

Progressive Delivery Techniques with Flagger

Stefan Prodan

 @stefanprodan



CICD - all-in-one monolith



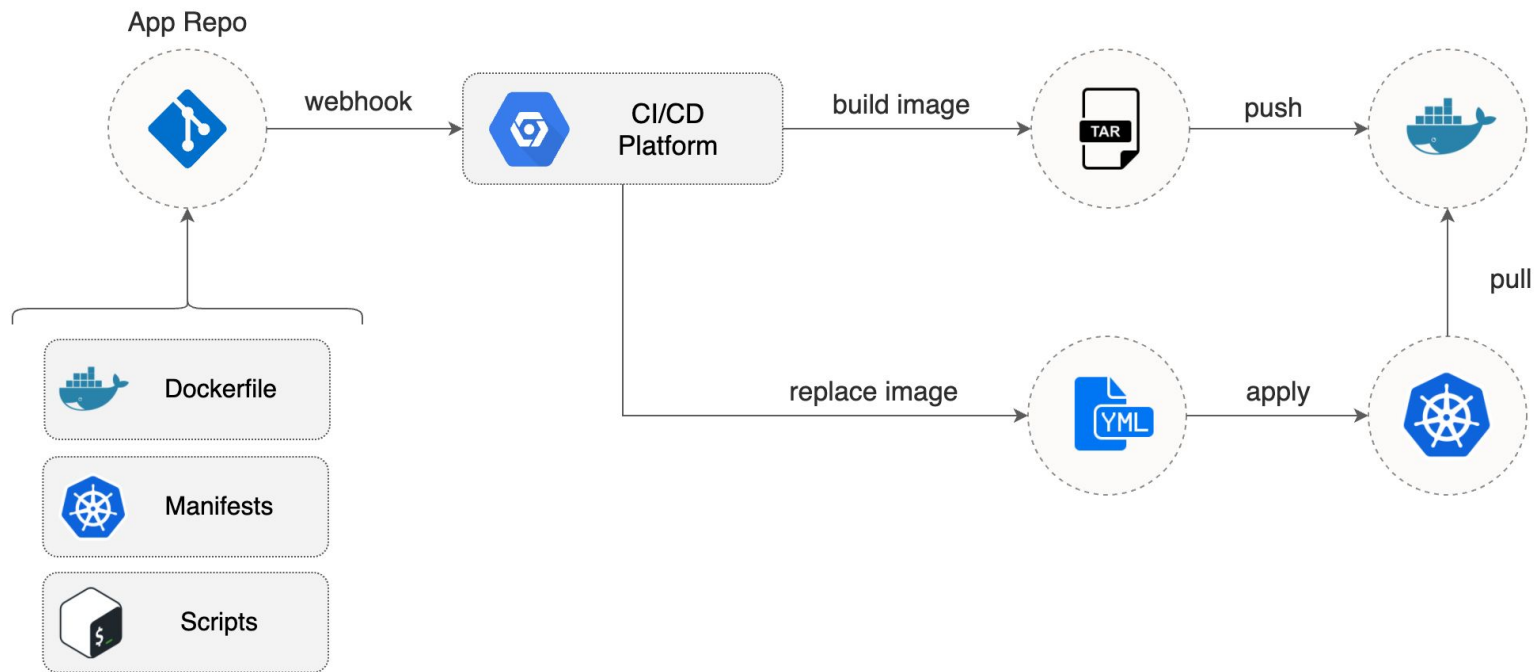
KubeCon



CloudNativeCon

North America 2020

Virtual





Traceability

- What app version is running on the cluster?
- What happens if multiple builds run in parallel?

Security

- How to manage cluster auth? What if 2FA is required?
- How to target multiple clusters?

Configuration Drift & Rollback

- How to deal with infra dependencies?
- Rollback an app means rebuilding the image?
- How to manage shared resources between apps?

