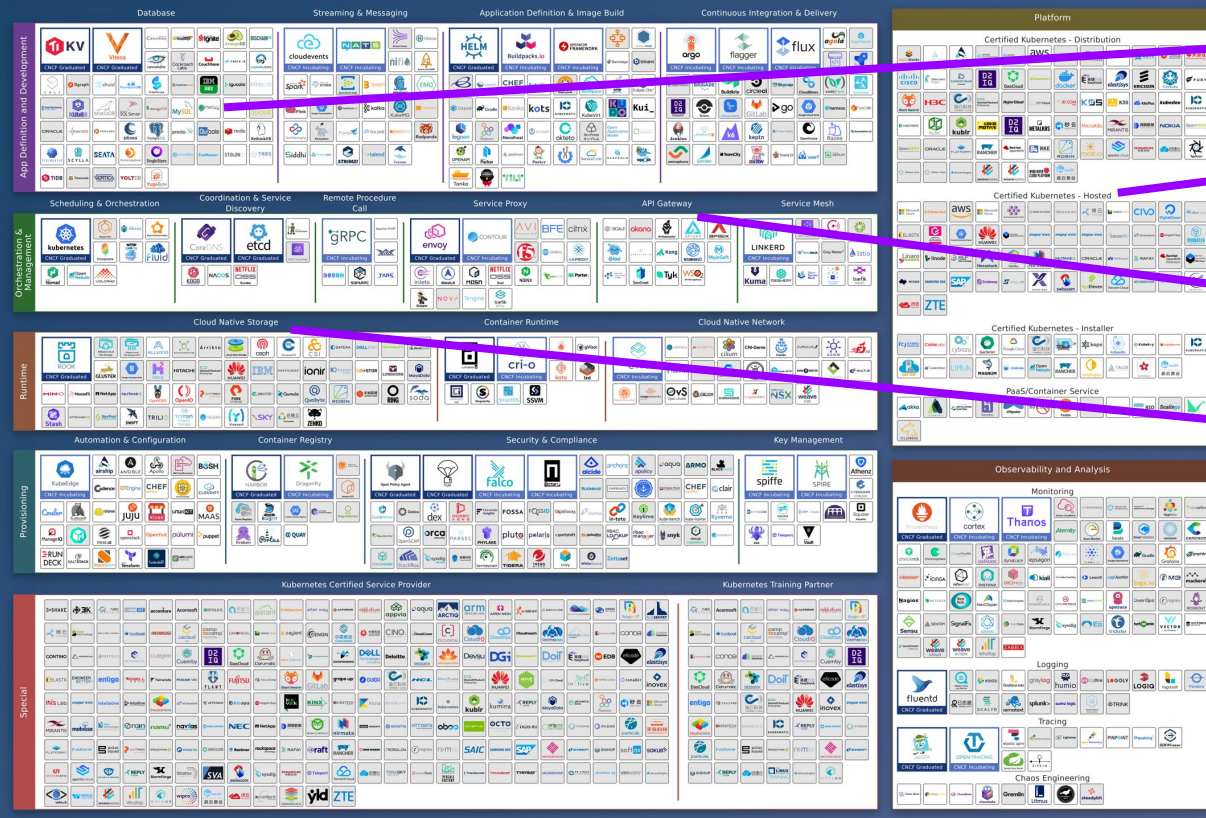
- Platform:** Certified Kubernetes - Distribution, Certified Kubernetes - Hosted, Certified Kubernetes - Installer, PaaS/Container Service.
- Observability and Analysis:** Monitoring, Logging, Tracing, Chaos Engineering.



Where to start to find the right tools?

