



OCTOBER 28, 2021

Annie Talvasto

Top new CNCF projects to look out for



Top new CNCF projects to look out for

What value do you get by attending this talk?

- Get inspired
- Get to know cool CNCF projects

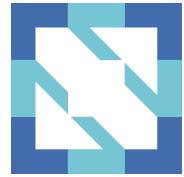
Who am I?

@AnnieTalvasto

CNCF Ambassador & Product Marketing Manager at CAST AI

- Kubernetes & CNCF meetup co-organizer
- Azure MVP
- Startup-coach
- Co-host of Cloudgossip podcast - cloudgossip.net





**CLOUD NATIVE
COMPUTING FOUNDATION**

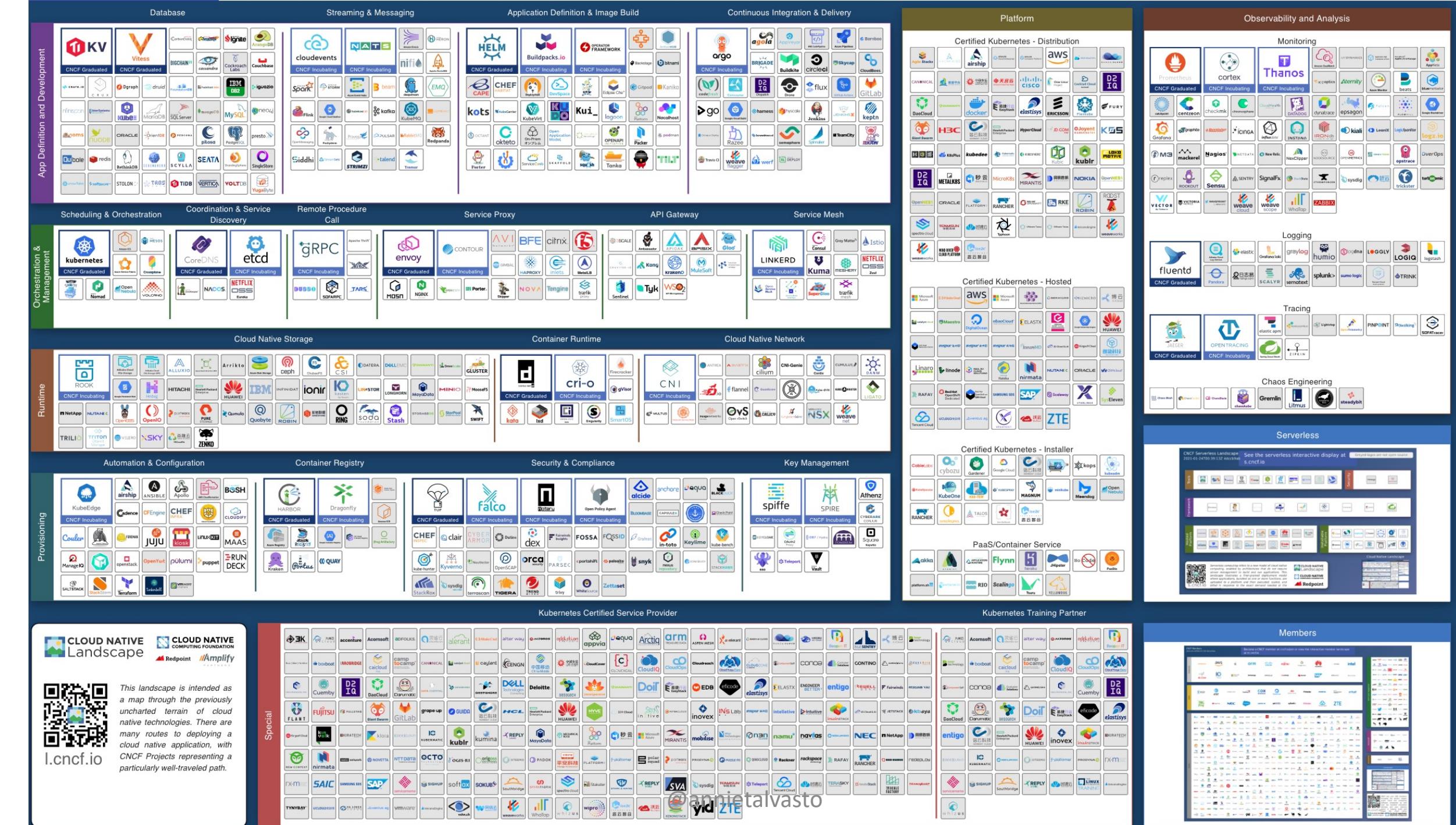
Building sustainable ecosystems for cloud native software

