Cluster API

A Kubernetes subproject to simplify cluster lifecycle management

Katie Gamanji - CNCF Technical Oversight Committee & Cloud Platform Engineer, American Express

Naadir Jeewa - Maintainer, Cluster API Provider AWS & Senior Member of Technical Staff, VMware

11 June 2020

Katie Gamanji

Cloud Platform Engineer

@American Express

Technical Oversight Committee

@CNCF

@k_gamanji

katie-gamanji

Naadir Jeewa

Senior Member of Technical Staff
VMware

Maintainer, Kubernetes Cluster API Provider AWS

randomvariable

jeewan@vmware.com

vmware®

Agenda

Kubernetes Lifecycle Management with Cluster API

Cluster API overview

How Cluster API delivers cloud independence

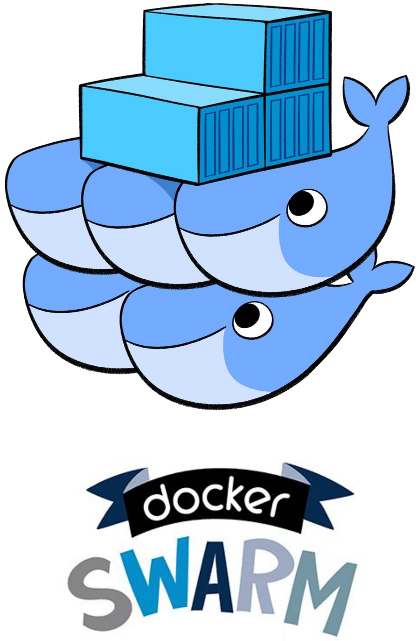
How controllers enable declarative infrastructure

Demo

Roadmap

Cluster API and you

Container Orchestrators



58%

—

Production Systems

2K

—

Contributors

23K

—

KubeCon Attendees

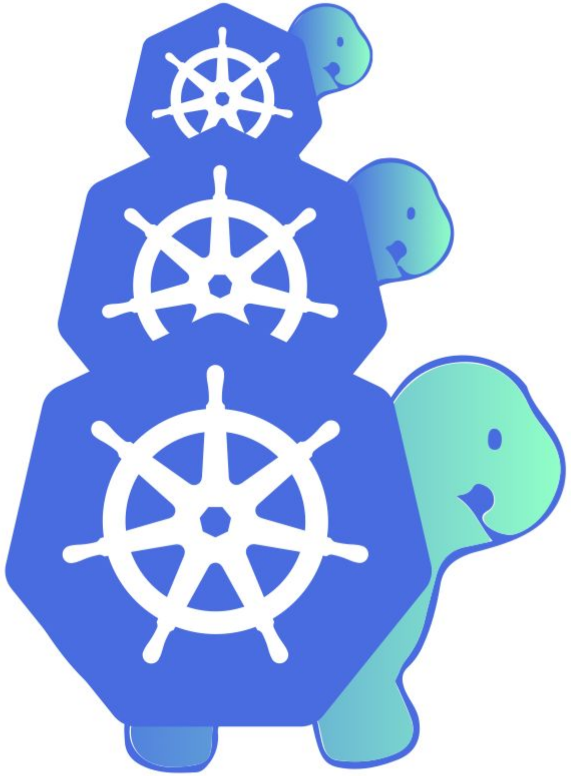
HELM
gRPC
mæsh

CLOUD NATIVE Landscape

gRPC
HELM
mæsh

ClusterAPI

Cluster Provisioning



Bootstrap Tools

kubeadm

KUBESPRAY

kops

TECTONIC

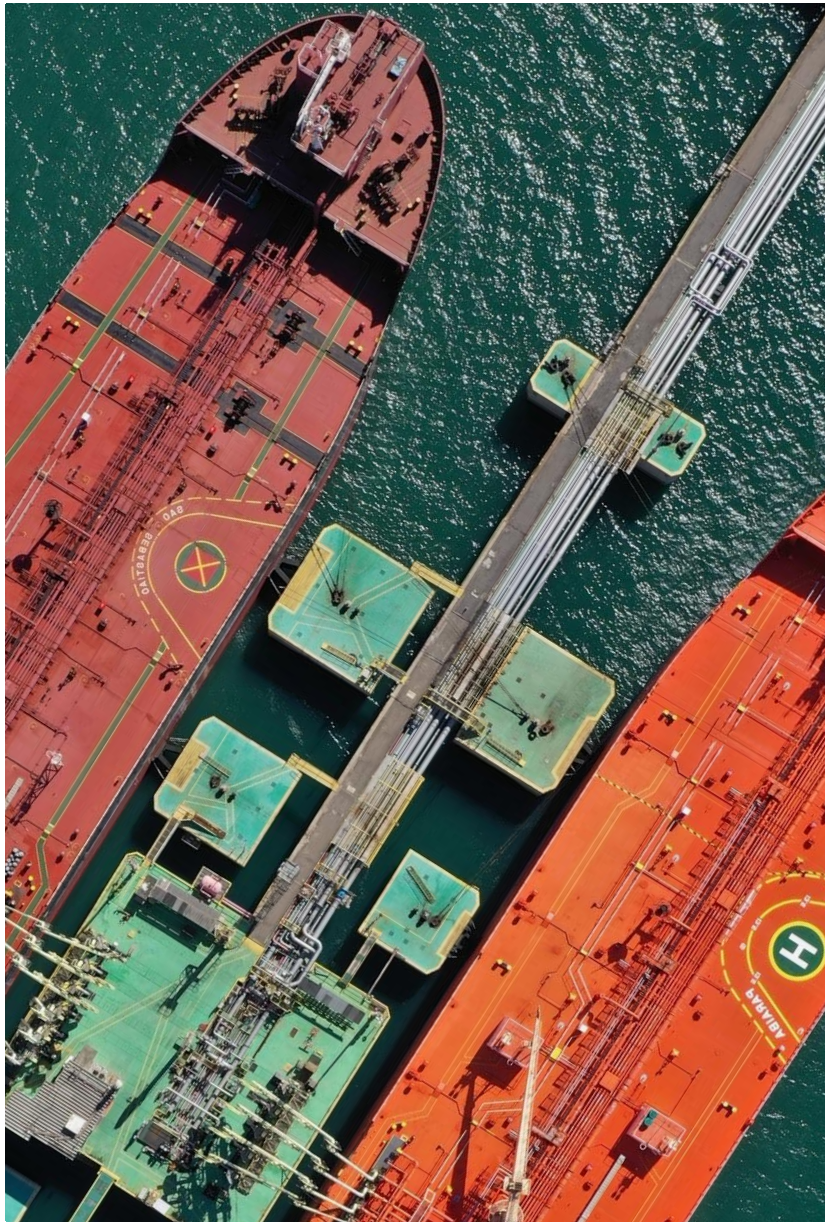
Challenges

Bootstrap Providers

- Cross-cloud
- Tool migration
- China

Cluster API

> “Provision of declarative APIs for cluster creation, configuration, and management.”



ClusterAPI

SIG-cluster-lifecycle:

- Initial release: April 2019
- API: **v1alpha3**

aws

Azure

packet
AN EQUINIX COMPANY

DigitalOcean

Google Cloud

TALOS

openstack

IBM Cloud

Tencent 腾讯

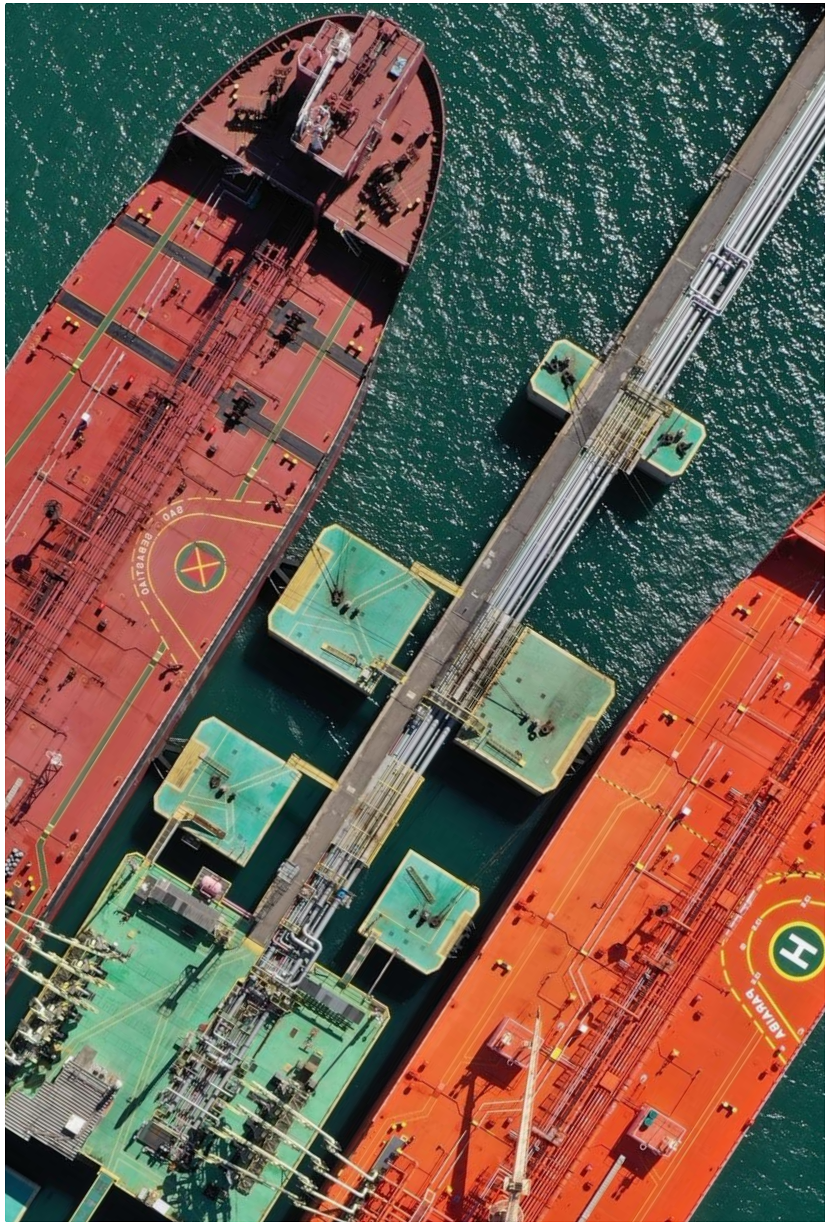
BAIDU AI CLOUD

Alibaba Cloud

vmware
vSphere

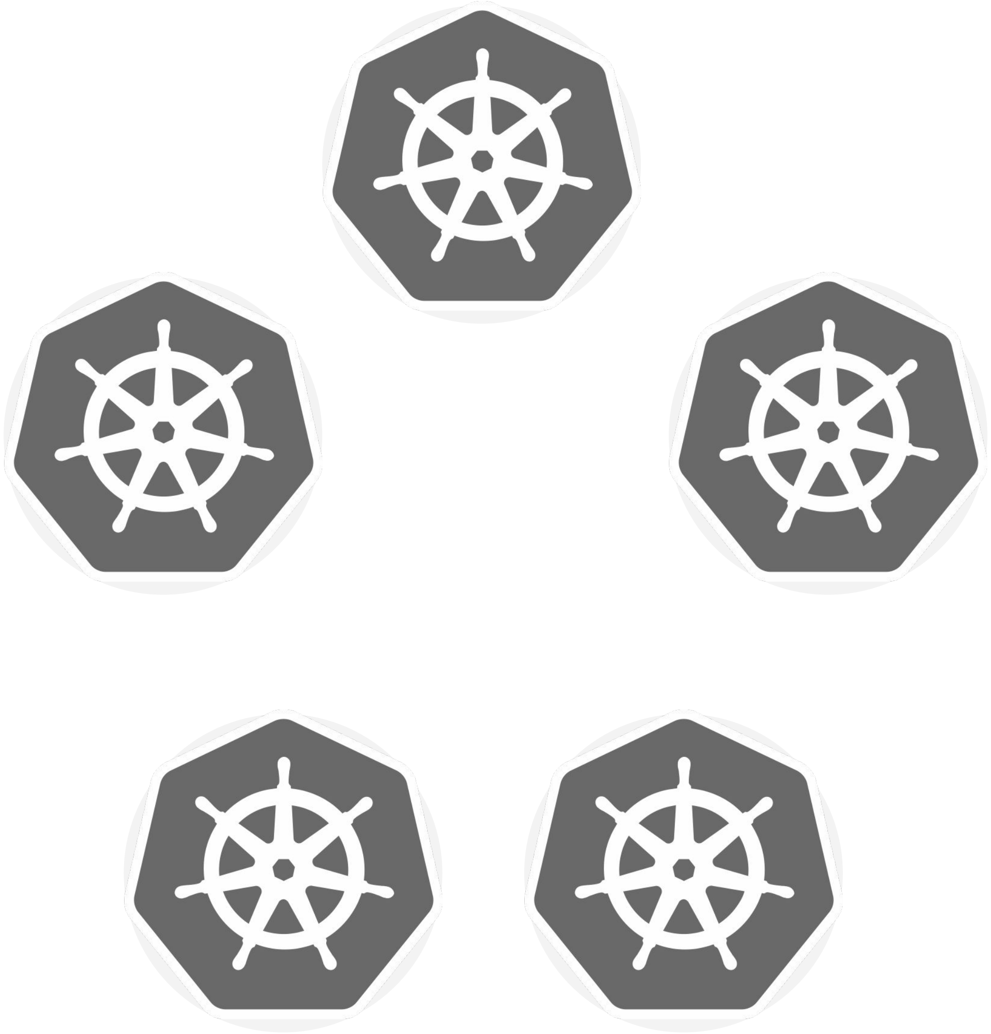
EXOSCALE

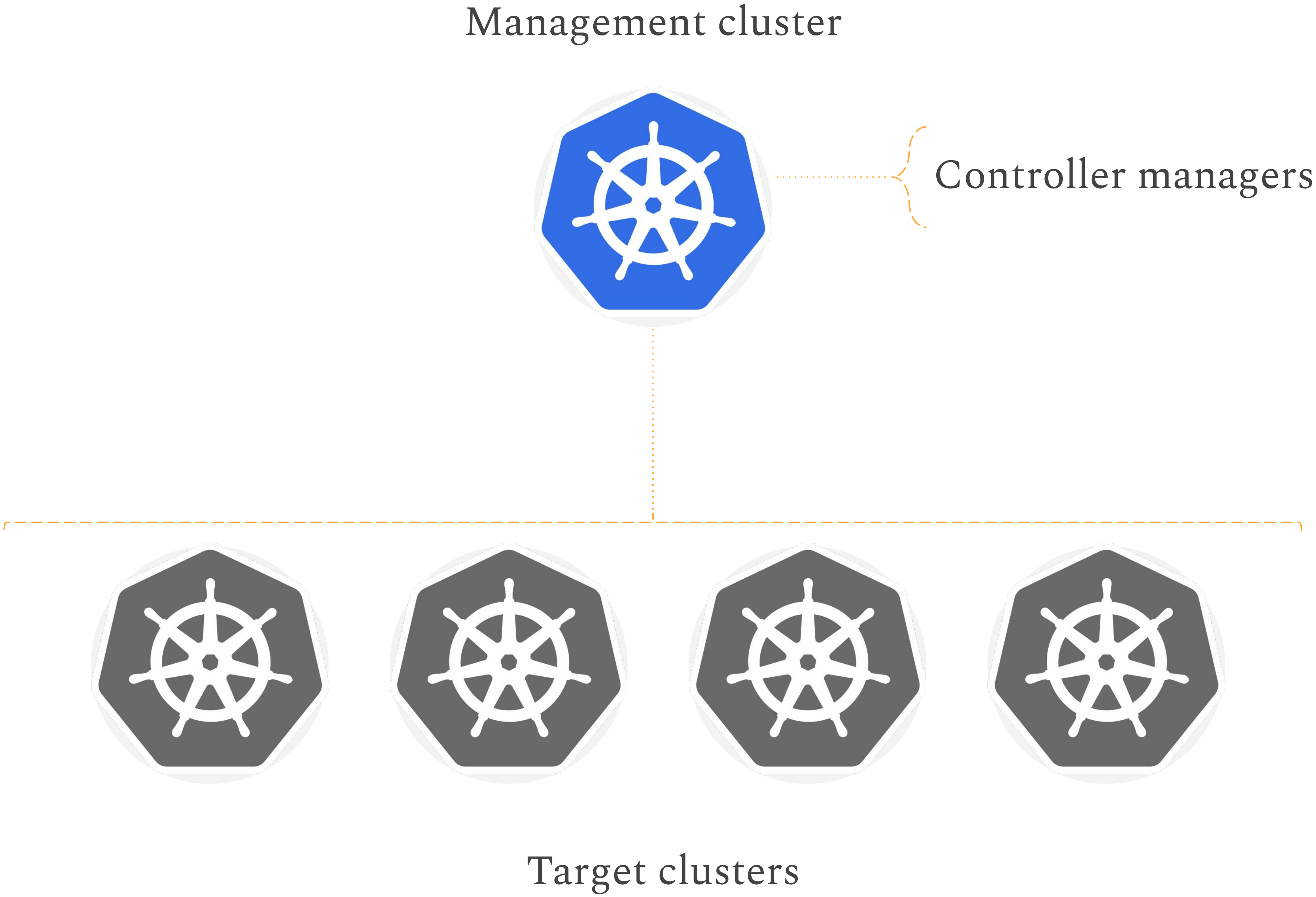
`cluster-api.sigs.k8s.io/reference/providers.html`



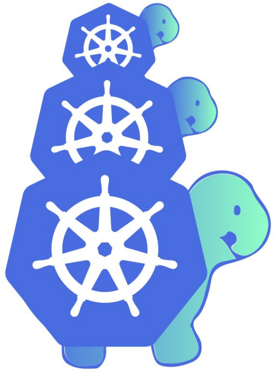
1 / 4

—




Management cluster
Controller managers
Target clusters

3 / 4: Controller Managers



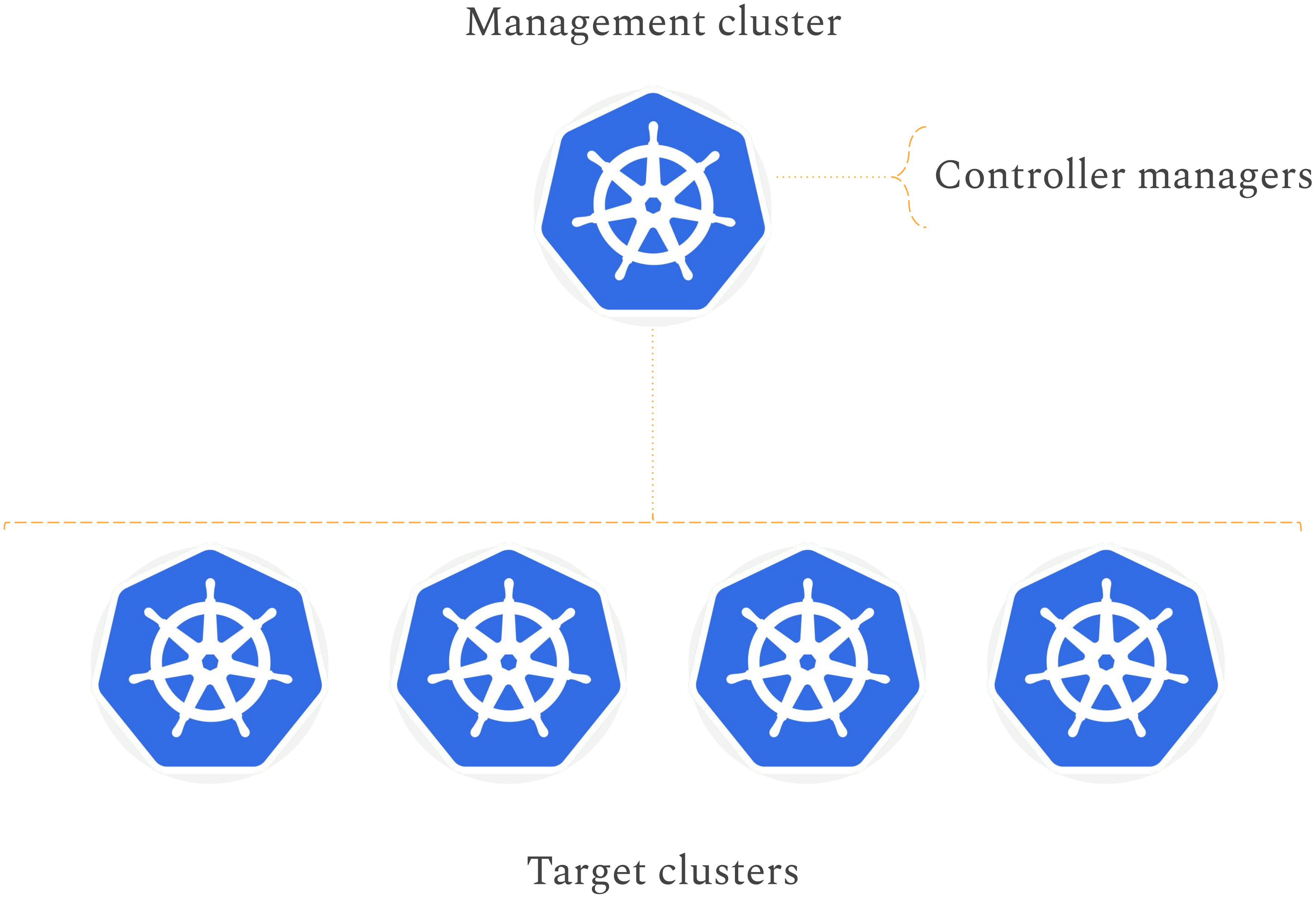
ClusterAPI
CRDs

+

Bootstrap
Providers

+

Infrastructure
Providers


Management cluster
Controller managers
Target clusters

KUBESPRAY

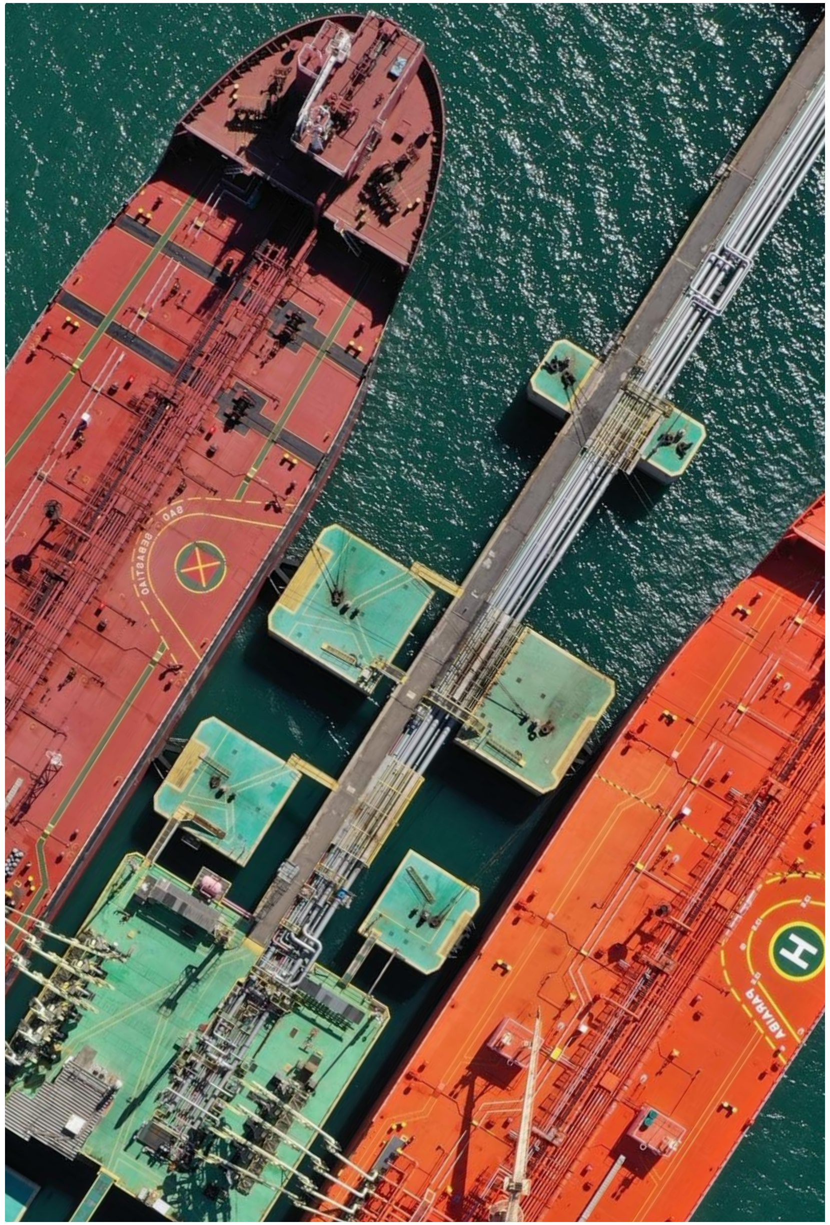
```
├── bastion-ssh-config
│    ├── tasks
│    └── templates
├── bootstrap-os
│    ├── defaults
│    ├── files
│    └── tasks
├── container-engine
│    ├── containerd
│    ├── cri-o
│    ├── docker
│    ├── docker-storage
│    └── meta
├── etcd
│    ├── defaults
│    ├── handlers
│    ├── meta
│    ├── tasks
│    └── templates
```

```
├── kubernetes
│    ├── client
│    ├── kubeadm
│    ├── master
│    ├── node
│    ├── node-label
│    ├── preinstall
│    └── tokens
├── kubernetes-apps
│    ├── ansible
│    ├── cloud_controller
│    ├── cluster_roles
│    ├──
container_engine_accelerator
│    ├── csi_driver
│    ├── external_provisioner
│    ├── helm
│    ├── ingress_controller
│    ├── meta
```

```
│    ├── metrics_server
│    ├── network_plugin
│    ├── persistent_volumes
│    ├── policy_controller
│    ├── registry
│    └── rotate_tokens
├── kubespray-defaults
│    ├── defaults
│    ├── meta
│    └── tasks
├── network_plugin
│    ├── calico
│    ├── canal
│    ├── cilium
│    ├── cloud
│    ├── cni
│    ├── flannel
│    ├── kube-router
│   [...]
```