CNCF Cloud Native Landscape
2021-06-04T05:29:06Z de3c1f20

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io



Databases

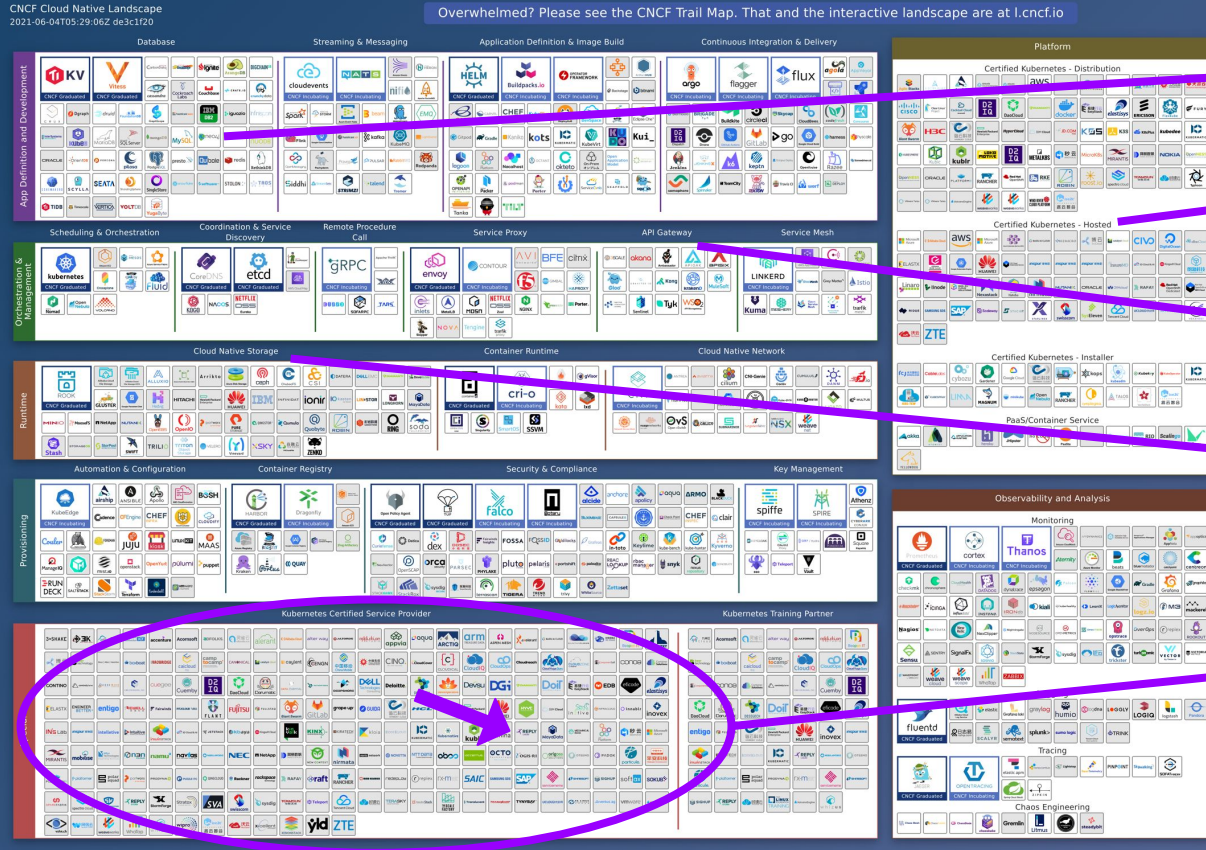
Hosted K8s

API Gateways

Storage



Where to start to find the right tools?



Databases

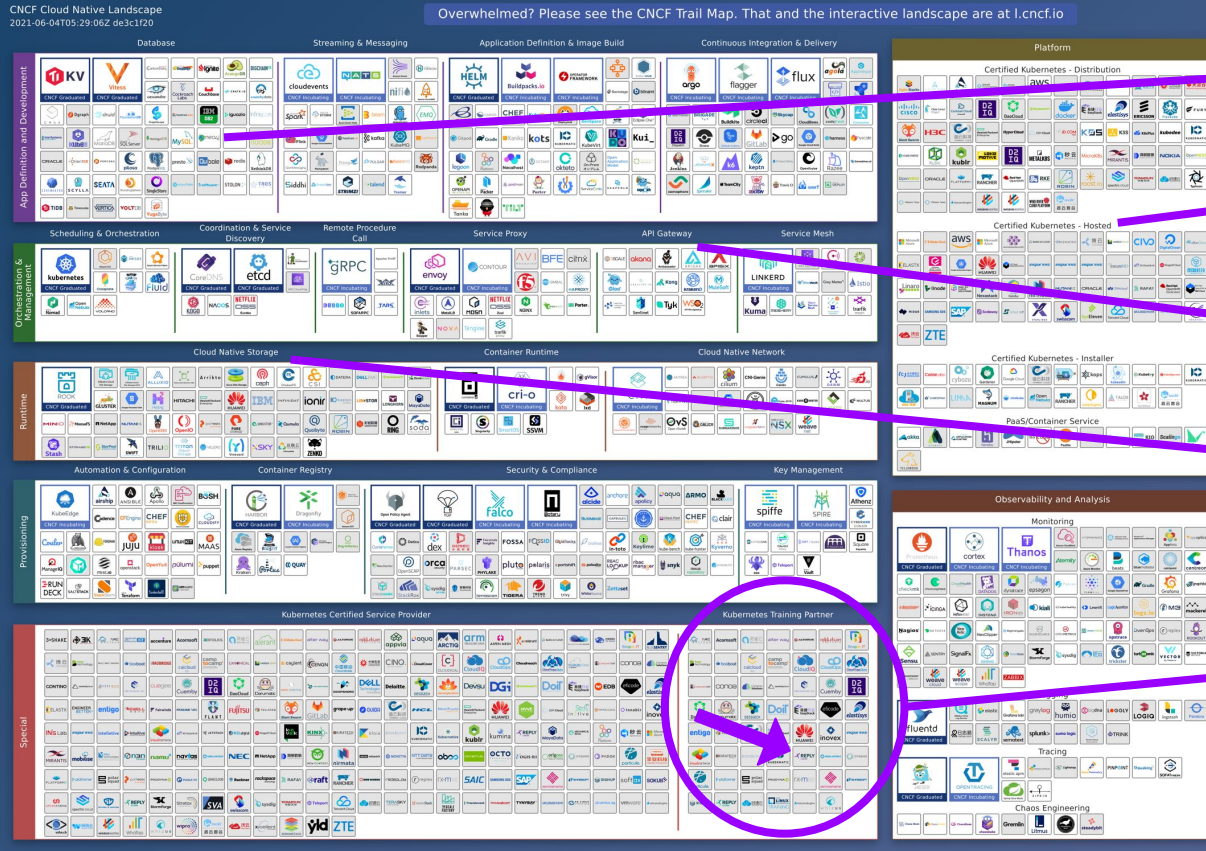
Hosted K8s

API Gateways

Storage



Where to start to find the right tools?