CICD break-up



KubeCon



CloudNativeCon

North America 2020

Virtual



CI - build/push



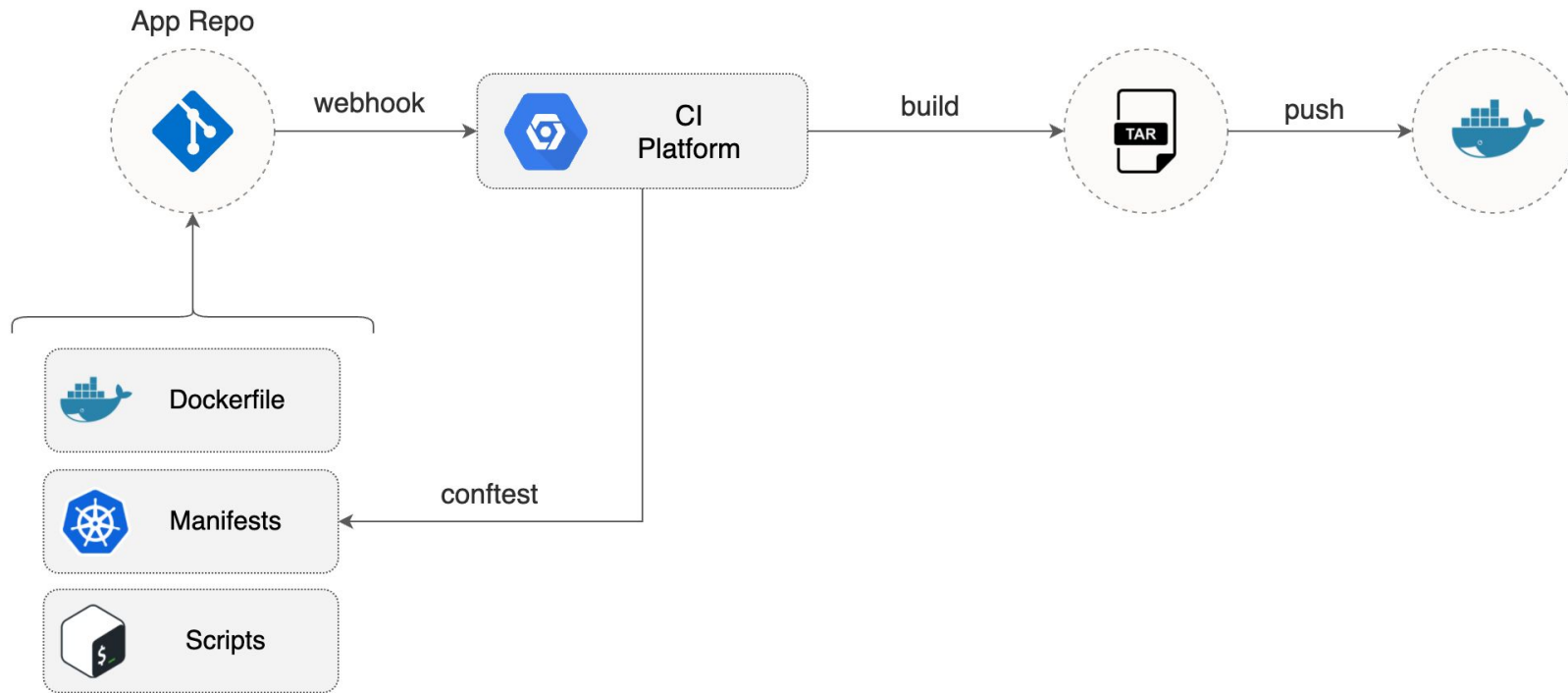
KubeCon



CloudNativeCon

North America 2020

Virtual



CD - GitOps reconciliation



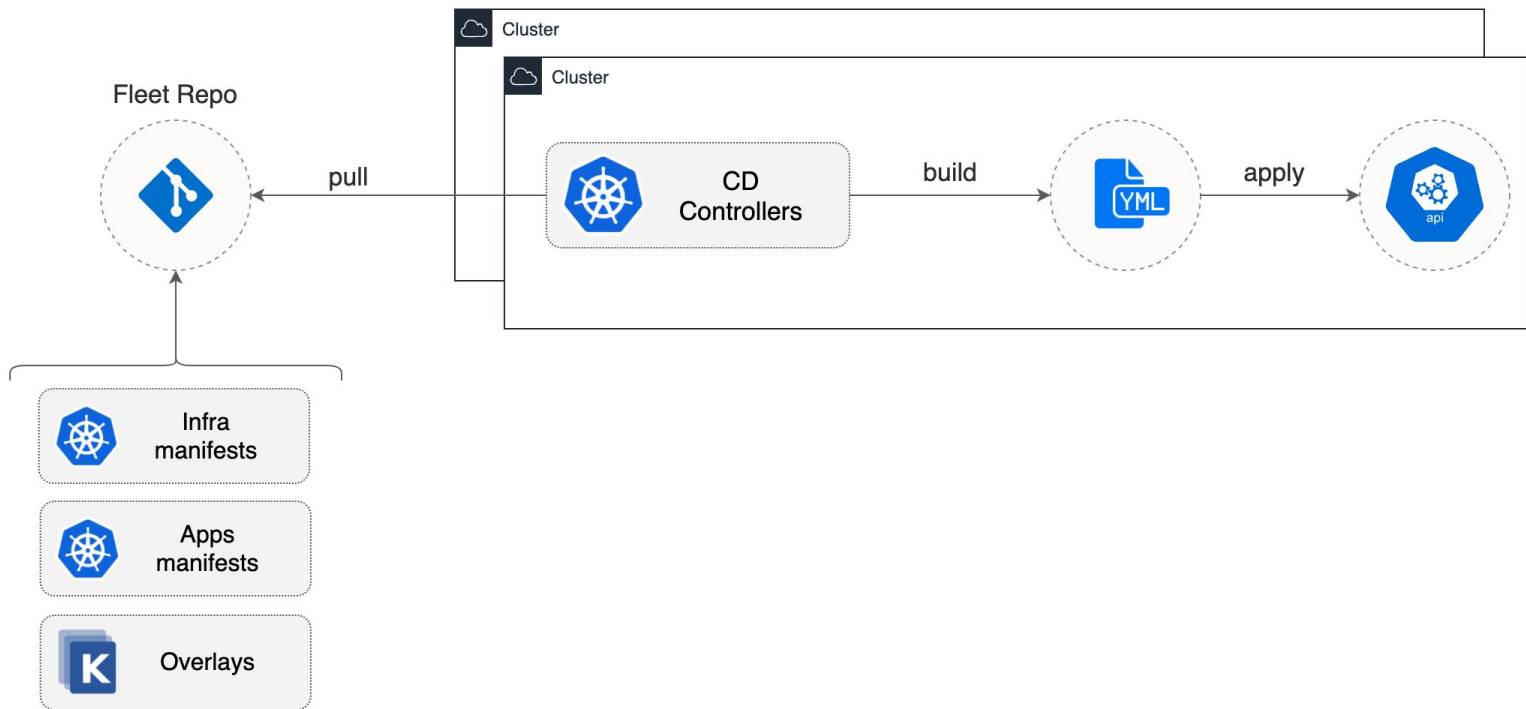
KubeCon



CloudNativeCon

North America 2020

Virtual



CD - challenges



KubeCon



CloudNativeCon

North America 2020

Virtual

- What if my app misbehaves after begin deployed?
- How can I run conformance tests after a deployment?
- Can I bundle Kubernetes configuration changes with an app version?
- How to rollback automatically?
- Can I run multiple versions of an app in parallel?
- How to A/B test between versions?