The Cloud Native Computing Foundation (CNCF) hosts critical components of the global technology infrastructure. CNCF brings together the world's top developers, end users, and vendors and runs the largest open source developer conferences. CNCF is part of the nonprofit Linux Foundation.

Impact of cloud native

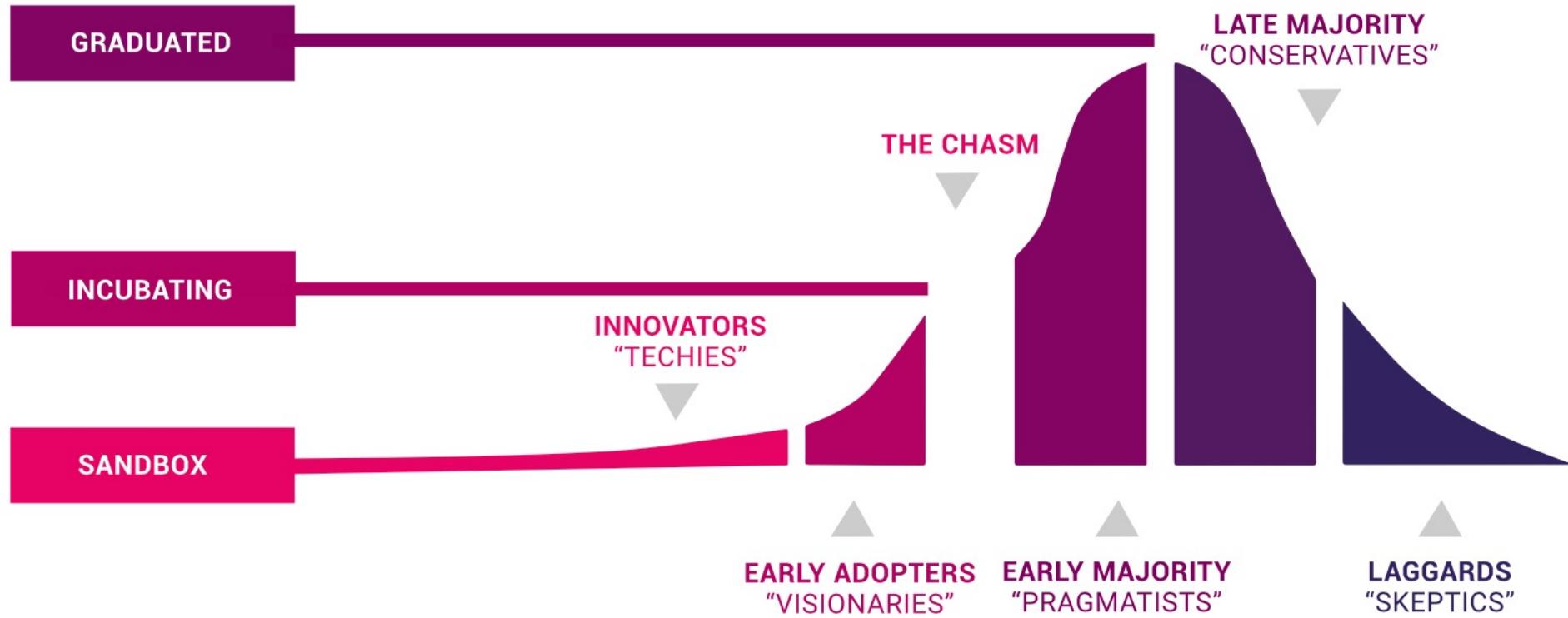
- The use of containers in production has increased to 92%, up from 84% last year, and up 300% from our first survey in 2016.
- Kubernetes use in production has increased to 83%, up from 78% last year.

