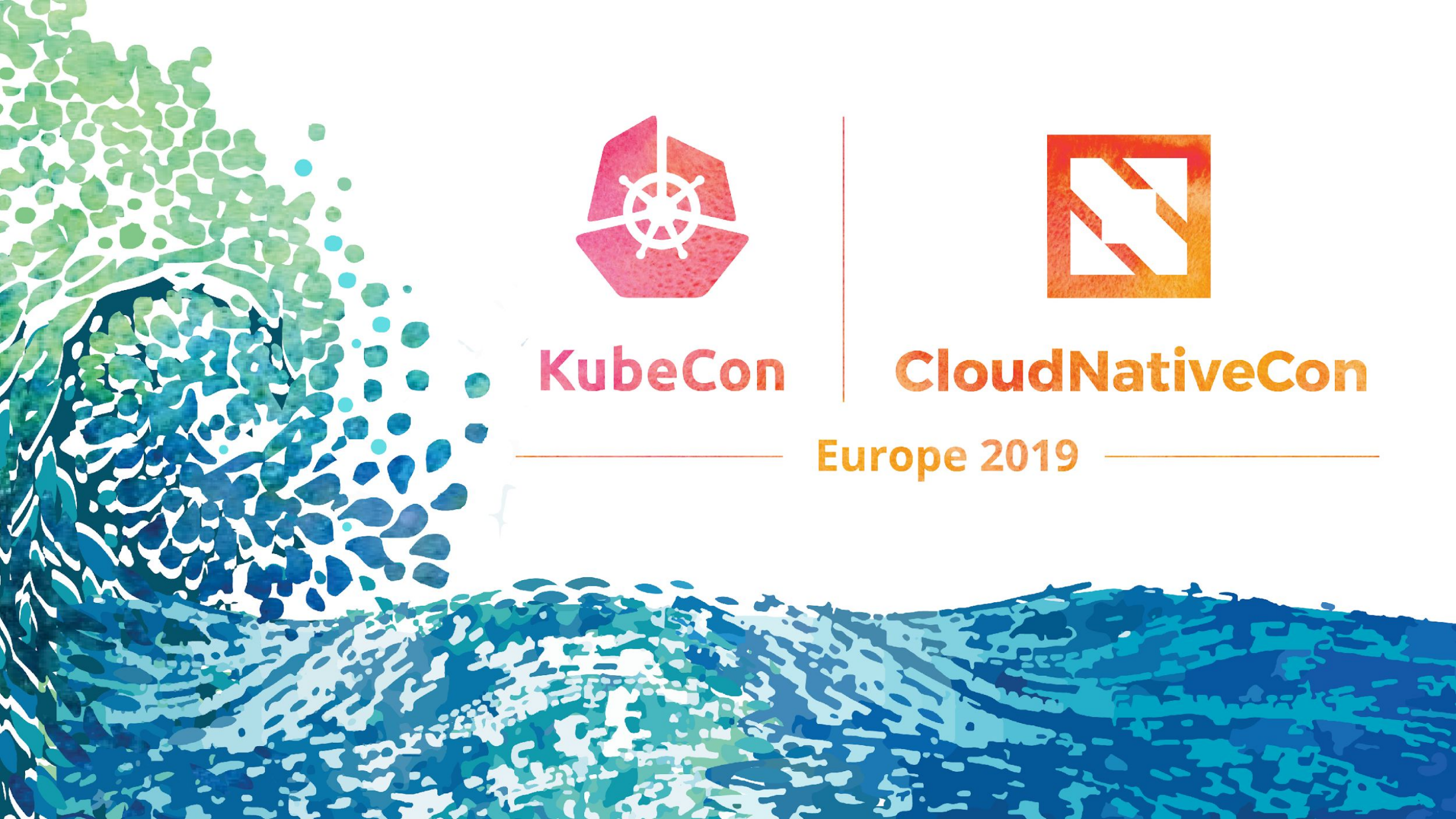

KubeCon
CloudNativeCon
Europe 2019

Introduction to SIG Cluster Lifecycle

Who Are We?