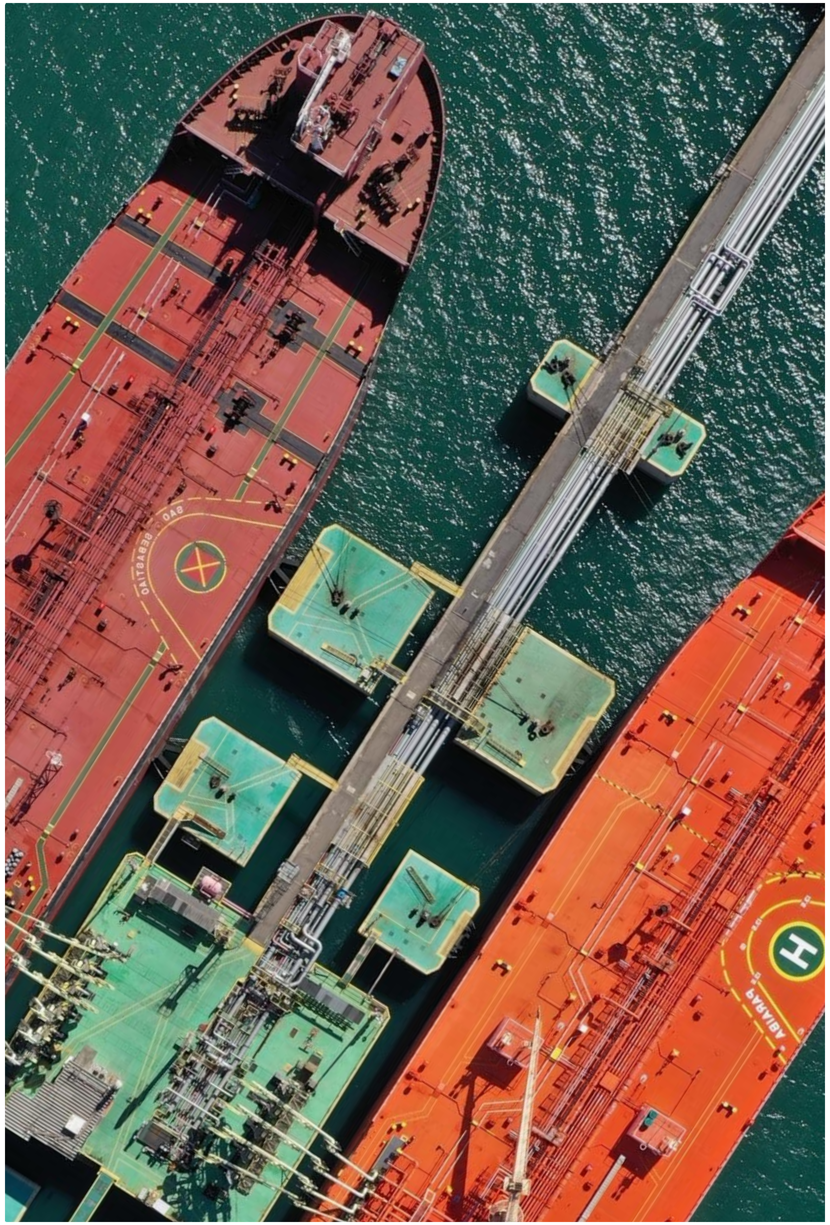
aws

```
apiVersion: ${{ v1alpha3 }}
kind: Cluster
metadata:
  name: test-cluster
spec:
   clusterNetwork:
    pods:
      cidrBlocks: ["192.168.0.0/16"]
  infrastructureRef:
    apiVersion: ${{ v1alpha3 }}
    kind: AWSCluster
    name: test-cluster
  controlPlaneRef:
    kind: KubeadmControlPlane
    [...]
```



aws

```
apiVersion: ${{ v1alpha3 }}
kind: Cluster
metadata:
  name: test-cluster
spec:
   clusterNetwork:
    pods:
      cidrBlocks: ["192.168.0.0/16"]
  infrastructureRef:
    apiVersion: ${{ v1alpha3 }}
    kind: AWSCluster
    name: test-cluster
  controlPlaneRef:
    kind: KubeadmControlPlane
    [...]
```



aws

```
apiVersion: ${{ v1alpha3 }}
kind: Cluster
metadata:
  name: test-cluster
spec:
   clusterNetwork:
    pods:
      cidrBlocks: ["192.168.0.0/16"]
  infrastructureRef:
    apiVersion: ${{ v1alpha3 }}
    kind: AWSCluster
    name: test-cluster
  controlPlaneRef:
    kind: KubeadmControlPlane
    [...]
```

```
apiVersion: ${{ v1alpha3 }}
kind: AWSCluster
metadata:
  name: test-cluster
spec:
  region: eu-central-1
  sshKeyName: default
```

```
apiVersion: ${{ v1alpha3 }}
kind: Cluster
metadata:
  name: test-cluster
spec:
   clusterNetwork:
    pods:
      cidrBlocks: ["192.168.0.0/16"]
  infrastructureRef:
    apiVersion: ${{ v1alpha3 }}
    kind: GCPCluster
    name: test-cluster
  controlPlaneRef:
    kind: KubeadmControlPlane
    [...]
```

```
apiVersion: ${{ v1alpha3 }}
kind: GCPCluster
metadata:
  name: test-cluster
spec:
  region: europe-west3
  project: CAPI
  network:
    name: default-capi
```

```
apiVersion: ${{ v1alpha3 }}
kind: Cluster
metadata:
 name: test-cluster
spec:
 clusterNetwork:
  pods:
   cidrBlocks: ["192.168.0.0/16"]
 infrastructureRef:
  apiVersion: ${{ v1alpha3 }}
  kind: AzureCluster
  name: test-cluster
 controlPlaneRef:
  kind: KubeadmControlPlane
  [...]
```

```
apiVersion: ${{ v1alpha3 }}
kind: AzureCluster
metadata:
 name: test-cluster
spec:
 location: westeurope
 networkSpec:
  vnet:
   name: default-capi
 resourceGroup: capi
```

ClusterAPI

Takeaways

Building blocks

Cloud agnostic

Community

What is a

Kubernetes Cluster?

Infrastructure

Networking
Firewall Rules
Servers
Load balancer

Kubernetes components

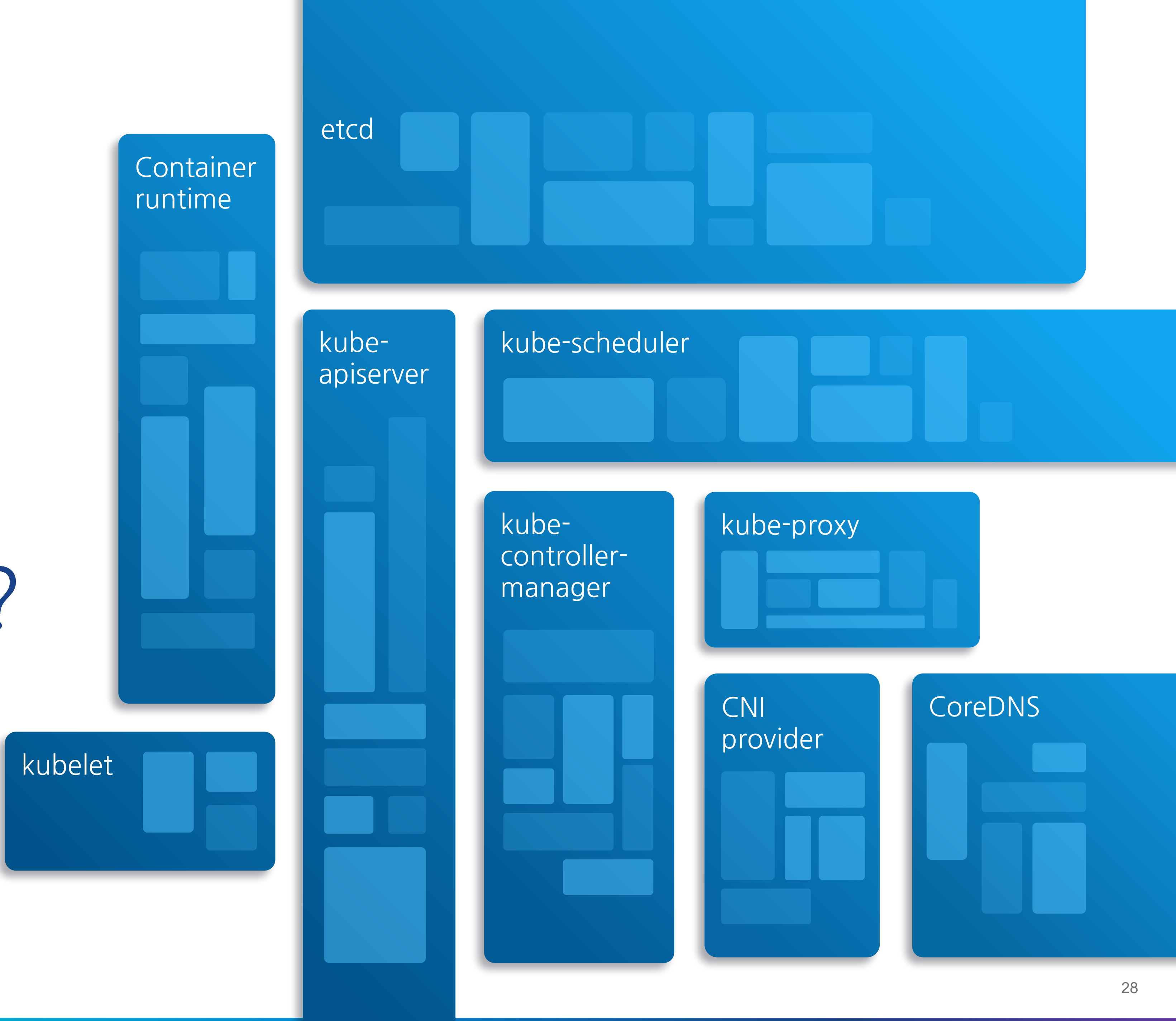
Add-ons

Disaster recovery (Velero)
Ingress (Contour, Gimbal)
Registry (Harbor)
Conformance (Sonobuoy)

User Apps

Option 2

Kubernetes Components: Bootstrapping?



etcd

Container runtime

kube-apiserver

kube-scheduler

kube-controller-manager

kube-proxy

kubelet

CNI provider

CoreDNS

A PROJECT OF
SIG CLUSTER LIFECYCLE

Kubeadm

Consistent Kubernetes
cluster bootstrapping

github.com/kubernetes/kubeadm

Kubernetes Cluster

Provisioning?