Databases

Hosted K8s

API Gateways

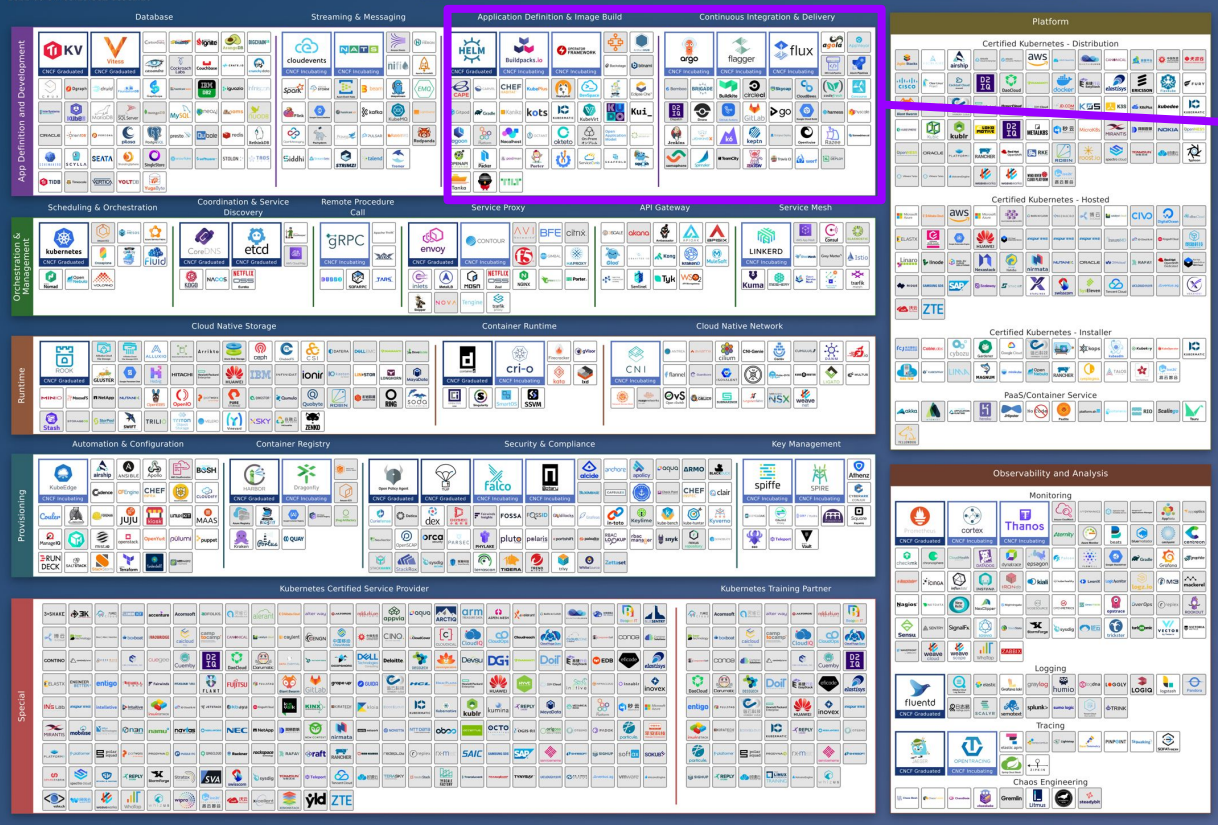
Storage



We want to focus

CNCF Cloud Native Landscape
2021-06-04T05:29:06Z de3c1120

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io



- Application Definition & Image Build
- Continuous Integration & Deliver (other than the standard CI/CD tools)

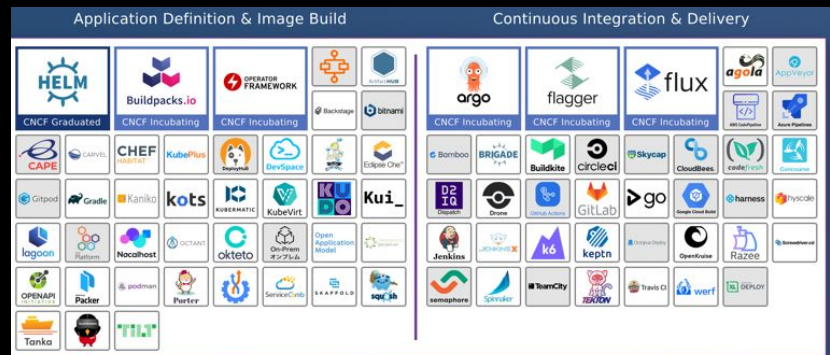


Let's focus on what is important

Application Definition & Delivery

Right now we ride a wave of complexity. The target systems getting highly complex, a simple executable is not enough to run and responsibilities getting newly sorted.

“Application Definition & Delivery”, ADD or short Application Delivery is a part of the platform engineering which is developer focused and try to support as good as possible their mission:

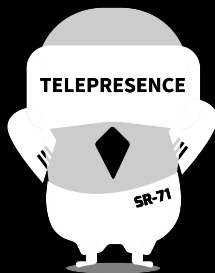


**NOT to learn 3x Cloud
Provider, K8s, Helm, min. 5
possible sidecar injections
and fixing your CI/CD every
2 days**





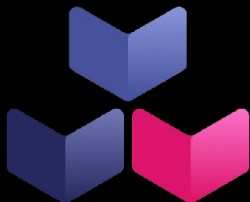
KubeVela



keptn



TILT



BuildPacks

```
pack build kotlin-sample \
--builder cnbs/builder:bionic \
--buildpack samples/kotlin \
--path samples/apps/kotlin/
```

```
# Buildpack API version
api = "0.2"

# Buildpack ID and metadata
[buildpack]
id = "samples/kotlin-gradle"
version = "0.0.1"
name = "Sample Kotlin Gradle
Buildpack"

homepage =
"https://github.com/buildpacks/samp
les/tree/main/buildpacks/kotlin-gra
dle"

# Stacks that the buildpack will
work with

[[stacks]]
id =
"io.buildpacks.samples.stacks.bioni
c"

[[stacks]]
id =
"io.buildpacks.samples.stacks.alpin
e"
```

```
<Main.kt>
package org.kotlinlang.play

fun main() {
    println("Hello, World!")
}
```

```
<Stack Dockerfile>
FROM ubuntu:bionic as base

ENV CNB_USER_ID=1000

ENV CNB_GROUP_ID=1000

ENV
CNB_STACK_ID="io.buildpacks.samples.sta
cks.bionic"

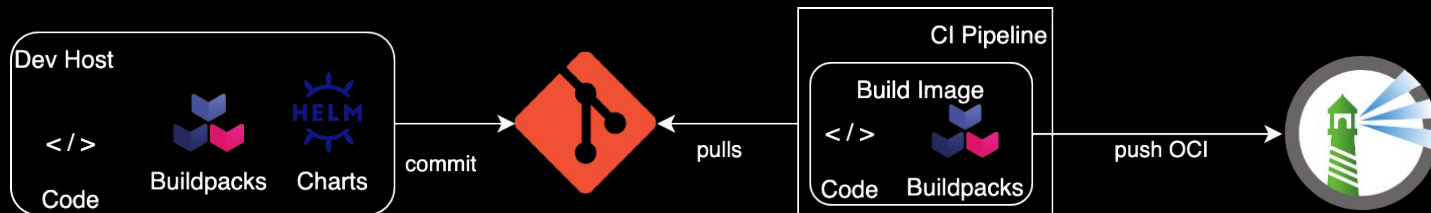
LABEL
io.buildpacks.stack.id="io.buildpacks.s
amples.stacks.bionic"

RUN groupadd cnb --gid ${CNB_GROUP_ID}
&& \
    useradd --uid ${CNB_USER_ID} --gid
${CNB_GROUP_ID} -m -s /bin/bash cnb
```





How to integrate BuildPacks in your daily doing?



- Require a custom place to run the build
- Someone has to specify, test and develop the stacks
- Feels like chaining up again tons of tools
- Theoretically can be “replaced” by Dockerfile & build
- Still we have to talk about the deployment configuration



Are CNBs better than the rest?

At least you should have a look at

	 Cloud Native Buildpacks	 Dockerfile	 source-to-image (s2i)	 Jib <small>Containerize your Java application.</small>	 ko
Advanced Caching	Yes	No	Yes	No	No
Auto-detection	Yes	No	Yes	Yes	Yes
Bill-of-Materials	Yes	No	No	No	No
Modular / Pluggable	Yes	No	No	N/A [†]	N/A [†]
Multi-language	Yes	Yes	Yes	No	No
Multi-process	Yes	No	No	No	No
Minimal app image	Yes	Yes [♦]	Yes [‡]	Yes	Yes
Rebasing	Yes	No	No	No	No
Reproducibility	Yes	No	No	Yes	Yes
Reusability	Yes	No	Yes	N/A [†]	N/A [†]





Tilt All Resources RESOURCES 1 0 1 2/3 v0.20.9

Alerts on Top

Filter resources by name

RESOURCES

- (Tiltfile) <30s ago Unknown
- example-python Updating...
- deploy <30s ago Completed in 0.4s

localhost:8000/

All Levels Errors (0) Warnings (0) Filter by text or /regexp/ Clear Logs

ARG flask_env=production

ENV FLASK_ENV \$flask_env

WORKDIR /app

ADD requirements.txt .
RUN pip install -r requirements.txt

ADD . .

ENTRYPOINT ["python", "/app/app.py"]

Tarring context_
Building image
copy /context / [done: 118ms]
[1/5] FROM docker.io/library/python:3.6@sha256:262365a100d18cf69958be545c9bafbe85f31974c383364254f1ab34cffb0ebd
[2/5] WORKDIR /app [cached]
[3/5] ADD requirements.txt . [cached]
[4/5] RUN pip install -r requirements.txt [cached]
[5/5] ADD . . [done: 32ms]
exporting to image [done: 41ms]

STEP 2/3 - Pushing example-python-image:tilt-e0d04924d2906ba4
Loading image to KIND
Image: "docker.io/library/example-python-image:tilt-e0d04924d2906ba4" with ID "sha256:e0d04924d2906ba418c244cc38c4600677dd50d9ea314be06b0776ce69919db6" not yet present on node "kind-control-plane", loading...

Tiltfile

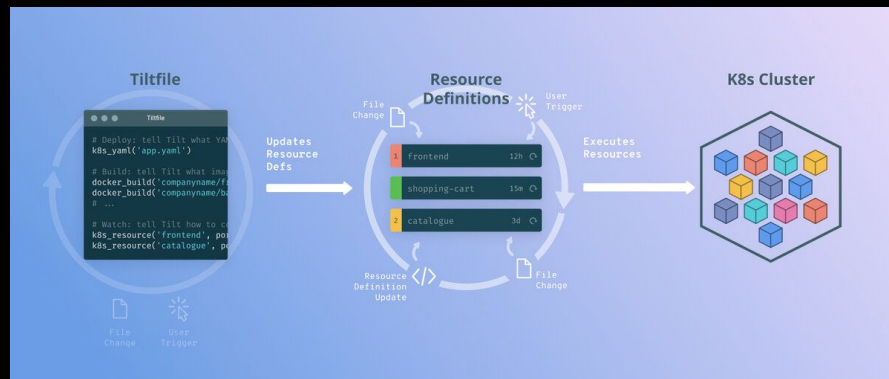
```
local_resource(  
    'deploy',  
    'python now.py >  
start-time.txt',  
)  
docker_build('example-python-image'  
, '.')  
k8s_yaml('kubernetes.yaml')  
k8s_resource('example-python',  
port_forwards=8000,  
resource_deps=['deploy'])
```

Demo



Tilt for teams (?!)