Lucas Käldström
SIG Cluster Lifecycle co-lead
CNCF Ambassador & CKA
Contractor for Weaveworks
@luxas

Timothy St. Clair
SIG Cluster Lifecycle co-lead
Steering Committee Member
Senior Staff Engineer @VMWare
@timothysc

Who are SCL?

- 600+ members on mailing list
- 2000+ members in #sig-cluster-lifecycle Slack
- 20+ companies represented during SIG meetings
- 5 continents with contributors
- O(10^3) contributions per cycle
- 15+ SIG sponsored subprojects

Mission


"SIG Cluster Lifecycle's objective is to simplify creation, configuration, upgrade, downgrade, and teardown of Kubernetes clusters and their components."

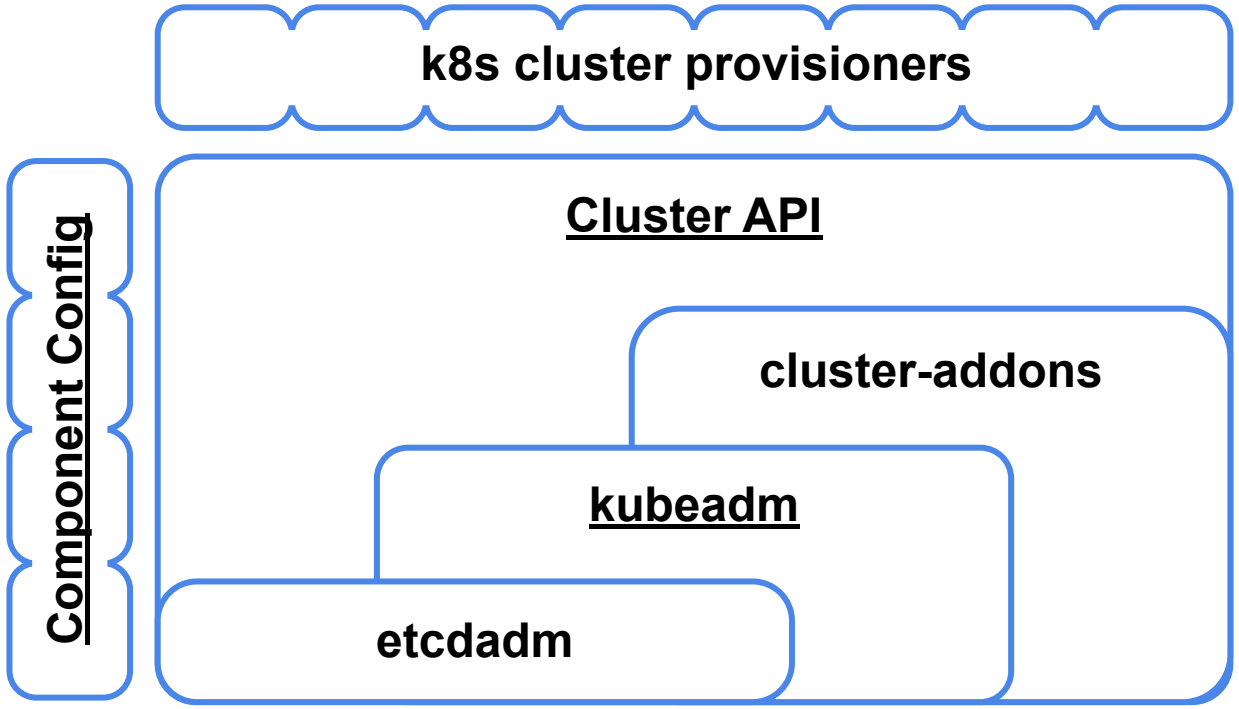
-- SIG Cluster Lifecycle Charter

Why?