- CNCF 2020 Survey



3 stages of CNCF projects





The projects in this session

- This is not scientific selection method, nor is it fortune telling
- Expectation management: usually CNCF intro to projects talks are around 30 to 45 minutes – this is a shorter talk covering many projects.
- Linkerd, Keda, Flux, Kudo, Meshery

Announcing Linkerd 2.9! mTLS for all TCP, and much more. [Read more »](#)



A different kind of service mesh

Ultra light, ultra simple, ultra powerful. Linkerd adds security, observability, and reliability to Kubernetes, *without* the complexity. CNCF-hosted and 100% open source.

[Get Started](#)[Join the Community](#)

 Star 6,489  Watch 184  Fork 703



Thriving open source community

Linkerd is 100% Apache-licensed, with an incredibly fast-growing, active, and friendly community.

[Come join the fun!](#)

Simple, minimalist design

No complex APIs or configuration. For most applications, Linkerd will "just work" out of the box.



Ultralight and ultra fast

Built in Rust, Linkerd's data plane proxies are incredibly small (<10 mb) and blazing fast (p99 < 1ms).



Installs in seconds with zero config

Linkerd's control plane installs into a single namespace, and services can be safely added to the mesh, one at a time.

Linkerd

- Service mesh
- Ultralight, ultrafast, security-first service mesh for Kubernetes.
- The overall goal is to reduce mental overhead of having service mesh
- What does it do?
 - Observability
 - Reliability
 - Security

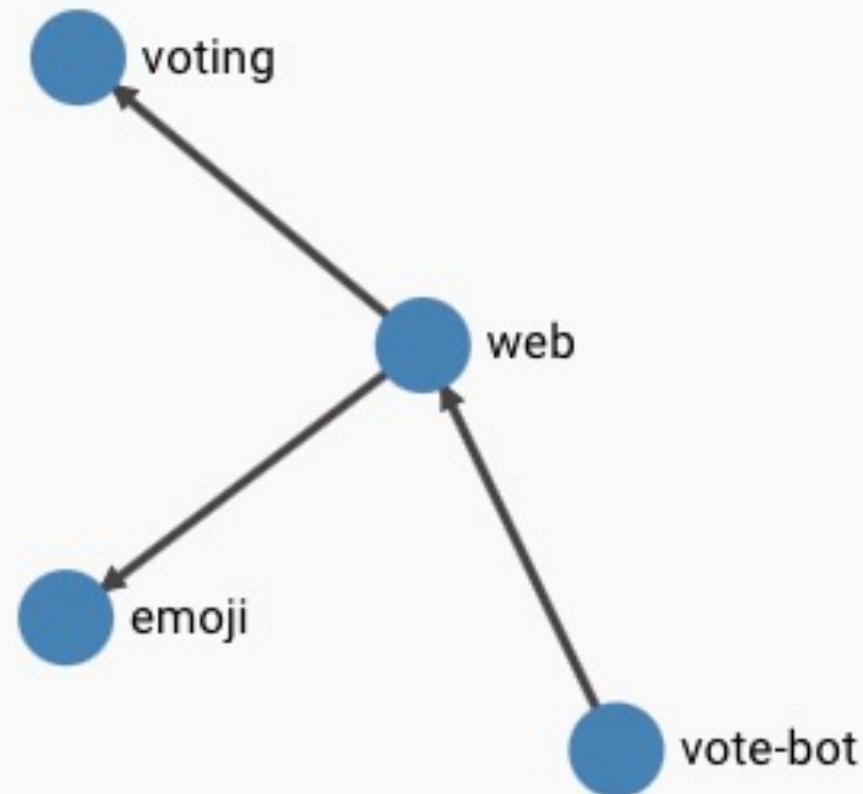
What are the benefits of Linkerd?