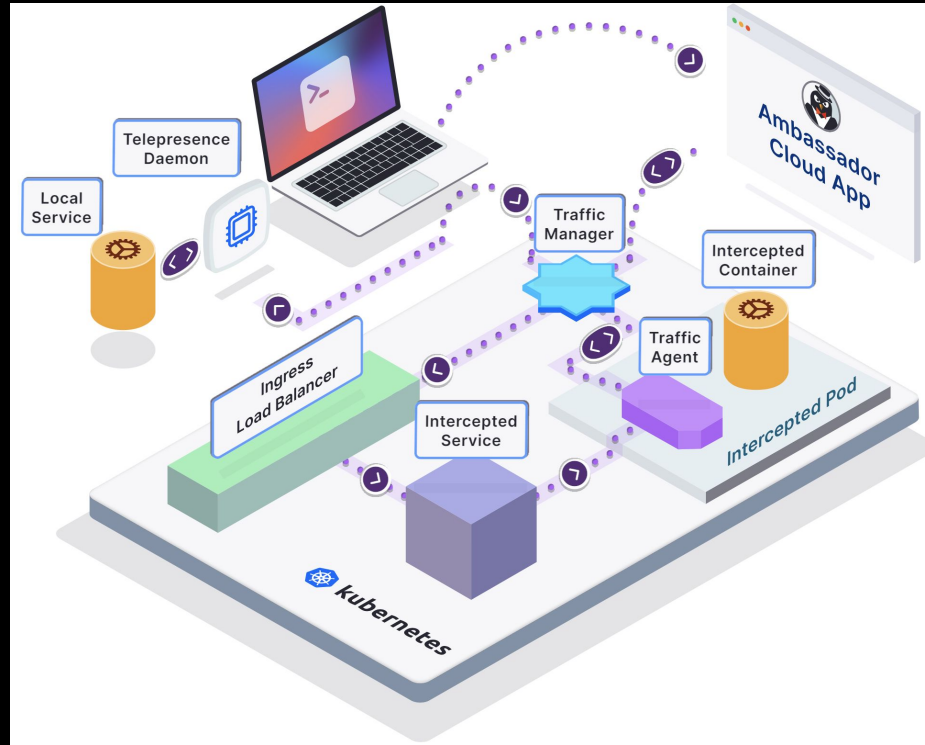
- Still requires docker running locally
- Focuses on K8s
- To interact with your team effectively, you need to use tilt cloud





Telepresence

Remote interception and debugging



Intercepts remote workload and allows to use your local tools to debug.

Bypass the deployment cycle to test fixes.

Only requires the resources of the services you test, no local k8s needed!

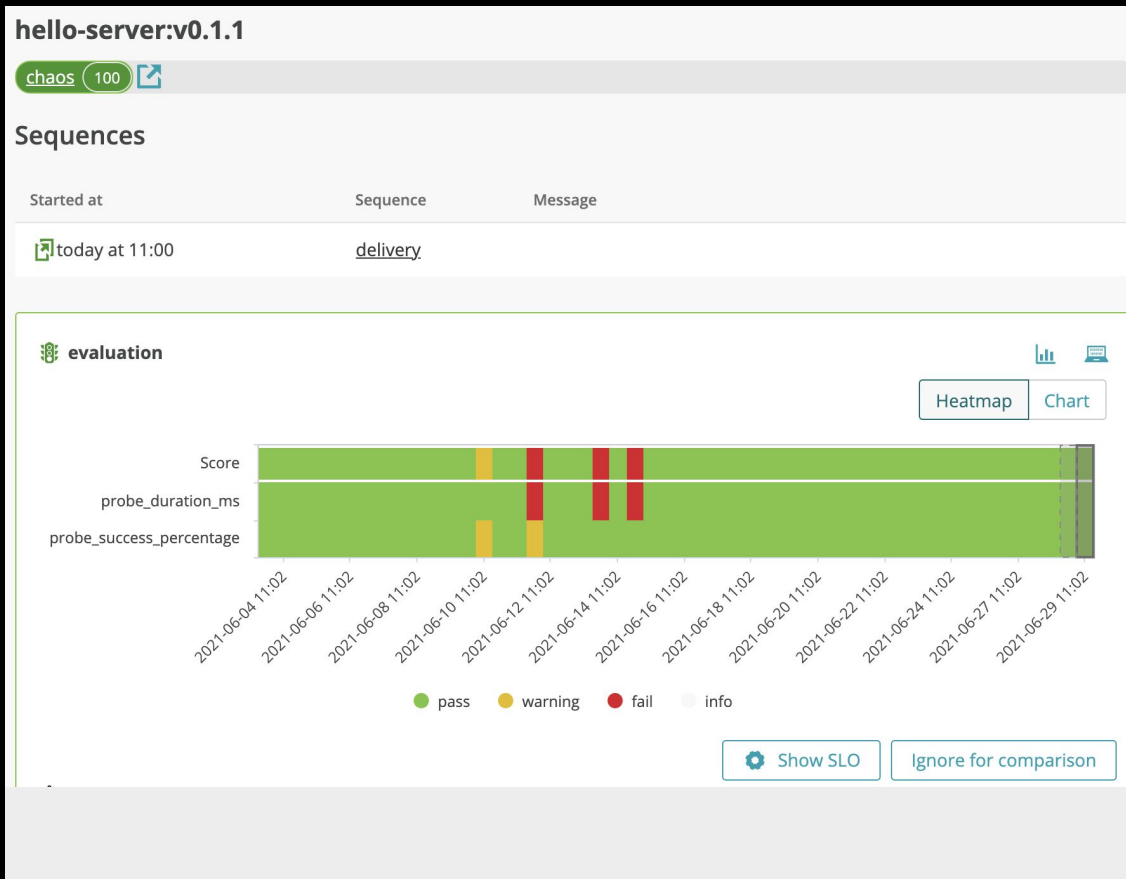


Keptn

- Support you in a cloud-native application life-cycle
- Integrates with observability and alerting to provide SLOs
- All declarative
- Closed-loop remediation