Infrastructure

Networking
Firewall Rules
Servers
Load balancer

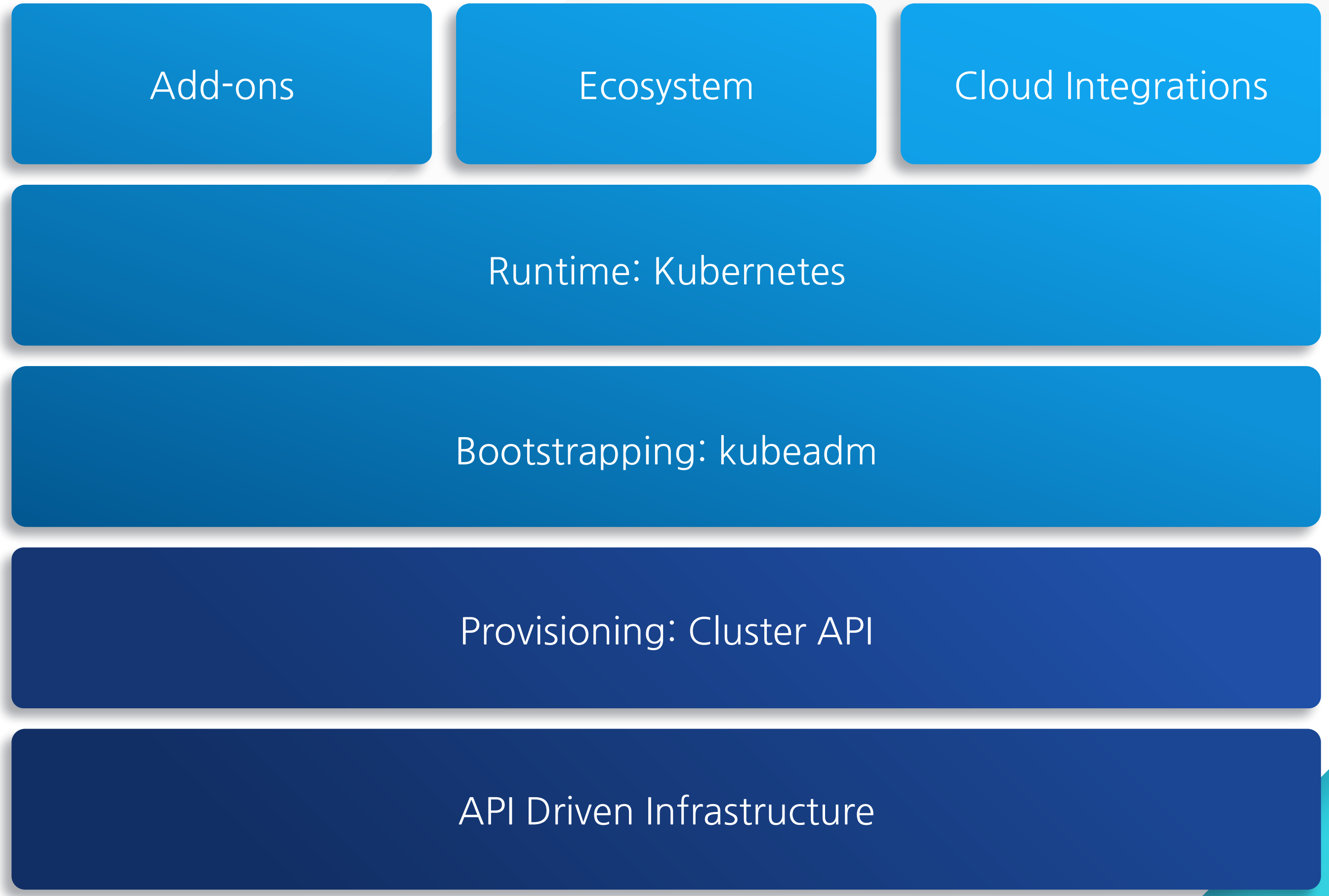
Kubernetes components

Add-ons

Disaster recovery (Velero)
Ingress (Contour)
Registry (Harbor)
Conformance (Sonobuoy)

User Apps

Kubernetes in Layers



Add-ons

Ecosystem

Cloud Integrations

Runtime: Kubernetes

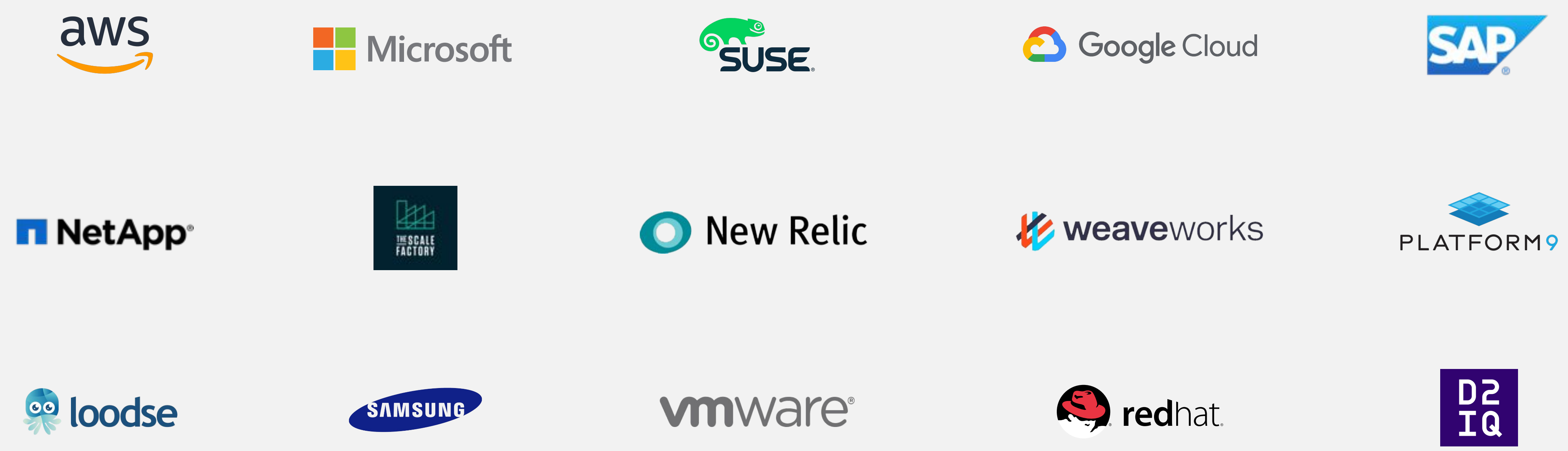
Bootstrapping: kubeadm

Provisioning: Cluster API

API Driven Infrastructure

Cluster API

Community Participation



bit.ly/capidevstats

Cluster API

Abstractions

Kubernetes

Pod

ReplicaSet

Deployment

Cluster API

Machine

MachineSet

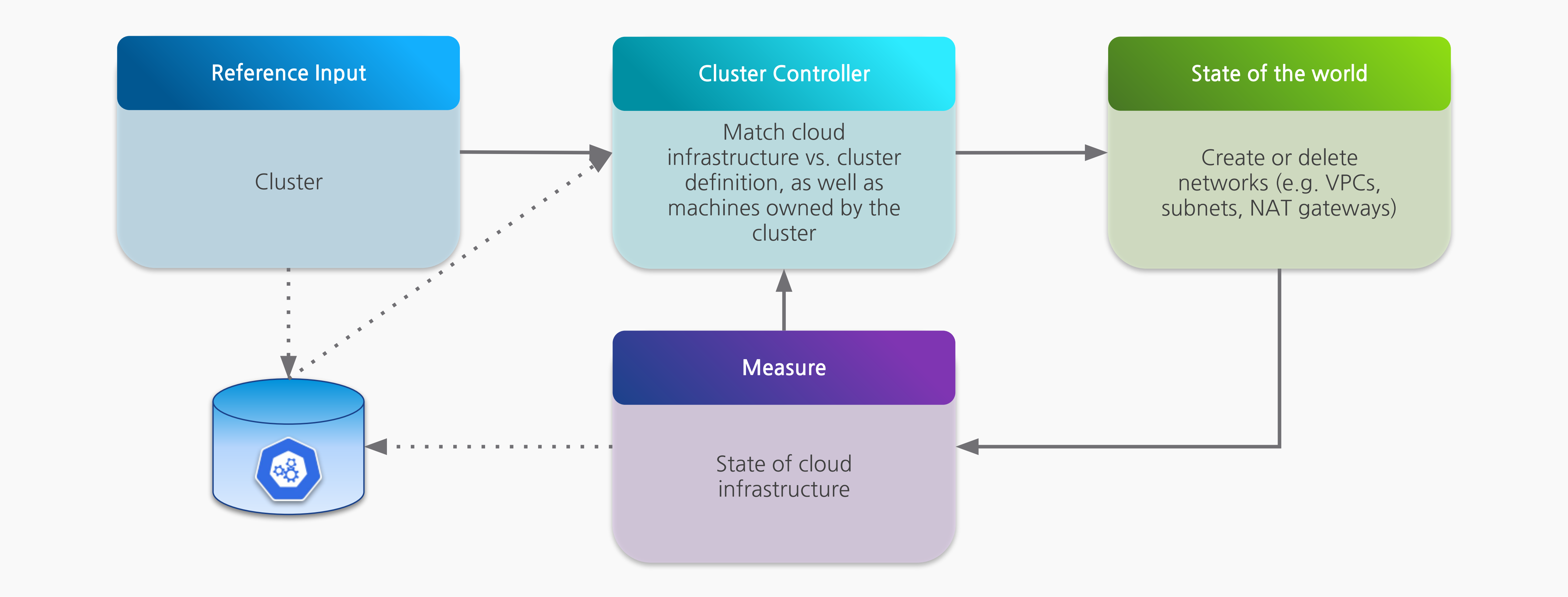
MachineDeployment

Cluster

ControlPlane

Cluster Controller

How Cluster API uses continuous reconciliation



Reference Input

Cluster

Cluster Controller

Match cloud infrastructure vs. cluster definition, as well as machines owned by the cluster

State of the world

Create or delete networks (e.g. VPCs, subnets, NAT gateways)