- To prevent the mistakes of other open source clustering tools, as...
  - Kubernetes is the beginning of the story, not the end
  - Commoditizing the deployment of the core raises all boats and allows the community to focus on solving end user problems
  - “Production Grade” shouldn’t be firewalled
  - It should “just work”
  - Because cross provider behavior matters (conformance)
- To make the management of (X) clusters across (Y) providers simple, secure, and configurable.
- unix philosophy 4 lyfe

SCL Overview

SCL is one of the biggest Kubernetes SIGs, with 100s of contributors across several companies actively contributing to 17 subprojects and several workgroups



k8s cluster provisioners:

- minikube
- kops
- kubespray
- kind (SIG Testing)
- kubeadm-dind-cluster
- cluster-api-provider-<name>
- ...

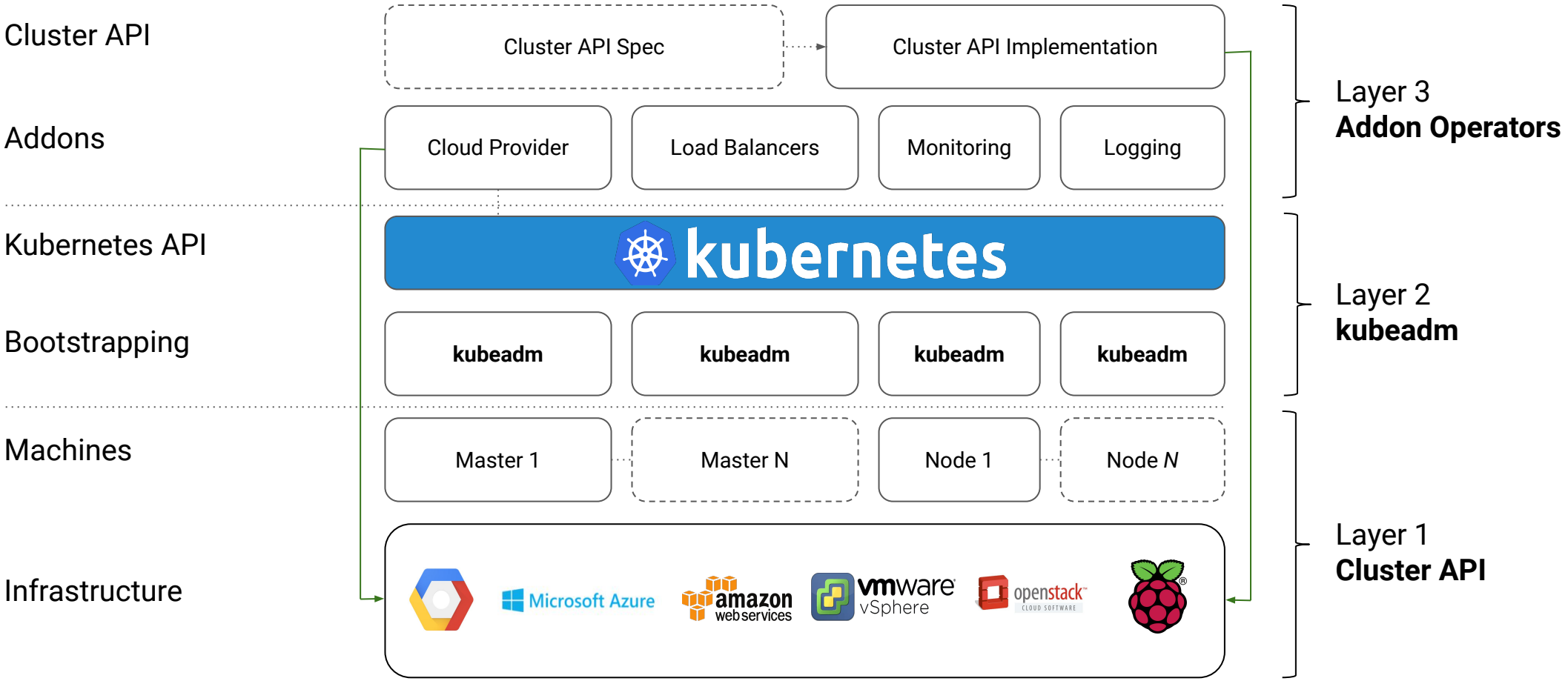
Key Subprojects

kubeadm (GA)