Live Demo System:

<https://keptn.public.demo.keptn.sh/>



Keptn

- Walks you through the deployment lifecycle
- Integrates:
 - CI/CD Systems
 - Observability Platforms
 - Testing Tools
 - Notification, further automation etc.

A SRE tool for developer?

The screenshot displays the Keptn dashboard for a project named 'sockshop'. The interface is divided into three main sections: '3 Stages', 'staging', and 'carts'.

3 Stages: This section shows three deployment stages: 'dev', 'staging', and 'production'. Each stage has a summary bar with icons for problems, quality gates, and services out-of-sync. Below each stage, there are two service cards: 'carts' and 'mongo'. Each card shows the service name, version, and deployment status (e.g., 'delivery' or 'delivery-direc').

staging: This section provides a detailed view of the 'staging' stage. It shows a summary bar with 0 Problems open, 1 Quality gate failed, and 0 Services out-of-sync. Below this, the 'carts' service is highlighted, showing a failed quality gate for version 'carts:0.11.2'. The message states: 'Quality gate of carts:0.11.2 failed with a score of 0. Rollback to carts:0.11.1 performed.' A 'View evaluation result' button is visible.

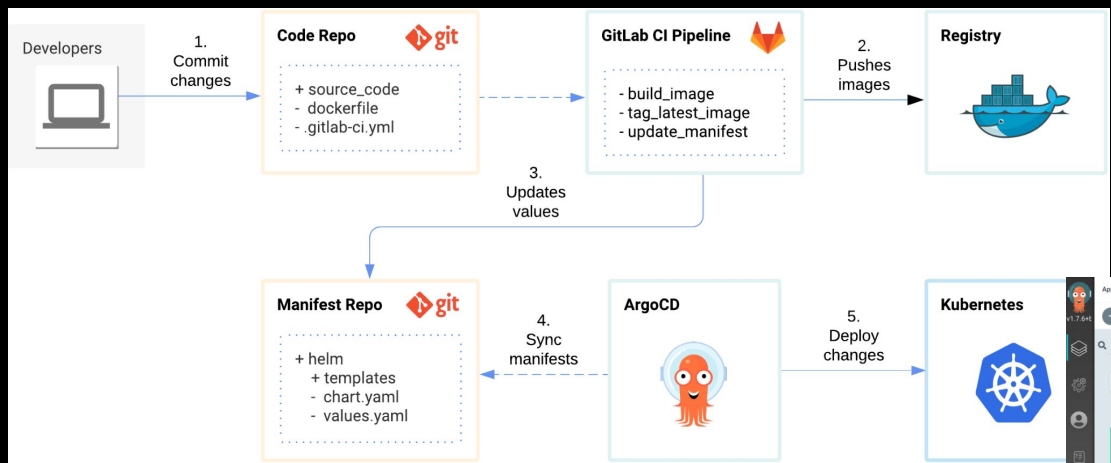
carts: This section shows the 'carts' service details, including the version 'carts:0.11.2' and the message: 'No pending deployments for this stage available.'





ArgoCD

GitOps with dev feedback



Demo

Application	Project	Status	Repository	Target Re.	Path	Destination	Namespace
argo-rollouts	production	healthy Synced	https://github.com/argoproj/argo-helm	HEAD	charts/argo-rollouts	https://kubernetes.default.svc	argo-rollouts
devops-garadox	production	healthy Synced	https://github.com/farco/devops-parado	HEAD	helm	https://kubernetes.default.svc	production
devops-toolkit	production	healthy Synced	https://github.com/farco/devops-toolkit	HEAD	helm	https://kubernetes.default.svc	production
production	production	Progressing Synced	https://github.com/devopsgaradox/argocd	HEAD	helm	https://kubernetes.default.svc	production
prometheus	production	Progressing Synced	https://github.com/prometheus-community	HEAD	charts/prometheus	https://kubernetes.default.svc	monitoring



Why ArgoCD is great?