- Thriving open source community
- Simple, minimalist design
- Deep Runtime Diagnostics
- Ultralight and ultra fast
- Installs in seconds with zero config
- Actionable service metrics

What are the Linkerd principles?

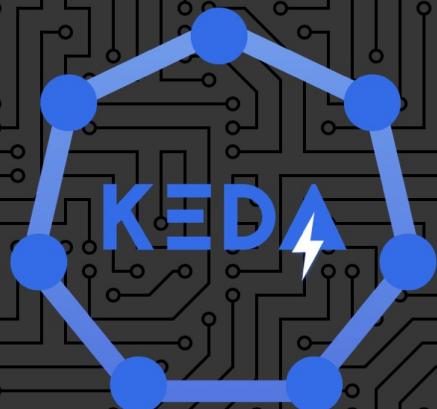
- Just works
- Ultralight
- Simple
- Security first
- Linkerd has a custom proxy, Linkerd2-proxy.

Linkerd demo app



What is needed to use Linkerd:

```
cat deployment.yml | linkerd inject - | kubectl apply -f -
```



Kubernetes Event-driven Autoscaling

Application autoscaling made simple

[Concepts](#)[Deploying KEDA](#)[Architecture](#)[Scalers](#)[Blog](#)

What is KEDA?

KEDA is a [Kubernetes](#)-based Event Driven Autoscaler. With KEDA, you can drive the scaling of any container in Kubernetes based on the number of events needing to be processed.

KEDA is a single-purpose and lightweight component that can be added into any Kubernetes cluster. KEDA works alongside standard Kubernetes components like the [Horizontal Pod Autoscaler](#) and can extend functionality without overwriting or duplication. With KEDA you can explicitly map the apps you
@annietalvasto

What is Keda?