- kubeadm's task is to set up a **best-practice cluster** for each *minor version*
- The user experience should be *simple*, and the cluster reasonably *secure*
- kubeadm's scope is limited; intended to be a **building block**
  - Only ever deals with the local filesystem and the Kubernetes API
  - Agnostic to ***how exactly*** the kubelet is run
  - Setting up or favoring a specific CNI network is **out of scope**
- Composable architecture with everything divided into **phases**
  - Allows for **DIY** using other higher order tools as chef/puppet/etc.

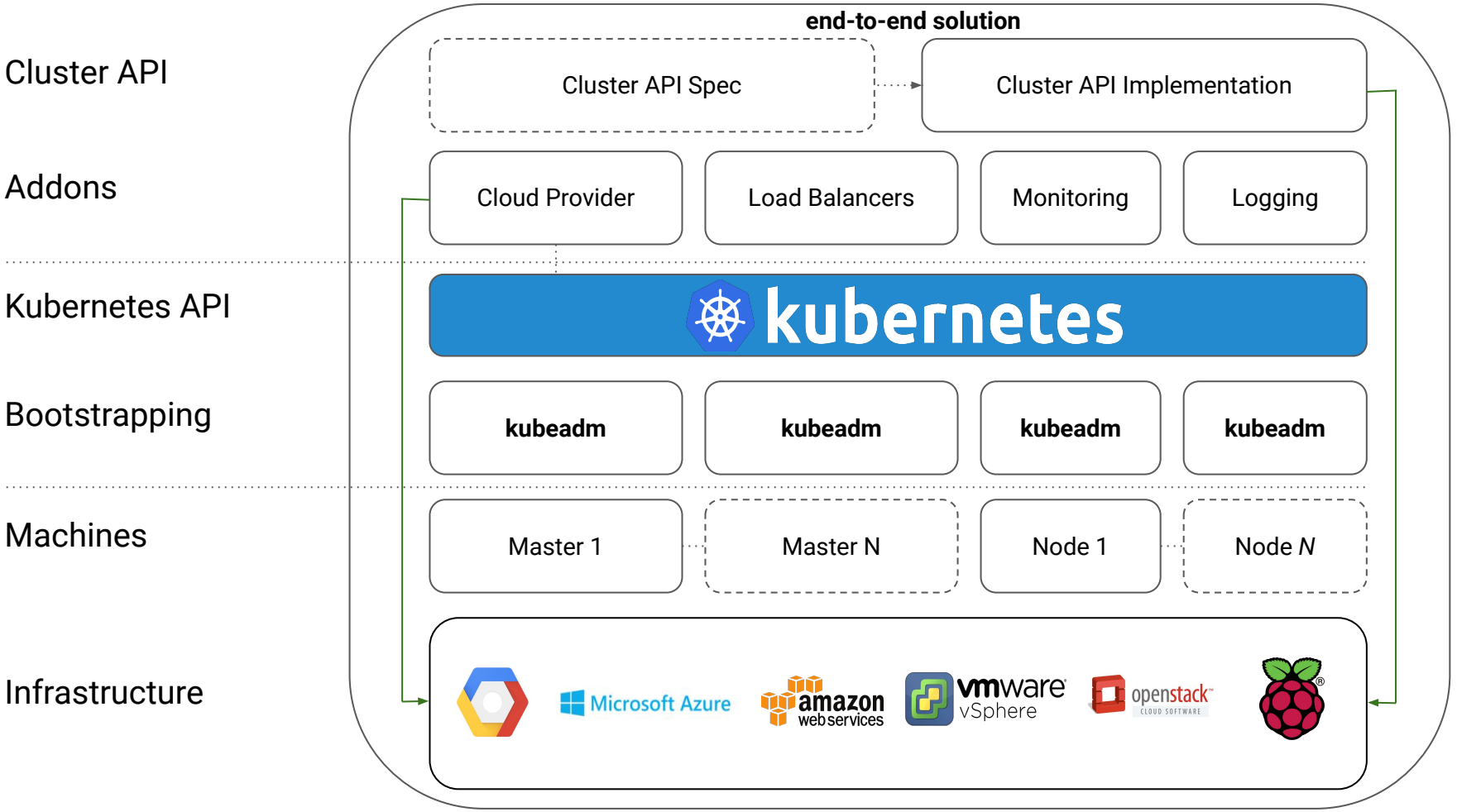
kubeadm (GA)