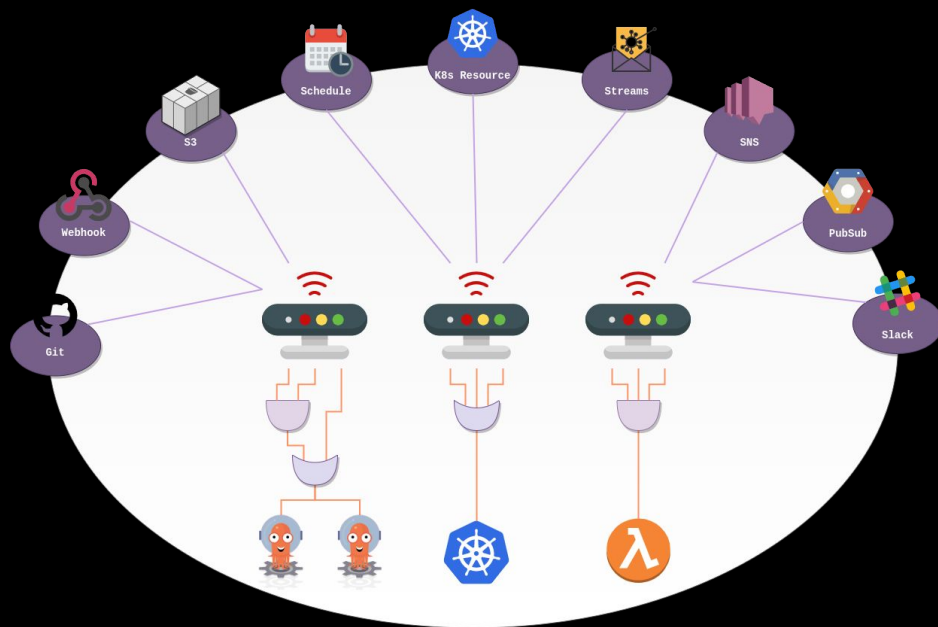
It keeps evolving!

The screenshot shows the Argo Rollouts interface for a 'canary-demo' deployment. At the top, there are navigation buttons: RESTART, RETRY, ABORT, and PROMOTE-FULL. The main content is divided into several sections:

- Summary:** Shows the deployment strategy as 'Canary', the current step '1/8', and the set weight '20'.
- Containers:** Lists the container 'argoproj/rollouts-demo:green' and includes a button to 'Add more containers to fill this space!'.
- Revisions:** Displays two revisions. Revision 9 is the current 'canary' revision, marked with a green checkmark. Revision 8 is the previous 'yellow' revision, with a 'ROLLBACK' button.
- Steps:** A vertical list of deployment steps: 'Set Weight: 20%' (highlighted in green), 'Pause', 'Set Weight: 40%', 'Pause: 10s', and 'Set Weight: 60%'.

Argo Rollout

Progressive Delivery and fine-grained deployments of you apps



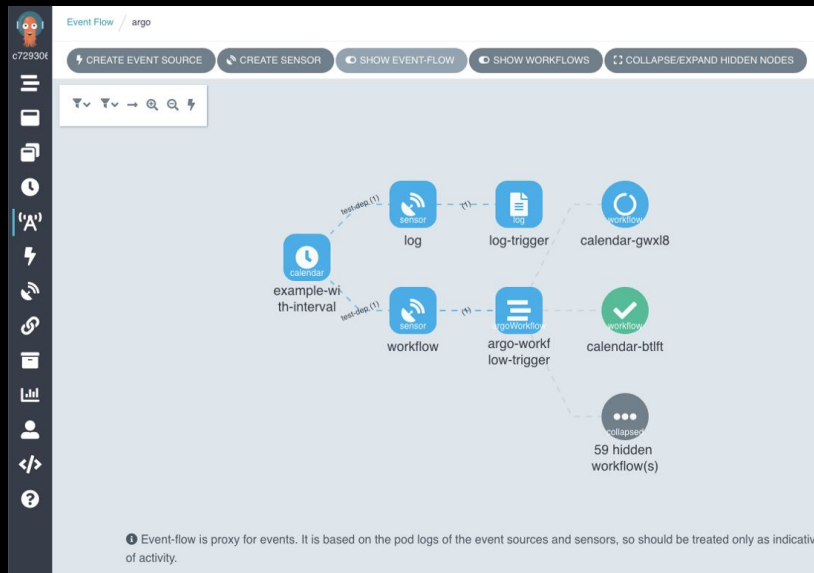
Argo Events

An event platform to trigger cloud native resources



Why ArgoCD is great? - Part 2

It keeps evolving!



Argo Workflow

Workflow engine to parallelize and orchestrate jobs on Kubernetes

```
1 apiVersion: argoproj.io/v1alpha1
2 kind: ApplicationSet
3 metadata:
4   name: guestbook
5 spec:
6   generators:
7     - list:
8       # Parameters are generated based on this cluster list, to be substituted
9       # into the template below.
10      elements:
11        - cluster: engineering-dev
12          url: https://1.2.3.4
13        - cluster: engineering-prod
14          url: https://2.4.6.8
15        - cluster: finance-preprod
16          url: https://9.8.7.6
17   template:
18     # An Argo CD Application template, with support for parameter substitution
19     # with values from parameters generated above.
20     metadata:
21       name: '{{cluster}}-guestbook'
22     spec:
23       project: default
24       source:
25         repoURL: https://github.com/argoproj-labs/applicationset.git
26         targetRevision: HEAD
27         path: examples/list-generator/guestbook/{{cluster}}
28       destination:
29         server: '{{url}}'
30         namespace: guestbook
```