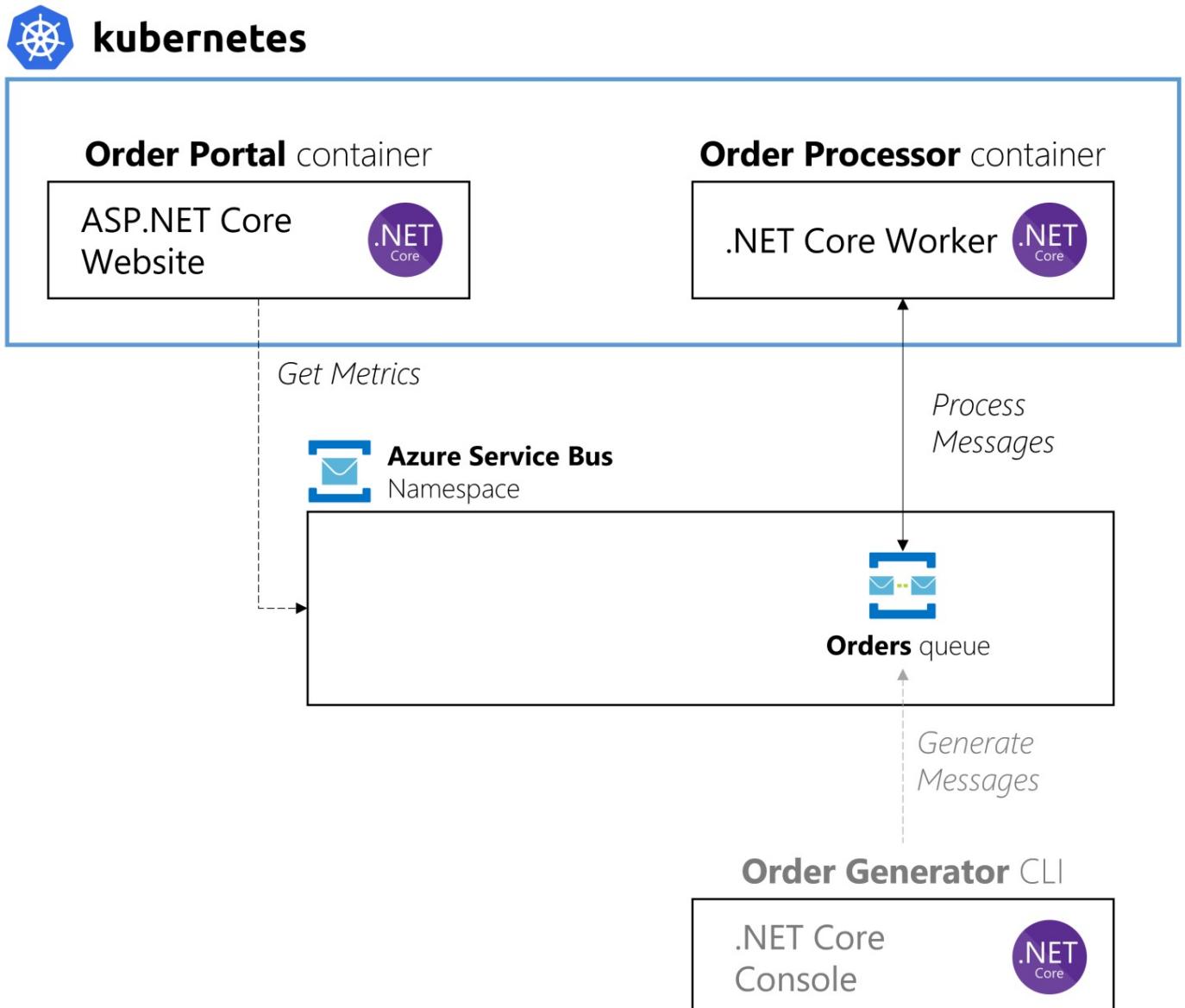
- Default Kubernetes Scaling is not well suited for event driven applications, kubernetes is more for resource based scaling (CPU and memory).
- Keda: Event driven scale controlling that can run inside any kubernetes cluster.
- You can install it into new or existing clusters.

What are the Keda principles?

- Not rebuilding anything that Kubernetes offers out of the box.
- Single purpose, simple, non-intrusive.
- Works with any container and any workload
- Two public case studies: Alibaba Cloud & CAST AI

Demo

KEDA:
Scaling .NET
Core worker
with Azure
Service Bus





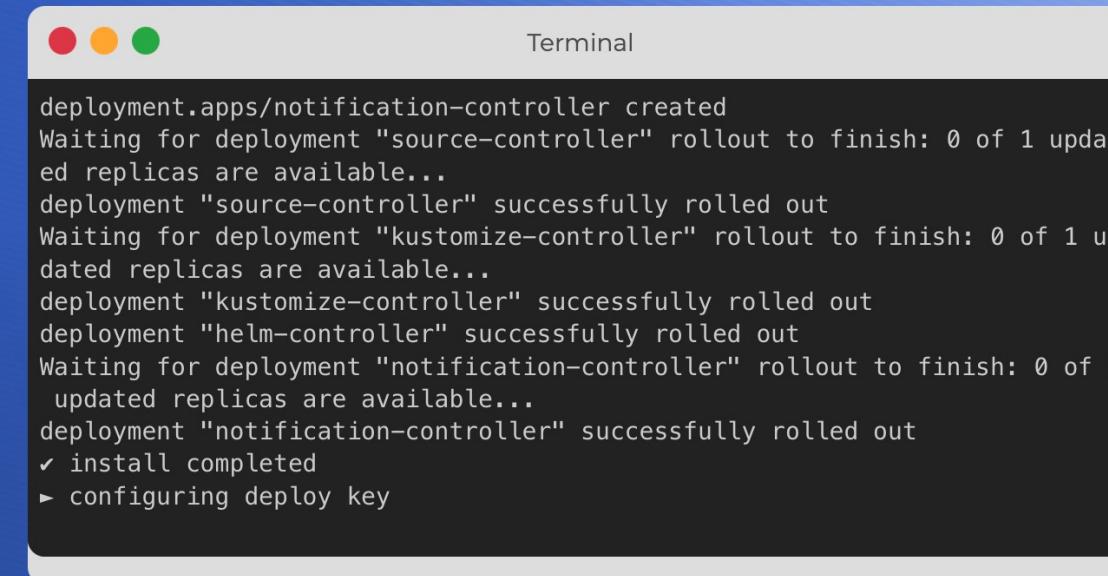
Flux APIs are now stable!