= The official tool to bootstrap a minimum viable, best-practice Kubernetes cluster



kubeadm vs an end-to-end solution

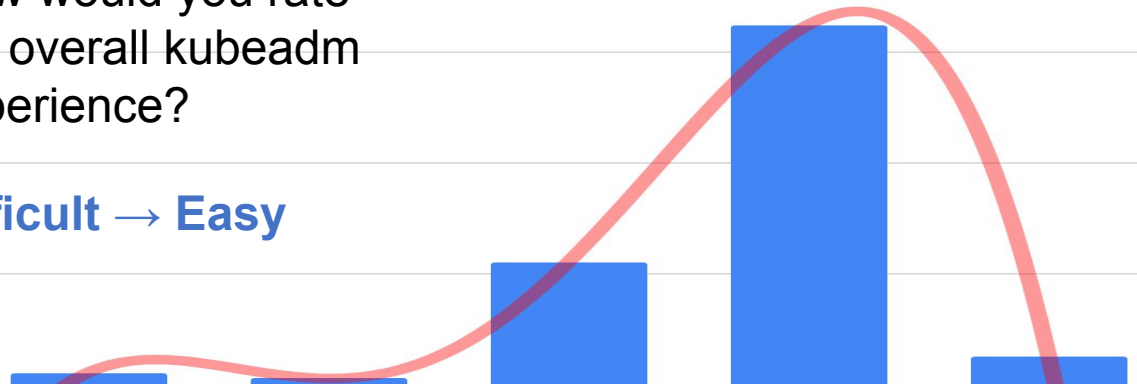
kubeadm is built to be part of a higher-level solution



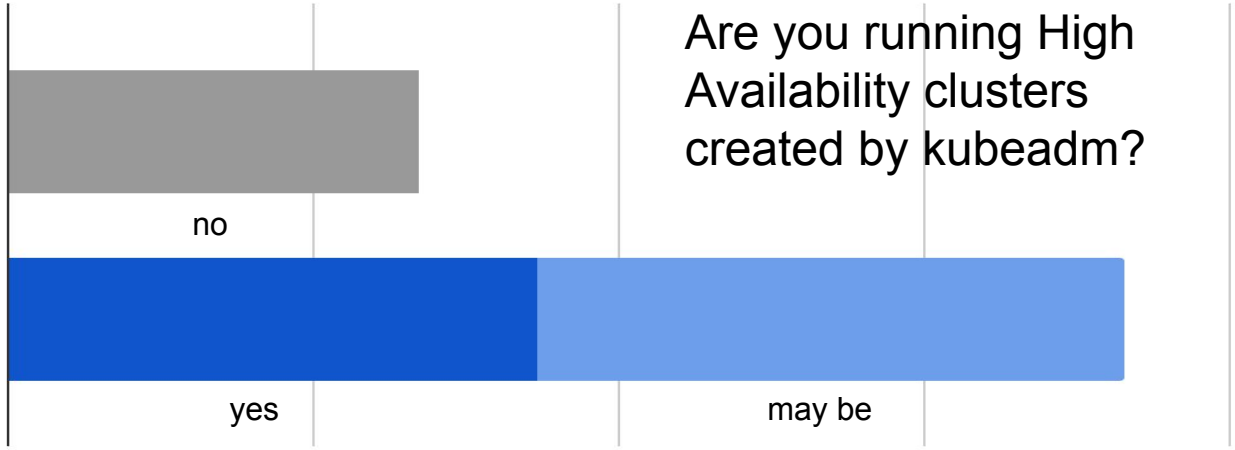
kubeadm Survey

How would you rate the overall kubeadm experience?