deploy  release

Progressive Delivery



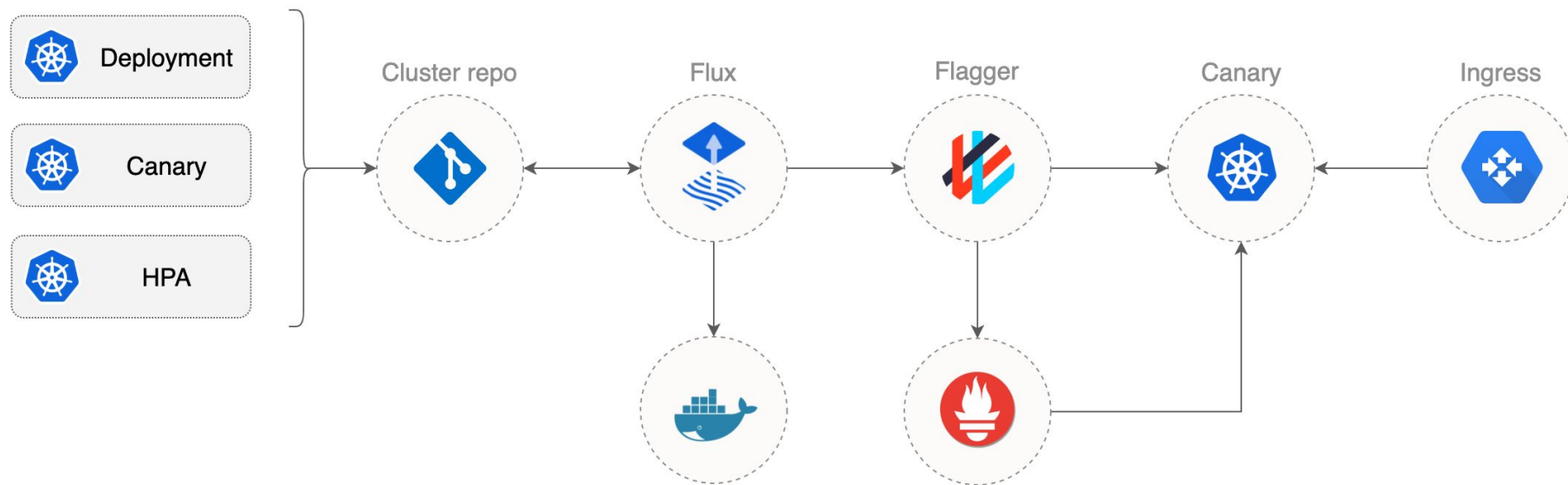
KubeCon



CloudNativeCon

North America 2020

Virtual



Flagger is a progressive delivery tool that **automates the release** process for applications running on Kubernetes.

It reduces the risk of introducing a new software version in production by **gradually shifting traffic** to the new version while measuring metrics and running conformance tests.

Flagger comes with a **declarative model** for decoupling the deployment of apps on Kubernetes from the release process.

Flagger - Traffic Management



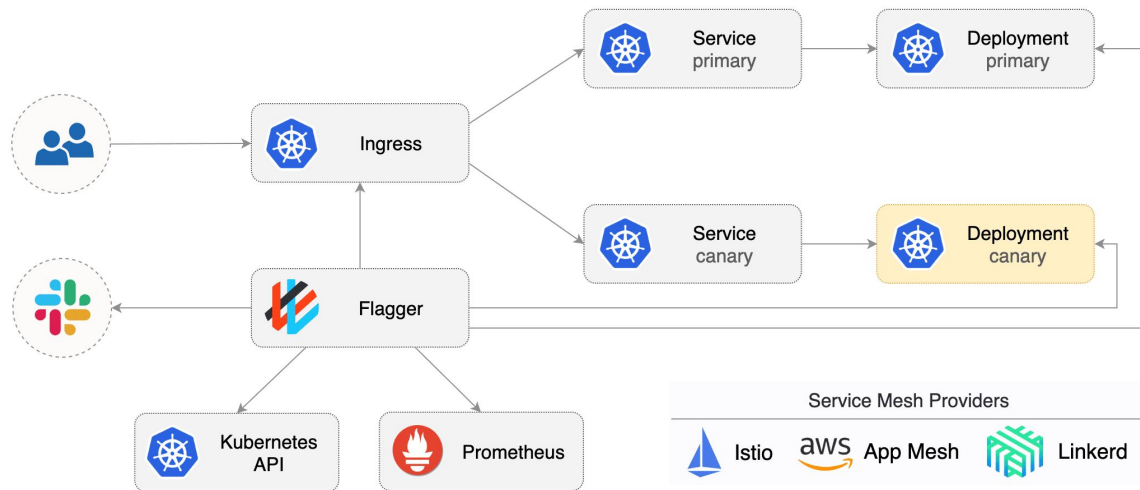
Virtual

- Service Mesh

- Istio
- Linkerd
- App Mesh

- Ingress Controllers

- Contour
- Gloo
- NGINX
- Skipper



Deployment Strategies



KubeCon



CloudNativeCon

North America 2020

Virtual

- Canary Release (progressive traffic shifting)
 - Applications that expose HTTP or gRPC APIs
- A/B Testing (HTTP headers and cookies traffic routing)
 - User-facing applications that need session affinity
- Blue/Green (traffic mirroring)
 - Idempotent APIs
- Blue/Green (traffic switch)
 - Stateful applications
 - Legacy applications