Flux - the GitOps family of projects

Flux is a set of continuous and progressive delivery solutions for Kubernetes, and they are open and extensible.

The latest version of Flux brings many new features and made Flux more flexible and versatile.

[Get started →](#)



A terminal window titled "Terminal" showing the output of a Flux deployment command. The logs indicate the creation of a notification controller, followed by a deployment rollout for source and kustomize controllers, each successfully rolled out. It then shows a deployment for a helm controller, followed by a notification controller rollout, both successfully rolled out. Finally, it shows an install completed message and a configuration step for a deploy key.

```
deployment.apps/notification-controller created
Waiting for deployment "source-controller" rollout to finish: 0 of 1 updated replicas are available...
deployment "source-controller" successfully rolled out
Waiting for deployment "kustomize-controller" rollout to finish: 0 of 1 updated replicas are available...
deployment "kustomize-controller" successfully rolled out
deployment "helm-controller" successfully rolled out
Waiting for deployment "notification-controller" rollout to finish: 0 of 1 updated replicas are available...
deployment "notification-controller" successfully rolled out
✓ install completed
► configuring deploy key
```

Flux in short

What is Flux?

- What is GitOps?

- Kubectl apply

- Kubectl set image

- Helm upgrade

- Kubectl upgrade

- > git push

GitOps provides one model for making infrastructure, apps and Kubernetes add-on changes, you have consistent end-to-end workflow across your entire organization

- Why is Flux great for GitOps?

Flux practices & benefits

Defined GitOps practices:

- 1) Describe your system declaratively
- 2) Keep configuration under source control
- 3) Use software agents to reconcile and ensure correctness and alert for drift

Benefits

- Collaboration on infra
- Access Control
- Auditable History
- Drift Correction
- Clear boundaries between dev-team and kubernetes



The Kubernetes Universal Declarative Operator

↔ Get Started ↔

Focus on your software ...

The Kubernetes Universal Declarative Operator (KUDO) is a highly productive toolkit for writing Kubernetes Operators.

... not on deploying to Kubernetes

Using KUDO you can deploy your applications, have the tools needed to operate them, and understand how they're behaving – all without a Ph.D. in Kubernetes.

Automate Day-2 Operations

KUDO lets you configure an Operator's entire lifecycle using a declarative spec, including things like backup/restore. You don't have to write Go unless you want to.

What is KUDO?

KUDO is a toolkit that makes it easy to build [Kubernetes Operators](#), in most cases just using YAML.

It provides a set of pre-built Operators, that you can use out of the box or easily customize.

Finally, KUDO lets you standardize the way you run Operators.