Difficult → Easy



Are you running High Availability clusters created by kubeadm?



kubeadm - Deep Dive

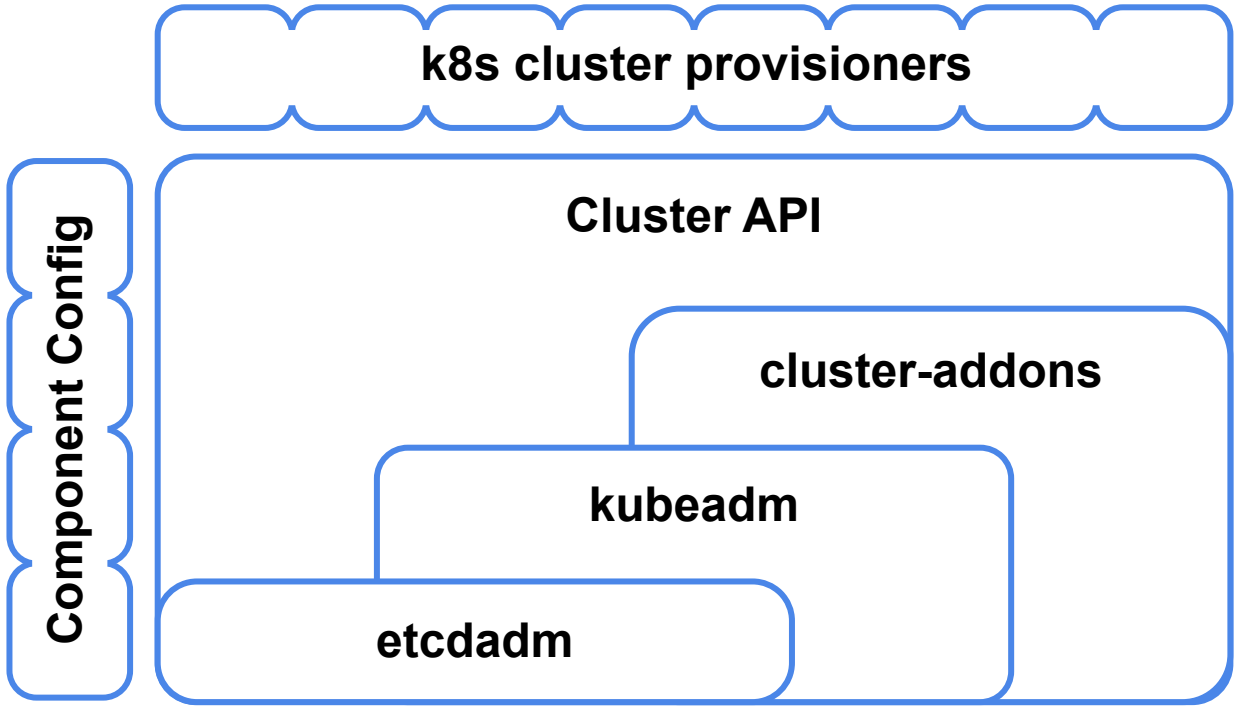
Wednesday, May 22 • 16:45 - 17:20 **(Lubomir Ivanov & Fabrizio Pandini)**

In this deep dive, we will take a look at recent changes in kubeadm, examine how kubeadm is going to implement support for high availability clusters, and finally peek through the window to see what will come next.

Cluster API

- The What and the Why of Cluster API “To make the management of (X) clusters across (Y) providers simple, secure, and configurable.”
  - “How do I provision all the other infrastructure I need for a Kubernetes cluster (load balancers, VPC, etc.)?”
  - “How do I manage other lifecycle events across that infrastructure (upgrades, deletions, etc.)?”
  - “How can I manage any number of clusters in a similar fashion to how I manage deployments in Kubernetes?”
  - “How can we control all of this via an API?”