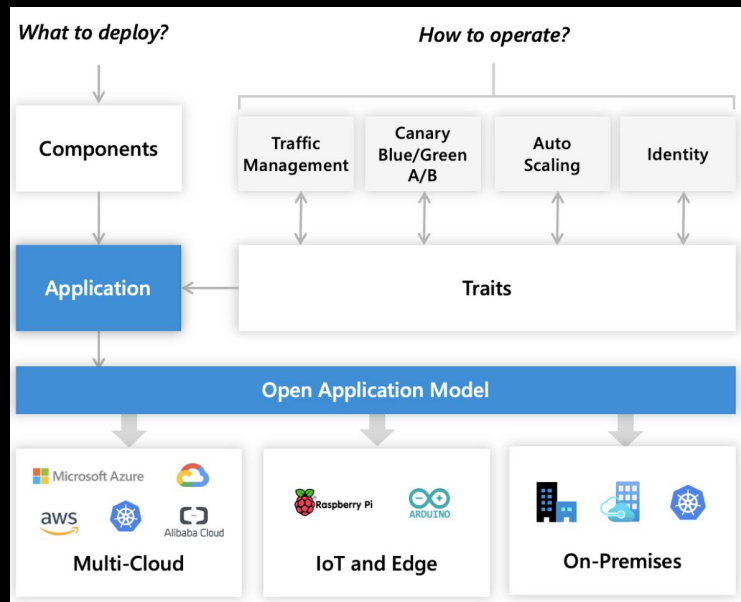
Argo Application Sets

Same app, multiple clusters, for unprivileged users



Open Application Model

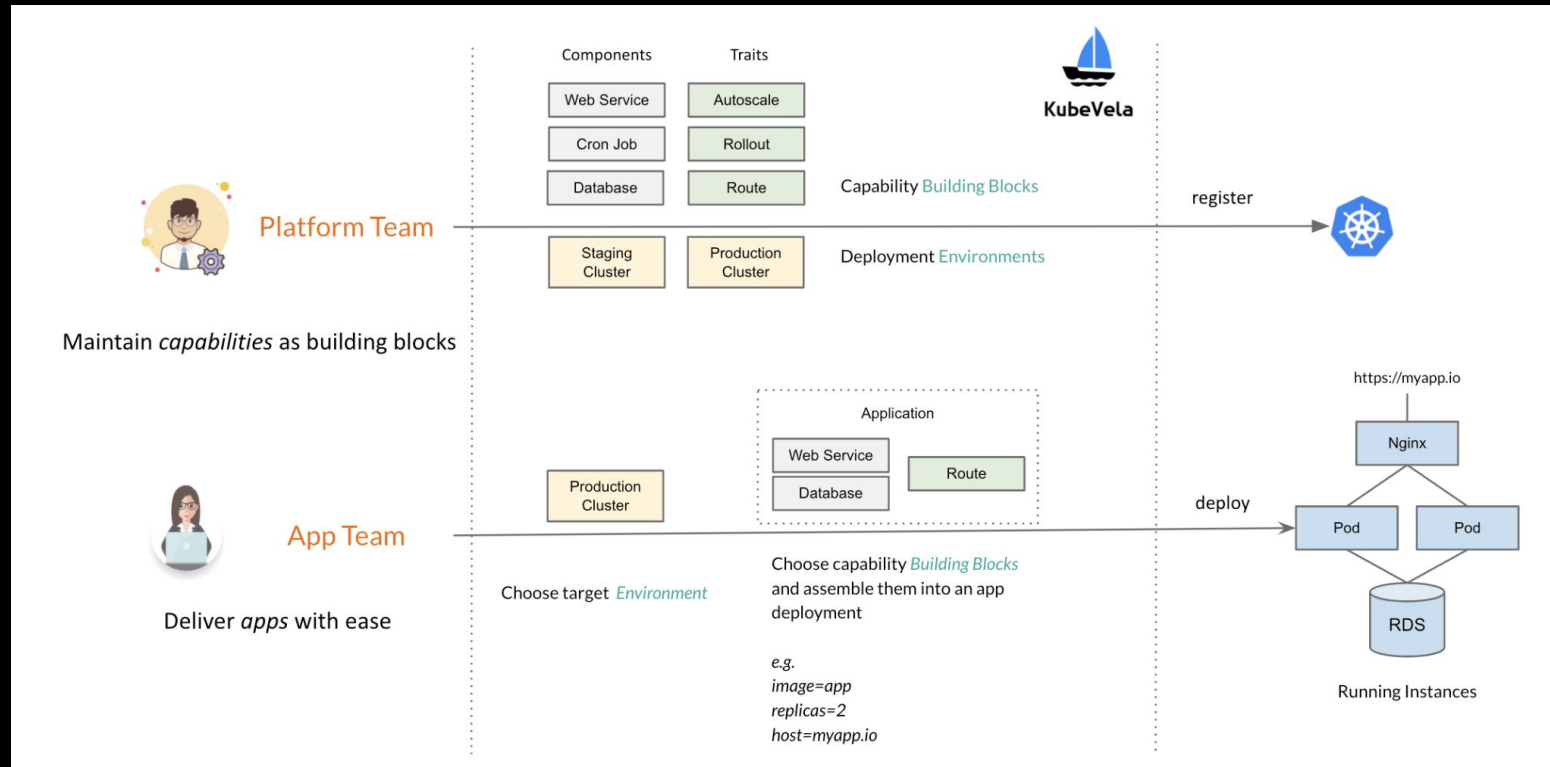
- Application first - define the app with a self-contained model, where operational behaviors as part of app definition
- Clarity and extensibility - an open standard to modularize infrastructure capabilities
- Runtime agnostic - a consistent experience to deploy and operate your apps across on-prem clusters, cloud providers or even edge devices





KubeVela

A reference implementation for OAM



Summary

Some things we will not get rid of

Deployment Manifests

How the app needs to run? How the app can run? What's the limits? When it is problematic?

You have to tell!

Infrastructure “Stacks”

Not the servers, but the container images, CSP integrations and supporting services.

That's why you have platform teams ;)

New Roles, new Responsibilities

A 100% clear role for Dev, Ops and DevOps will not be possible.

I believe Platform Teams can mediate between the roles.



Q&A

Let's connect!



Max Körbächer

Co-Founder & Sr. Manager Cloud Native
Engineering | Kubernetes Release Team