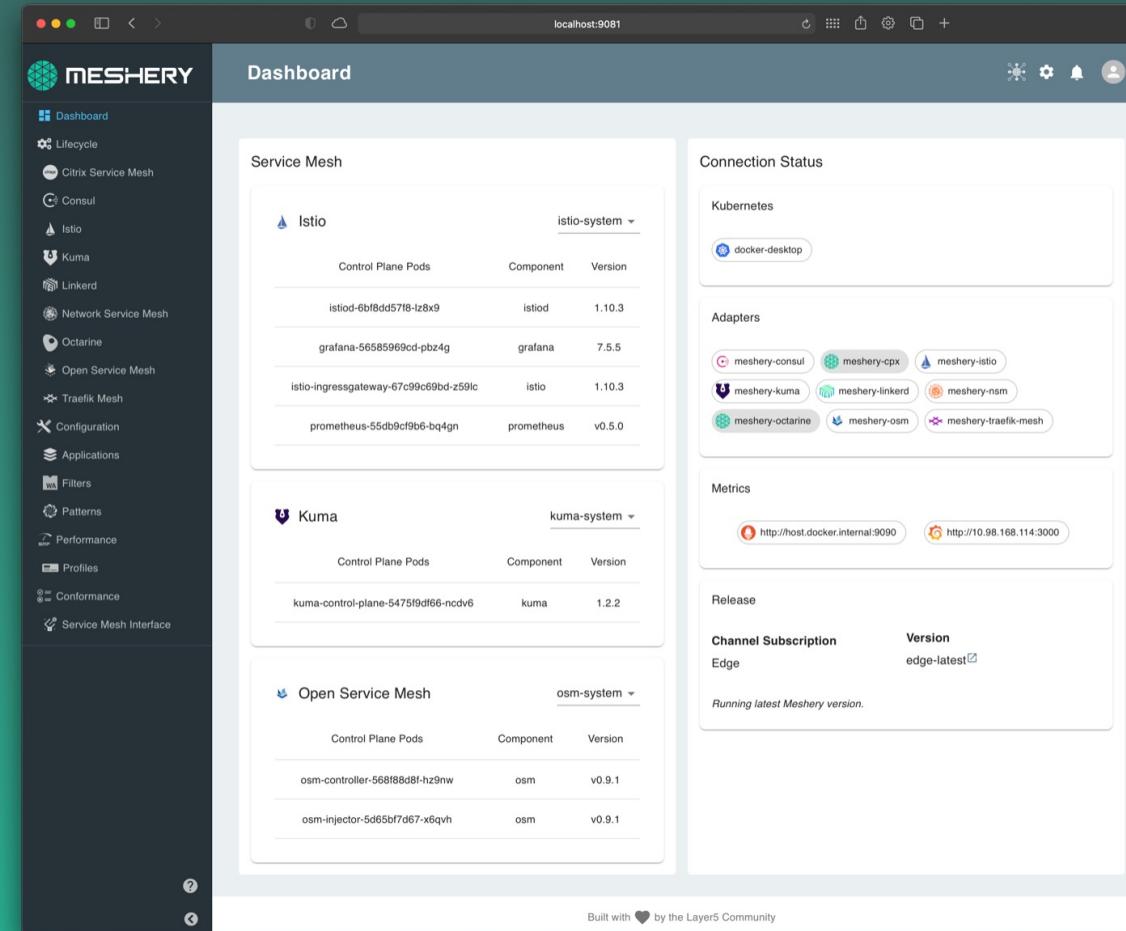
What is Kudo?

- Stateless vs stateful app
- Kubernetes has been very focused on stateless apps – and stateful apps do not like it -> solution: operators.
- Building operators requires deep expertise and may require thousands of lines of code -> substantial engineering resource needed.
- **Kudo = Kubernetes Universal Declarative Operator**

What are the benefits of Kudo?

- Kudo can create operators without needing deep knowledge of kubernetes or coding – by defining lifecycle stages.
- Just kubernetes APIs, a lot easier to learn
- Has kubernetes native management, aka using of kubectl and other familiar tools

The service mesh management plane



The screenshot shows the Meshery Dashboard interface running at `localhost:9081`. The left sidebar contains a navigation menu with links to Dashboard, Lifecycle, Citrix Service Mesh, Consul, Istio, Kuma, Linkerd, Network Service Mesh, Octarine, Open Service Mesh, Traefik Mesh, Configuration, Applications, Filters, Patterns, Performance, Profiles, Conformance, and Service Mesh Interface.