Cluster API



Tools atop of Cluster API

- kops
- kubicorn
- Multiple control plane managers
  - SAP Gardener
  - KaaS layers

Cluster API Overview

- With Kubernetes we manage our applications declaratively
  a. Why not for the cluster itself?
- With the Cluster API, we can declaratively define the desired cluster state
  a. Operator implementations reconcile the state
  b. Use Spec & Status like the rest of k8s
  c. Common management solutions for e.g. upgrades, autoscaling and repair
  d. Allows for “GitOps” workflows

```yaml
apiVersion: cluster.k8s.io/v1alpha1
kind: MachineDeployment
metadata:
  name: my-nodes
spec:
  replicas: 3
  selector:
    matchLabels:
      foo: bar
  template:
    metadata:
      labels:
        foo: bar
    spec:
      providerConfig:
        value:
          apiVersion: "baremetalconfig/v1alpha1"
          kind: "BareMetalProviderConfig"
          zone: "us-central1-f"
          machineType: "n1-standard-1"
          image: "ubuntu-1604-lts"
      versions:
        kubelet: 1.14.2
        containerRuntime:
          name: containerd
          version: 1.2.0
```

Cluster API - Deep Dive

KubeCon | CloudNativeCon Europe 2019

Thursday, May 23 • 11:05 - 11:40 **(Jason DeTiberus & Hardik Dodiya)**

In this deep dive, we will examine how the Cluster API simplifies the cluster management experience for cluster operators by enabling consistent machine management across environments, and bringing declarative upgrades to Kubernetes clusters.

WG Component Standard

- Problem 1: The core Kubernetes components are not consistent in
  - how they are configured
  - how they should be set up
  - what HTTP(S) endpoints they register
  - how they do (delegated) auth

- Problem 2: It’s pretty hard to write a k8s-like component with declarative config

- Solution: Factor common component-related code into a `k8s.io/component-base` toolkit repository. Make it easier to write a non-core component that follows the k8s style

ComponentConfig

- **Maintainability:**

  When $component's flag set grows over 50+ flags, configuring it becomes painful
- **Upgradability:**

  On upgrades, $component still works using versioned config vs. flags
- **Programmability:**

  Configuration expressed as JSON/YAML objects allows for consistent manipulation
- **Possibility:**

  Many types of config simply can't be expressed as simple key-value
- **Declarative:**

  OpenAPI information can easily be exposed / used for doc generation
- See Lucas' talk on this here: Configuring Your Kubernetes Cluster on the Next Level

ComponentConfig End Goal


```
$ kube-controller-manager --config config.yaml
```

```yaml
apiVersion: kubecontrollermanager.config.k8s.io/v1
kind: KubeControllerManagerConfiguration
controllers:
  csrSigning:
    clusterSigningCertFile: /some/path
  namespace:
    concurrentNamespaceSyncs: 5
  nodeLifecycle:
    enableTaintManager: true
```

Getting Involved!

Getting Involved!

SIG Cluster Lifecycle

- 100s of contributors across several companies
- We’re working on growing the contributor/reviewers pool
- We’re EMEA contributors friendly