Canary Release



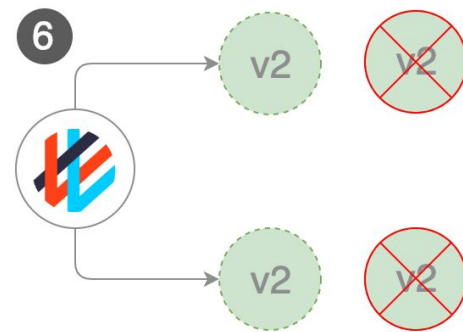
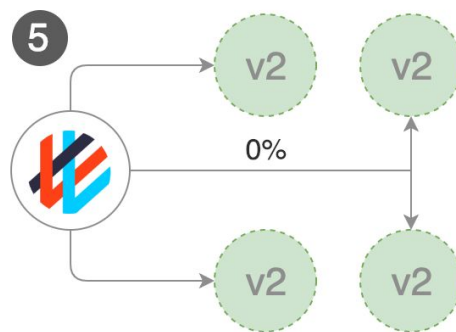
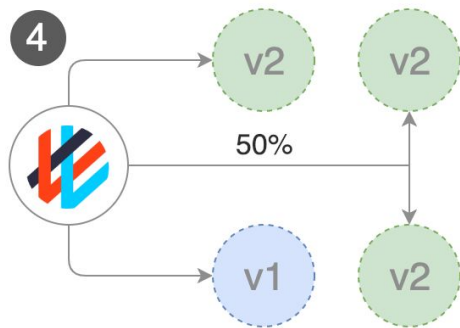
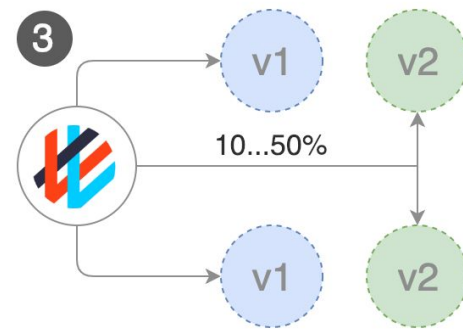
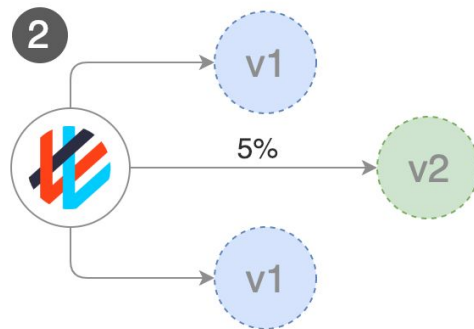
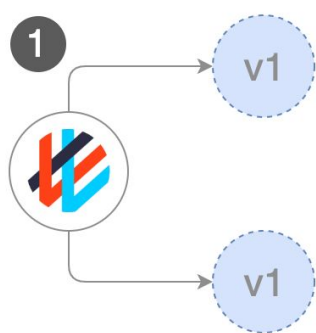
KubeCon