The main content area is divided into several sections:

- Service Mesh**:
 - Istio**:

Control Plane Pods	Component	Version
istiod-6bf8dd57f8-lz8x9	istiod	1.10.3
grafana-56585969cd-pbz4g	grafana	7.5.5
istio-ingressgateway-67c99c69bd-z59lc	istio	1.10.3
prometheus-55db9cf9b6-bq4gn	prometheus	v0.5.0
 - Kuma**:

Control Plane Pods	Component	Version
kuma-control-plane-5475f9df66-ncdv6	kuma	1.2.2
 - Open Service Mesh**:

Control Plane Pods	Component	Version
osm-controller-568f88d8f-hz9nw	osm	v0.9.1
osm-injector-5d65bf7d67-x6qvh	osm	v0.9.1
- Connection Status**:
 - Kubernetes**: docker-desktop
- Adapters**:
 - meshery-consul
 - meshery-cpx
 - meshery-istio
 - meshery-kuma
 - meshery-linkerd
 - meshery-nsm
 - meshery-octarine
 - meshery-osm
 - meshery-traefik-mesh
- Metrics**:
 - <http://host.docker.internal:9090>
 - <http://10.98.168.114:3000>
- Release**:
 - Channel Subscription: Edge
 - Version: edge-latest
 - Running latest Meshery version.

What is Meshery?

The Service Mesh Management Plane

Service mesh:

Control plane

Data plane

+

Management plane

What is Meshery?

- Supports over 10 different service meshes
- Multi-mesh management
 - Lifecycle
 - Workload
 - Performance
 - Configuration
 - Patterns and practices
 - Chaos and filters
- Meshery is about halfway to complete architecture

Wrap up

CNCF overview

Projects:

- Linkerd
- Kudo
- Flux
- Keda
- Meshery

Learn more

- CNCF survey -<https://www.cncf.io/blog/2020/11/17/cloud-native-survey-2020-containers-in-production-jump-300-from-our-first-survey/>
- All project sites
 - Linkerd - <https://linkerd.io>
 - Kudo - <https://kudo.dev>
 - Keda - <https://keda.sh>
- Support your favorite projects in GitHub!

- Case studies:
- <https://www.cncf.io/case-studies/babylon/>
- <https://www.cncf.io/case-studies/elkjop/>

- CNCF End User Technology Radar: <https://radar.cncf.io>
- Keynote: Predictions from the Technical Oversight Committee (TOC) - Liz Rice, CNCF TOC Chair -
<https://www.youtube.com/watch?v=bESogtuHwX0&feature=youtu.be>
- Techworld with Nana - <https://www.youtube.com/channel/UCdngmbVKX1Tgre699-XLIUA>
- Cncf youtube - <https://www.youtube.com/channel/UCvqbFHwN-nwalWPjPUKpvTA>

- [Links and slides: github.com/annietalvasto](https://github.com/annietalvasto)

[BACK TO EPISODES](#)

Adventures in open source with Tom Kerkhove

JANUARY 14TH, 2021 | 46:06 | S2:E4

[SHARE](#)[EMBED](#)[RECAST](#)[SUBSCRIBE](#)[DOWNLOAD MP3](#)EPISODE DETAILS / [TRANSCRIPT](#)

EPISODE SUMMARY

Today's guest on Cloud Gossip is Tom Kerkhove!

Tom works as an Azure Architect at Codit, he's a Github Star, CNCF Ambassador, Azure MVP and he's active as maintainer of Promitor and Keda.

Tom is going to talk to us about how the world of Open-Source projects works, the importance of supporting them, and his personal experience as a maintainer.

EPISODE NOTES

We're going to learn about KEDA and CNCF Sandbox projects, what they are and how they work, and learn about some of Tom's insights in the industry.

He's going to talk about how GitHub is helping the world of Open Source projects and how he uses the platform to engage with the users.



One thing to take away

Got questions later?
@annietalvasto





OCTOBER 28, 2021

Thank you!