Measure

State of cloud infrastructure

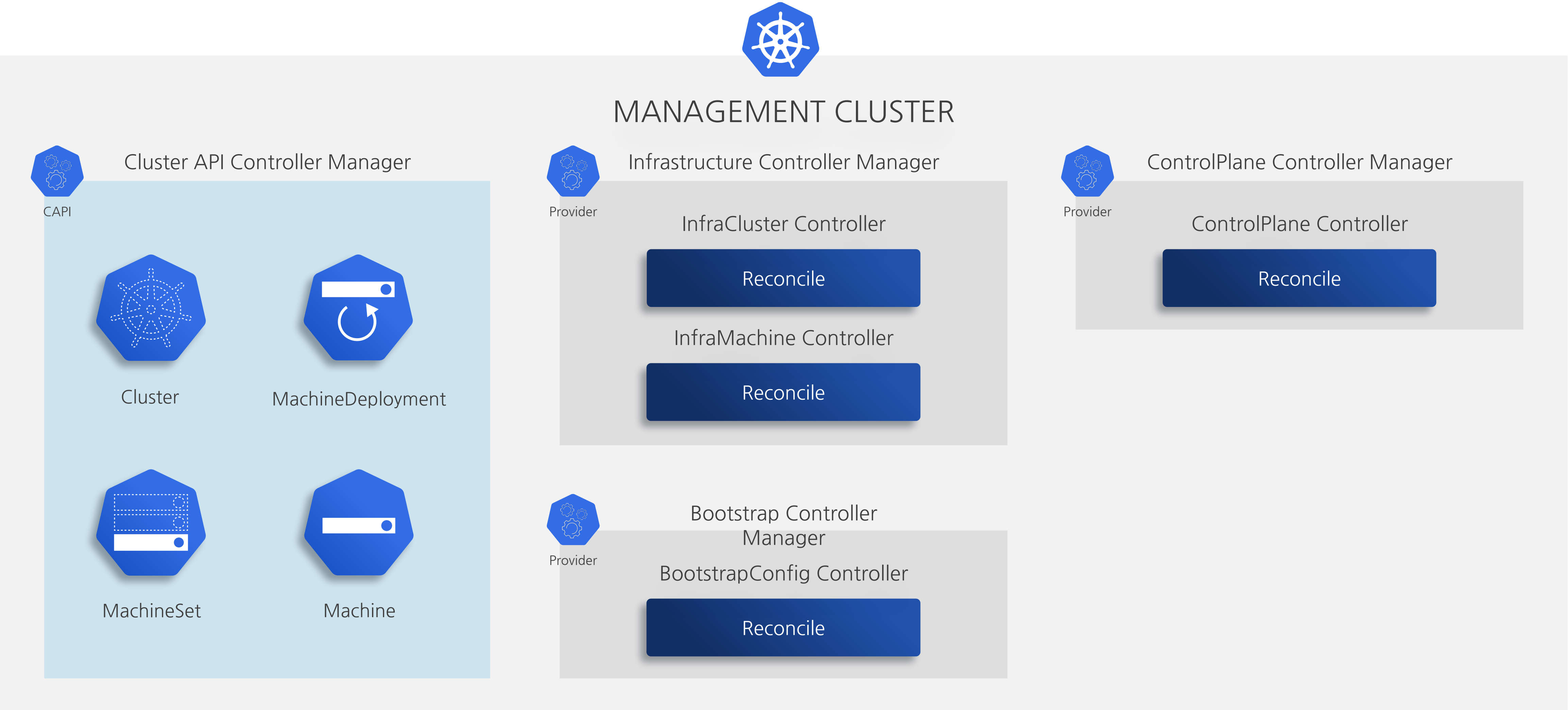
Cluster API

Current Architecture

▢ Kubebuilder

▢ Cluster API

■ Provider



MANAGEMENT CLUSTER

Cluster API Controller Manager

CAPI

Cluster

MachineDeployment

MachineSet

Machine

Infrastructure Controller Manager

Provider

InfraCluster Controller

Reconcile

InfraMachine Controller

Reconcile

Bootstrap Controller Manager

Provider

BootstrapConfig Controller

Reconcile

ControlPlane Controller Manager

Provider

ControlPlane Controller

Reconcile

Cluster Creation

Management Cluster

Install CRDs, controllers, providers into management cluster

Create objects

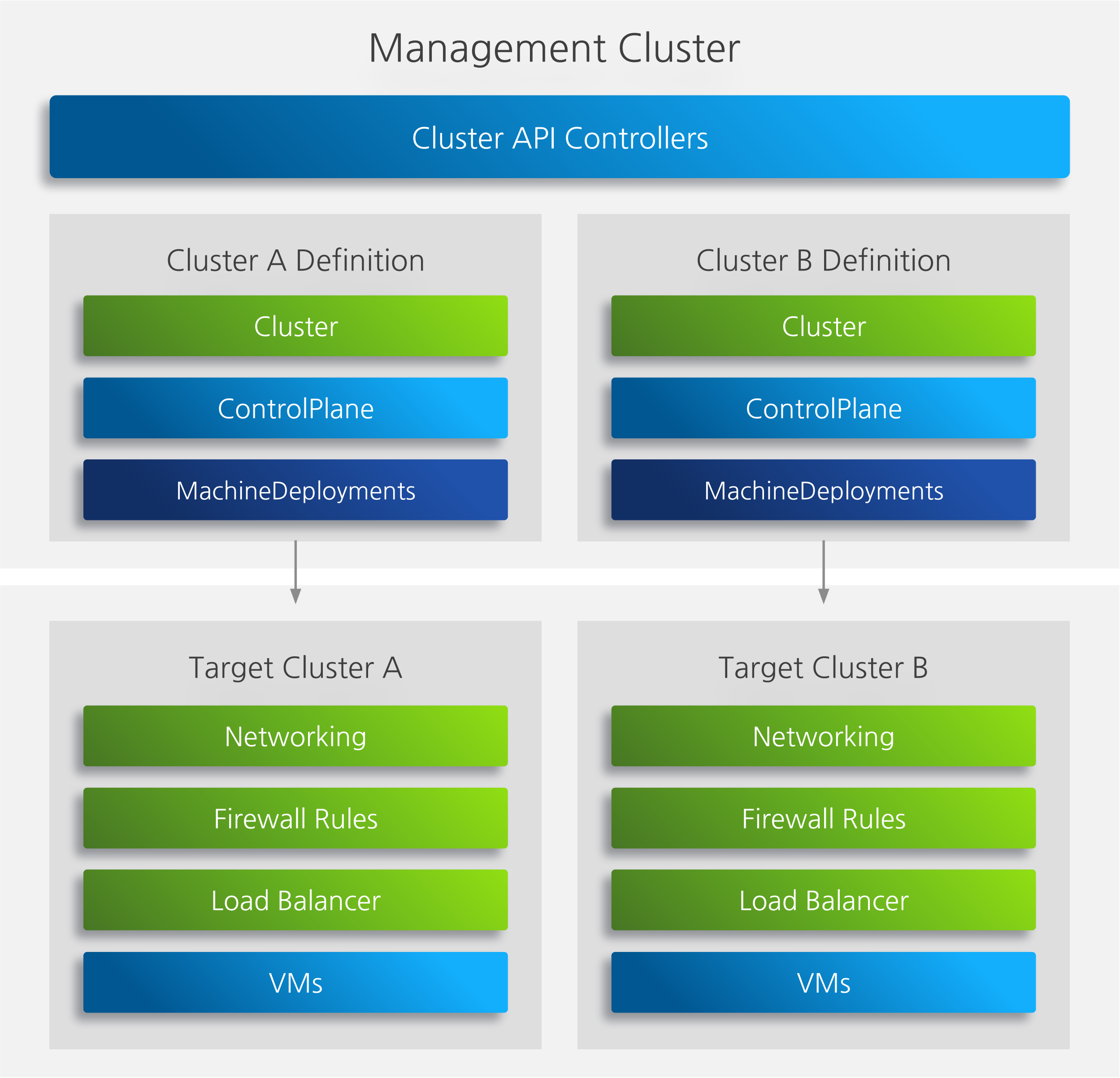
Cluster

ControlPlane

Add-ons (CNI)

MachineDeployment(s) for worker nodes

Cluster API creates infrastructure and bootstraps target cluster



Cluster Creation

Workflow

Start with “bootstrap” cluster (kind)

Install CRDs and controllers into bootstrap cluster

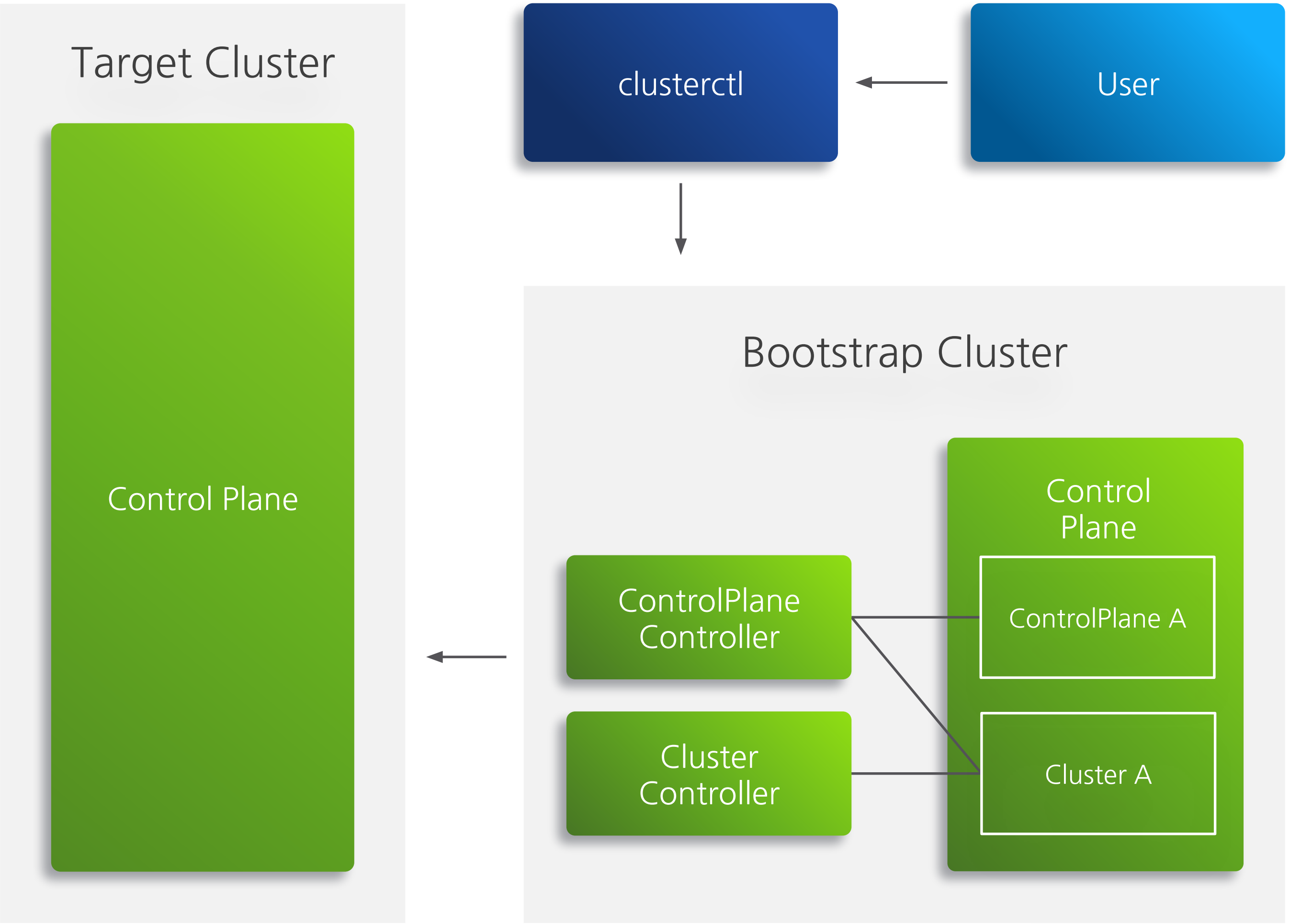
Create objects (cluster, control panel, etc.)

Creates infrastructure and bootstraps target cluster

Copy Cluster API and objects into target cluster

Delete kind bootstrap cluster

Use target cluster for all future management



clusterctl

Command-line interface for unified lifecycle management

`clusterctl init`

Installs the cluster API components in target cluster to make it into a management cluster

`clusterctl upgrade`

Upgrades cluster API and provider components installed in the management cluster

`clusterctl delete`

Deletes provider components from the management cluster

`clusterctl config cluster`

Generates workload cluster manifest based on provided cluster template

`clusterctl move`

Moves Cluster API objects between management clusters

Cluster Templates

Predefined workload cluster configurations

Enables infrastructure providers and Cluster API operators to provide recommended deployment configurations for creating workload clusters.

Example templates could include:

- Highly available control plane for production clusters
- Development setup with single node control plane
- GPU-enabled nodes for machine learning workloads
- High memory nodes for memory intensive workloads

Demo

Cluster API Roadmap

Long Term Priorities

Extending control plane lifecycle management to offer more flexibility around upgrade strategies, making configuration changes

Improving observability and visibility into Cluster API for end-users

Working towards Cluster API beta to stabilize the API

Project Governance

Building community muscle for the long term

Release cadence

Reduce scope to iterate faster

At least every 6 months with public planning

v0.3.0 released in March 2020

Communication and collaboration through KEPs

Many alphas and betas until APIs stabilize (similar to kubeadm)

Underlying controllers and providers can be robust, though!

What’s New in v0.3

Control Plane

Control Plane managed as a single entity instead of individual machines

Upgrade of the Control Plane nodes

User Experience

Lifecycle Management of Cluster API components via clusterctl

Developer and end-user documentation improvements

Testing

Testing framework to enable infrastructure and bootstrap providers to validate Cluster API behaviors

What’s New in v0.3

Machine Pools

Infrastructure-provider specific implementations for managed group of nodes

Machine Health Checks

Automated monitoring and removal of unhealthy nodes

Does not currently support KubeadmControlPlane; work to add support is under way

Failure Domains

Distribution of Control Plane nodes across infrastructure-specific failure domains

Cluster API and You

How to Get Involved

Join Us!