CloudNativeCon

North America 2020

Virtual



A/B Testing



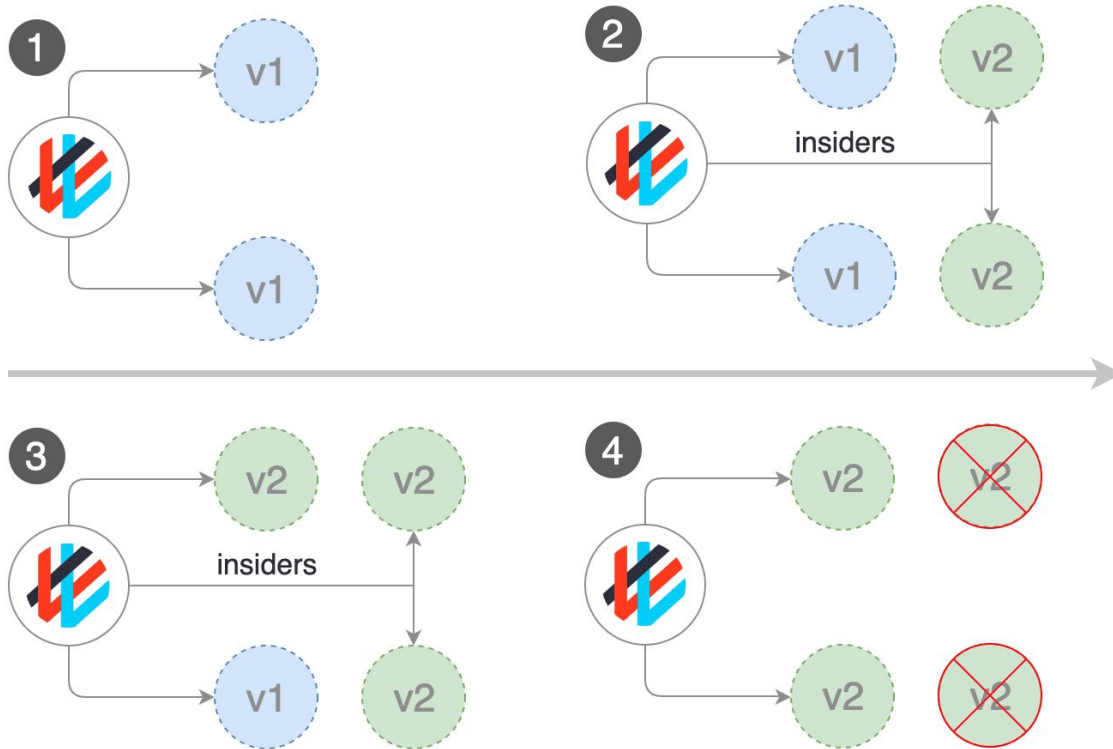
KubeCon



CloudNativeCon

North America 2020

Virtual



Blue/Green



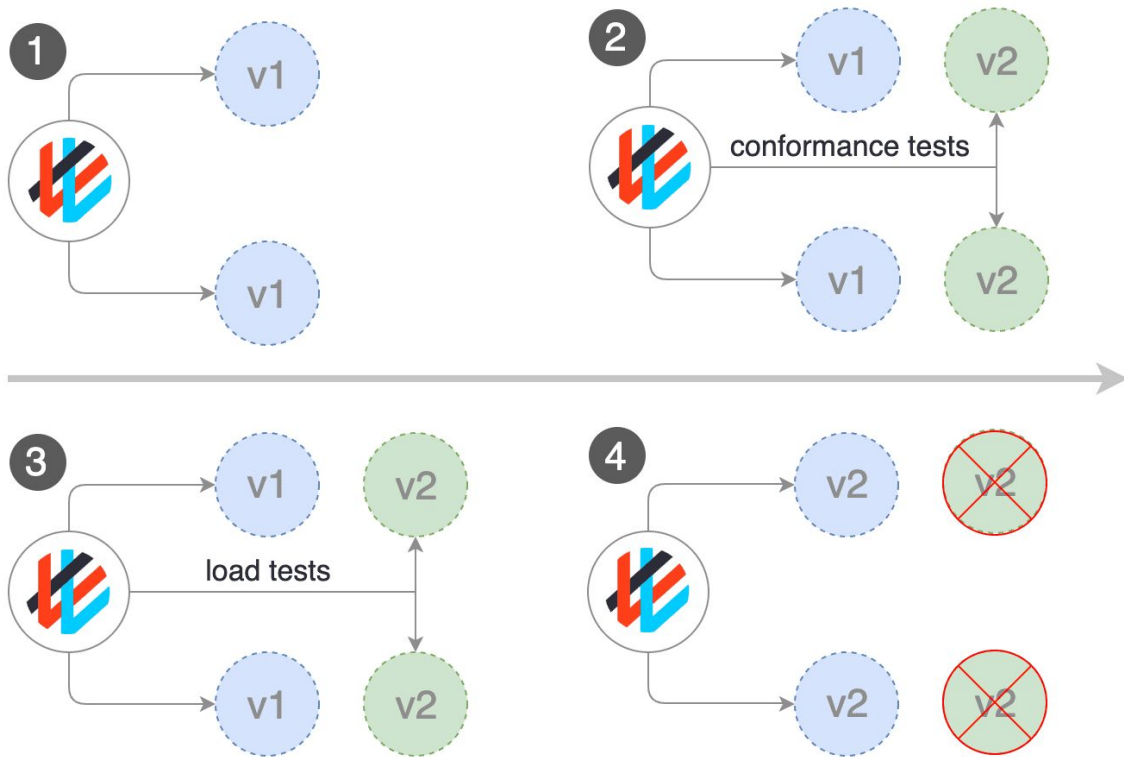
KubeCon



CloudNativeCon

North America 2020

Virtual



Demo

Continuous Delivery with Flux v2

Progressive Delivery with Flagger

Traffic management with Contour and Linkerd v2



Links



KubeCon



CloudNativeCon

North America 2020

Virtual

Flagger Repo

<https://github.com/weaveworks/flagger>

Flagger Docs

<https://docs.flagger.app>

Linkerd + Contour Demo

<https://github.com/stefanprodan/gitops-linkerd>

Istio Demo

<https://github.com/stefanprodan/gitops-istio>