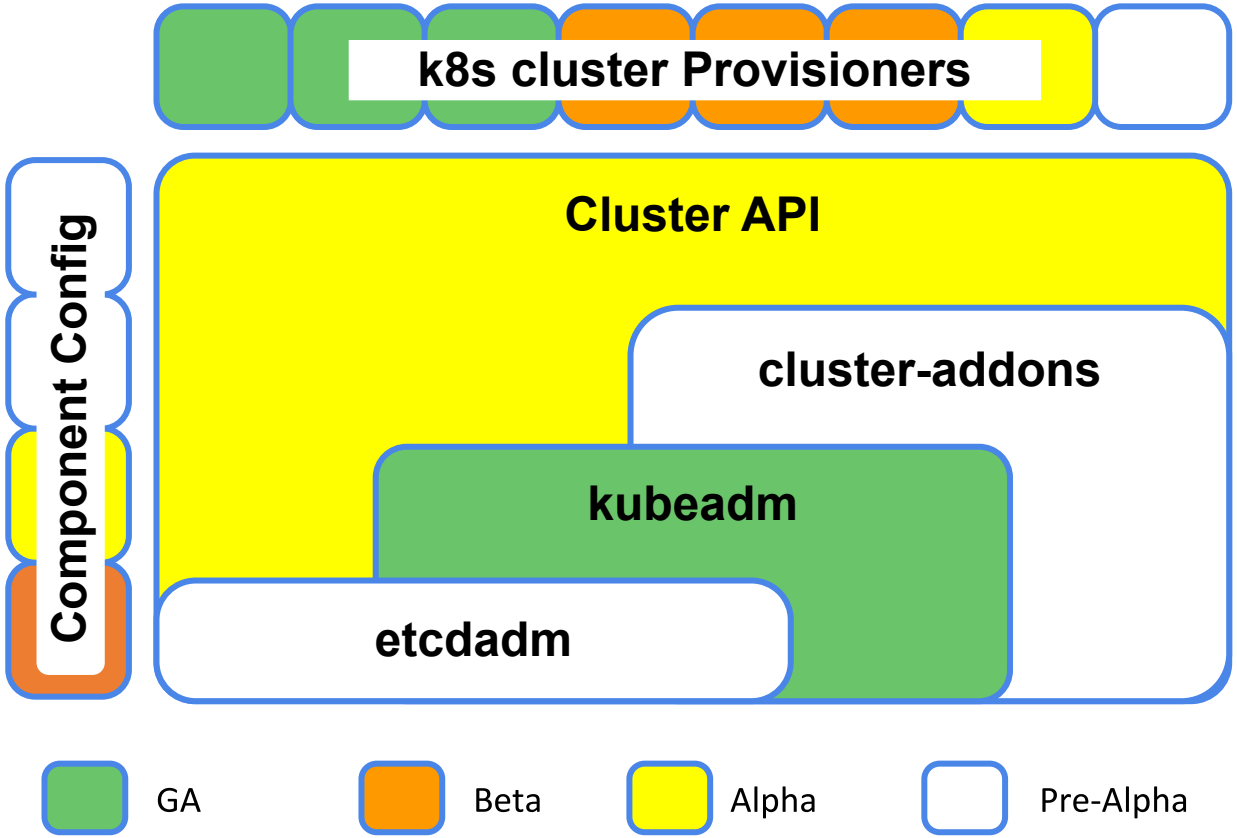
How can you Contribute

- SIG Cluster Lifecycle New Contributor Onboarding
- Look for “good first issue”, “help wanted” and “sig/cluster-lifecycle” labeled issues in our repositories (in k/k or in various project repository)
- Attend our Zoom meetings / be around on Slack
- We have “Office Hours” for our projects: weekly for kubeadm and Cluster API, bi-weekly for kops and kubespray
- Full list of SIG meetings and links to minutes and recordings can be found on SIG page
- Contributing to SIG Cluster Lifecycle documentation

The SCL Roadmap

We need your help!

There is still a lot of work to do in onder to get the full puzzle in place!



Other Logistics

- Follow the SIG Cluster Lifecycle YouTube playlist
- Check out the meeting notes for our weekly office hours meetings
- Join #sig-cluster-lifecycle, #kubeadm, #cluster-api, #kops-dev, #kops-users, #kubespray, #minikube, …channels
- Check out the kubeadm setup guide, reference doc and design doc
- Read how you can get involved, and watch the new contributor onboarding session!

Other SCL Talks

- Kubeadm Deep Dive - Wednesday, May 22 • 16:45 - 17:20
- Cluster API Deep Dive - Thursday, May 23 • 11:05 - 11:40
- Kops Deep Dive - Thursday, May 23 • 11:55 - 12:30
- Kubespray Deep Dive - Thursday, May 23 • 14:50 - 15:25

Thank you!
Q/A