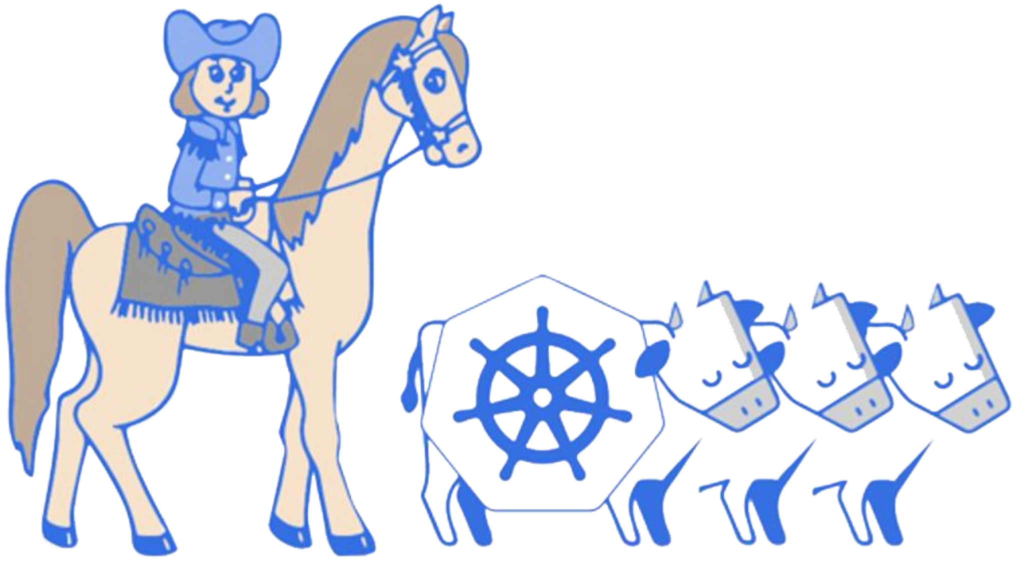
Cluster API discuss forum

sig-cluster-lifecycle Google Group to gain access to documents and calendars

Cluster API working group sessions

Weekly on Wednesdays
10:00 PT on Zoom

Chat with us on Slack: #cluster-api



sigs.k8s.io/cluster-api

AMERICAN EXPRESS

Thank You

vmware®

Appendix
Cluster API CRDs

Cluster API:

CRDs

- **Cluster**
- ControlPlane
- Machine
- MachineSet
- MachineDeployment

Spec

pod CIDR

service CIDR

service DNS suffix

Infrastructure provider-specific cluster reference

Status

API server endpoints

Error details

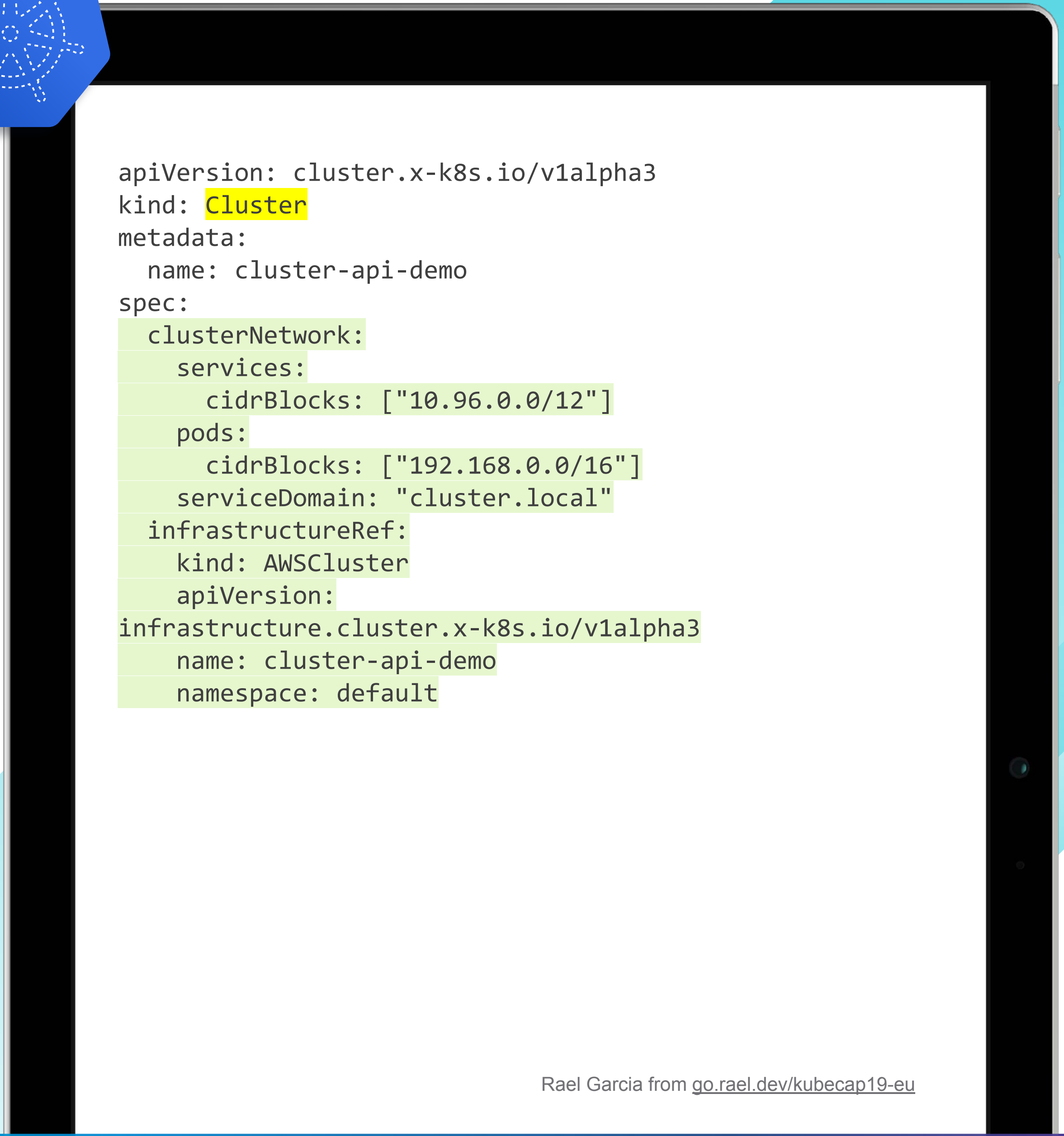
Cluster Example

Cluster

Cluster-wide configuration

Generic networking concepts like pod and service ranges or DNS domain

Providers can modify and override behavior where needed



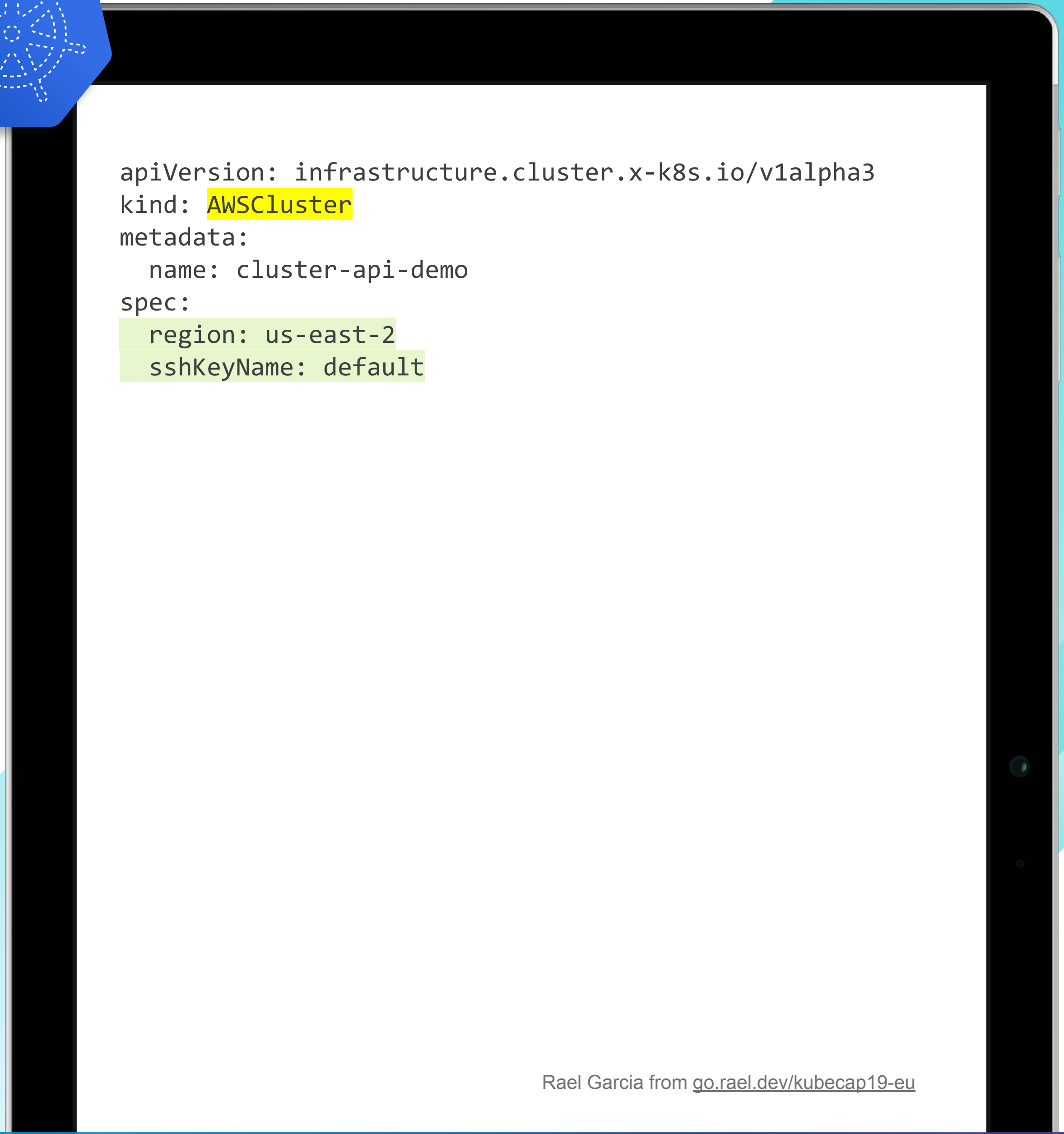
```
apiVersion: cluster.x-k8s.io/v1alpha3
kind: Cluster
metadata:
  name: cluster-api-demo
spec:
  clusterNetwork:
    services:
      cidrBlocks: ["10.96.0.0/12"]
    pods:
      cidrBlocks: ["192.168.0.0/16"]
    serviceDomain: "cluster.local"
  infrastructureRef:
    kind: AWSCluster
    apiVersion:
infrastructure.cluster.x-k8s.io/v1alpha3
    name: cluster-api-demo
    namespace: default
```

Rael Garcia from go.rael.dev/kubecap19-eu

InfraCluster Example

InfraCluster (AWSCluster)

Provider-specific cluster configuration



```
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSCluster
metadata:
  name: cluster-api-demo
spec:
  region: us-east-2
  sshKeyName: default
```

Rael Garcia from go.rael.dev/kubecap19-eu

Cluster API:

CRDs

- Cluster
- **ControlPlane**
- Machine
- MachineSet
- MachineDeployment

Spec

Desired replica count

Desired Kubernetes version

Infrastructure provider-specific template reference

Kubeadm Configuration for initializing & joining nodes to the control plane

Timestamp after which the control plane should be upgraded

Status

Number of replicas in the control plane

Number of replicas updated to match desired version

Number of replicas ready for the control plane

Number of unavailable replicas

Indication whether control plane has been initialized

Indication whether control plane is ready

ControlPlane Example

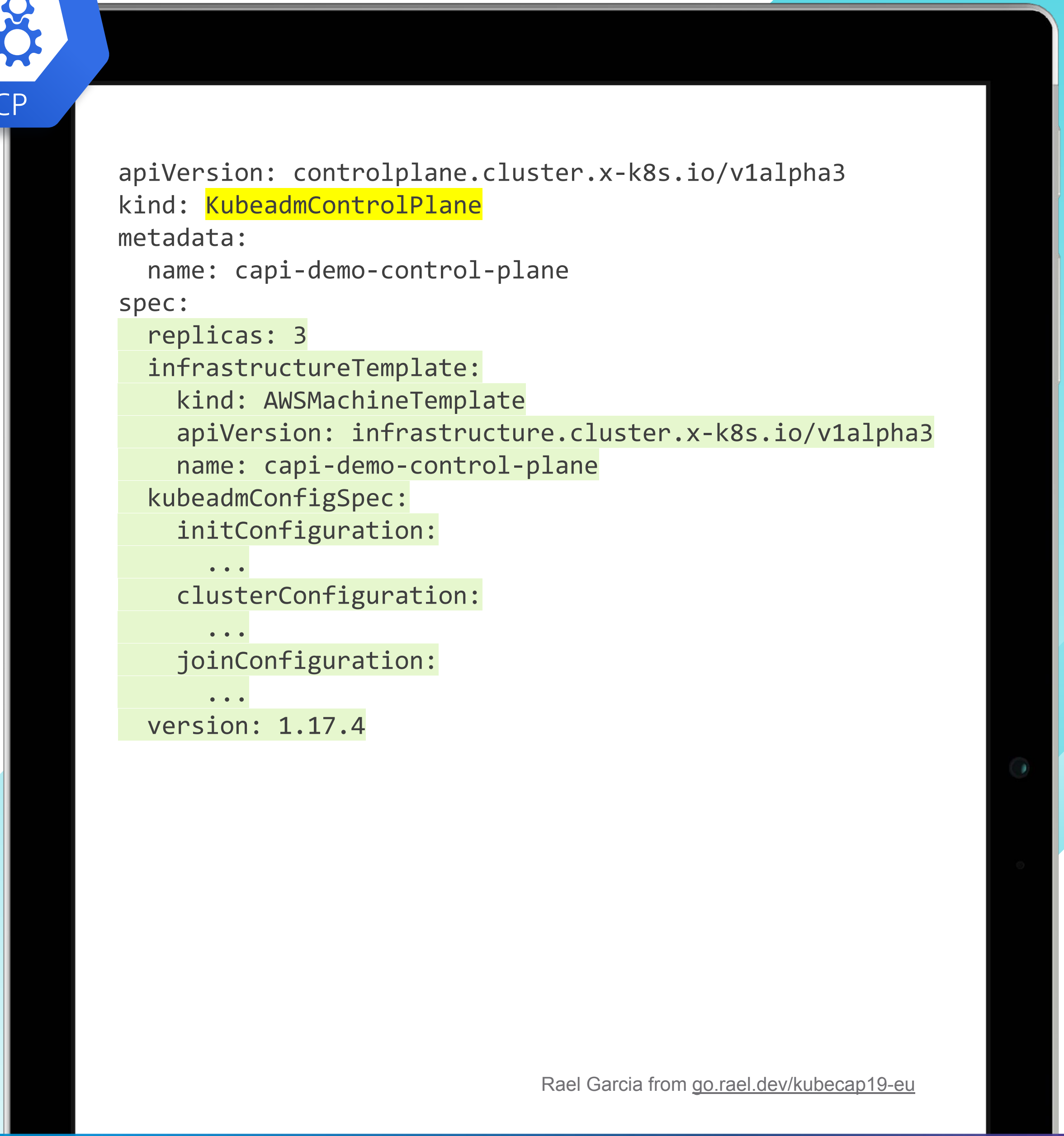
KubeadmControlPlane

Declarative control plane lifecycle management with Kubeadm

Replicas has the desired number of control plane machines

InfrastructureTemplate provides pluggable provider-specific machine definitions for control plane machines

KubeadmConfig provides means for configuring initialization, cluster and join configuration for control plane machines



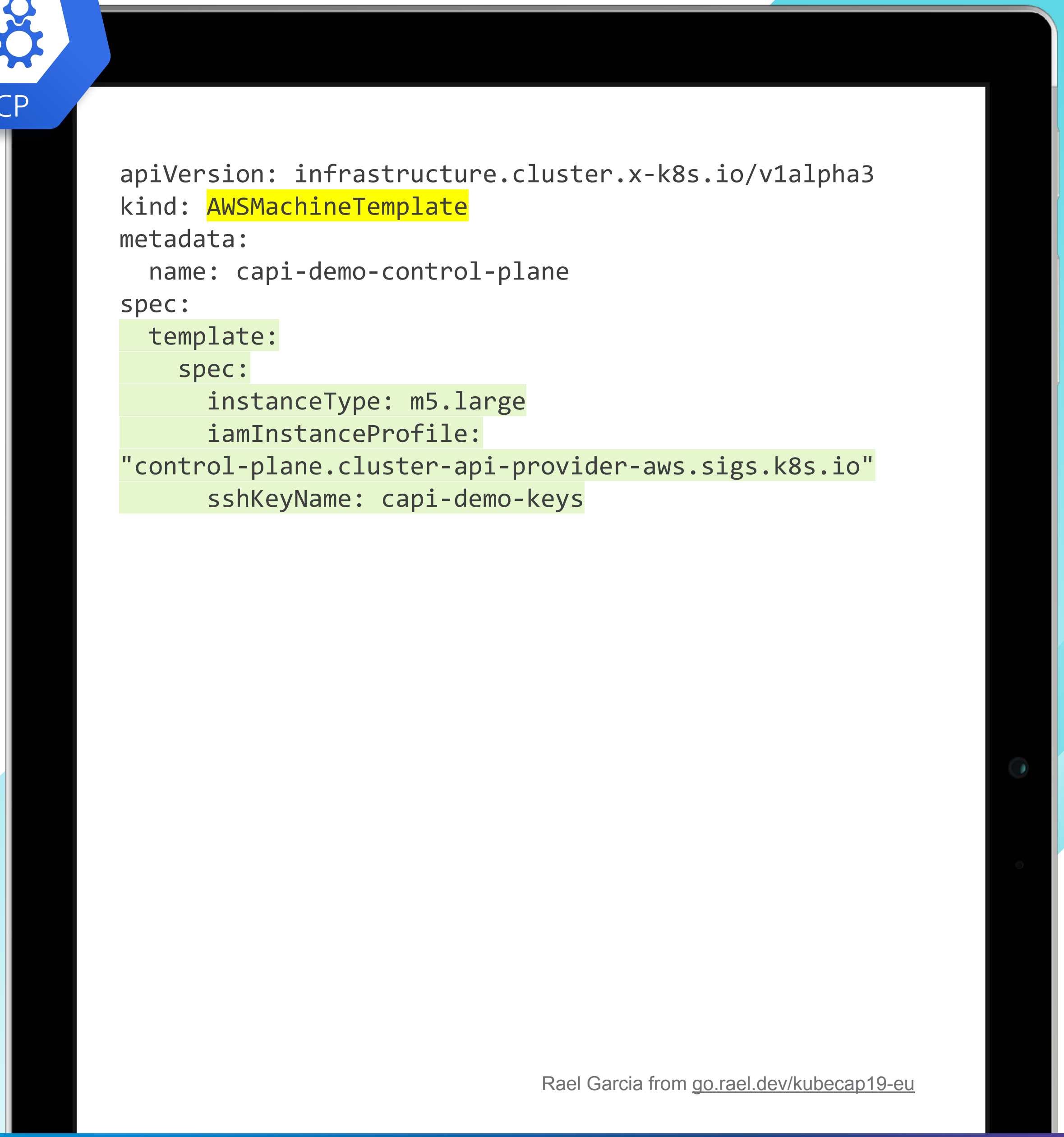
```yaml
apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
kind: KubeadmControlPlane
metadata:
  name: capi-demo-control-plane
spec:
  replicas: 3
  infrastructureTemplate:
    kind: AWSMachineTemplate
    apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
    name: capi-demo-control-plane
  kubeadmConfigSpec:
    initConfiguration:
      ...
    clusterConfiguration:
      ...
    joinConfiguration:
      ...
  version: 1.17.4
```

Rael Garcia from go.rael.dev/kubecap19-eu

ControlPlane Example

AWSMachineTemplate

Defines a template for AWS-specific machine creation



```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSMachineTemplate
metadata:
  name: capi-demo-control-plane
spec:
  template:
    spec:
      instanceType: m5.large
      iamInstanceProfile:
"control-plane.cluster-api-provider-aws.sigs.k8s.io"
      sshKeyName: capi-demo-keys
```

Rael Garcia from go.rael.dev/kubecap19-eu

Cluster API:

CRDs

- Cluster
- ControlPlane
- **Machine**
- MachineSet
- MachineDeployment

Spec

Kubernetes version

Infrastructure provider-specific reference

Provider ID

Bootstrap provider-specific configuration reference

Status

Node reference*

Last updated time

Kubernetes version

Error details

IP addresses

Phase

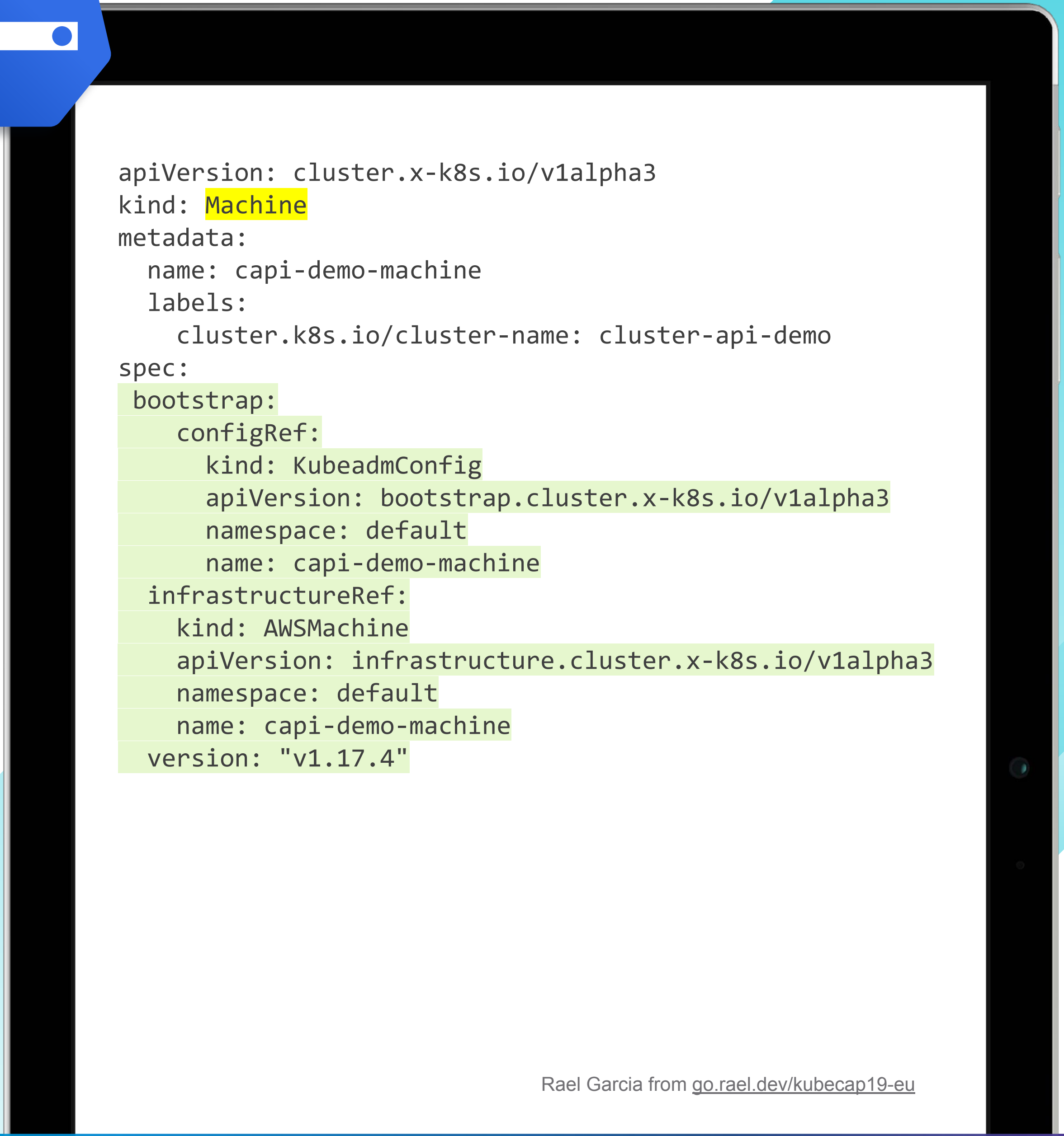
Machine Example

Machine

Configuration for a specific machine

Spec has the desired kubelet version

Providers can modify and override behavior where needed



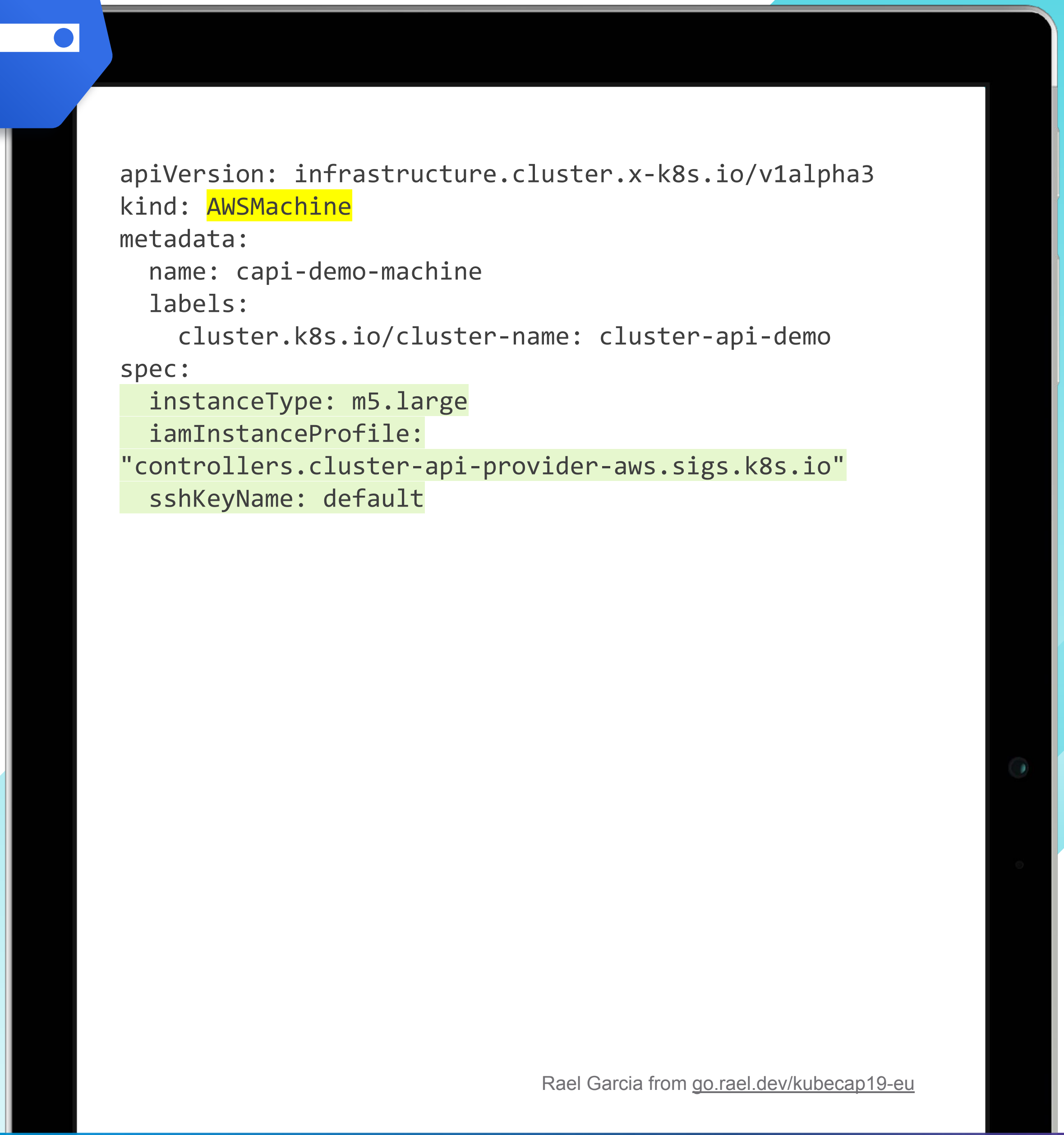
```yaml
apiVersion: cluster.x-k8s.io/v1alpha3
kind: Machine
metadata:
  name: capi-demo-machine
  labels:
    cluster.k8s.io/cluster-name: cluster-api-demo
spec:
 bootstrap:
    configRef:
      kind: KubeadmConfig
      apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
      namespace: default
      name: capi-demo-machine
  infrastructureRef:
    kind: AWSMachine
    apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
    namespace: default
    name: capi-demo-machine
  version: "v1.17.4"
```

Rael Garcia from go.rael.dev/kubecap19-eu

InfraMachine Example

InfraMachine (AWSMachine)

Infrastructure provider-specific
machine configuration



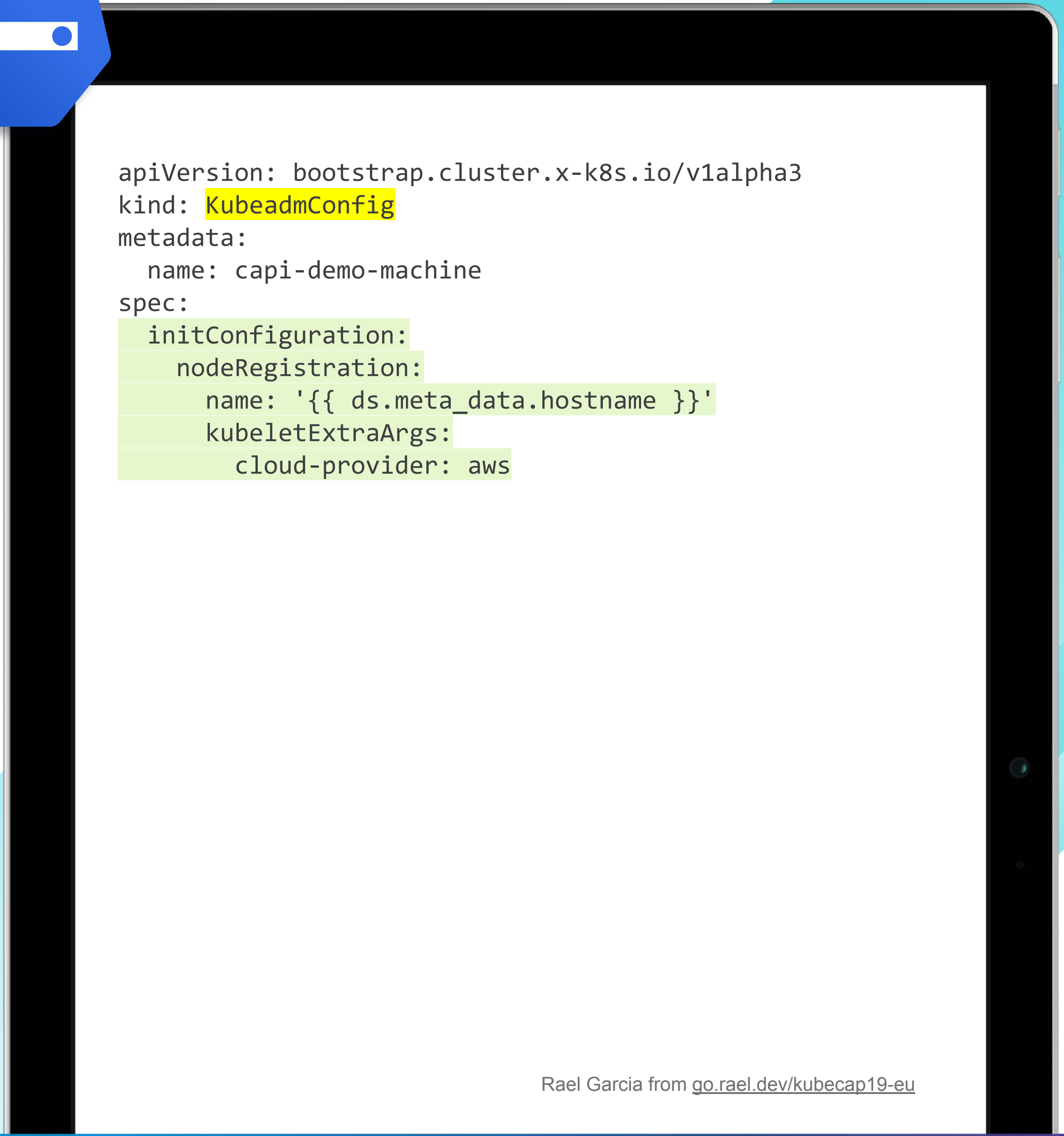
```
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSMachine
metadata:
  name: capi-demo-machine
  labels:
    cluster.k8s.io/cluster-name: cluster-api-demo
spec:
  instanceType: m5.large
  iamInstanceProfile:
"controllers.cluster-api-provider-aws.sigs.k8s.io"
  sshKeyName: default
```

Rael Garcia from go.rael.dev/kubecap19-eu

Bootstrap Configuration Example

BootstrapConfig (KubeadmConfig)

Bootstrap configuration for
a specific machine



```yaml
apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
kind: KubeadmConfig
metadata:
  name: capi-demo-machine
spec:
  initConfiguration:
    nodeRegistration:
      name: '{{ ds.meta_data.hostname }}'
      kubeletExtraArgs:
        cloud-provider: aws
```

Rael Garcia from go.rael.dev/kubecap19-eu

Cluster API:

CRDs

- Cluster
- ControlPlane
- Machine
- **MachineSet**
- MachineDeployment

Spec

Replica count

Minimum ready seconds

Deletion policy

Label selector

Machine template

Status

Replicas (observed, fully labeled, ready, available)

Observed generation

Error details

Cluster API:

CRDs

- Cluster
- ControlPlane
- Machine
- MachineSet
- **MachineDeployment**

Spec

Replica count

Label selector

Machine template

Replacement strategy

Minimum ready seconds

Revision history limit

Paused

Progress deadline seconds

Infrastructure provider-specific InfraMachineTemplate

Bootstrap provider-specific BootstrapConfigTemplate

Status

Observed generation

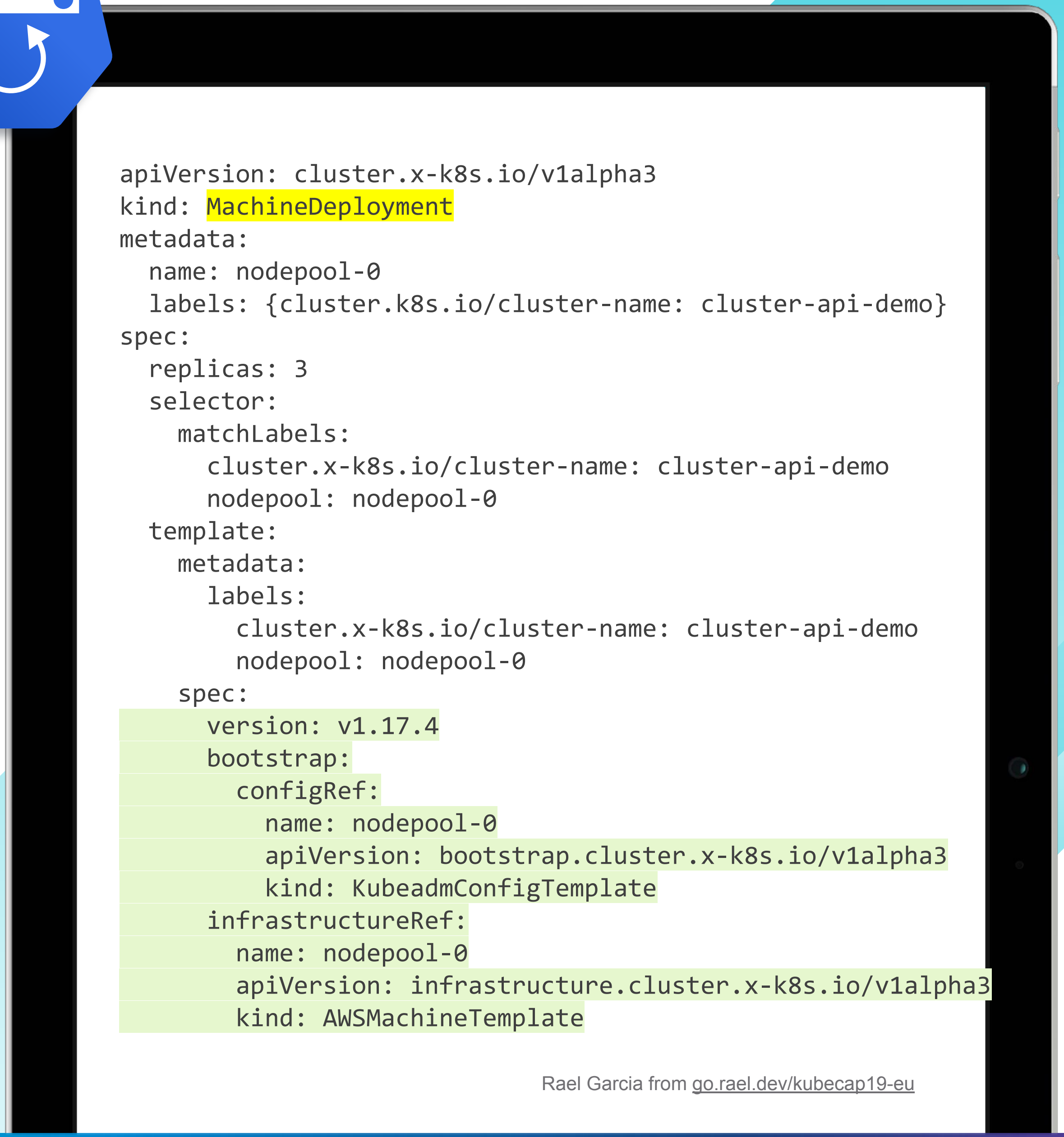
Replicas (total, updated, ready, available, unavailable)

MachineDeployment Example

MachineDeployment

Declarative updates for Machines via MachineSets

Update strategy allows control of the rate at which a change is applied



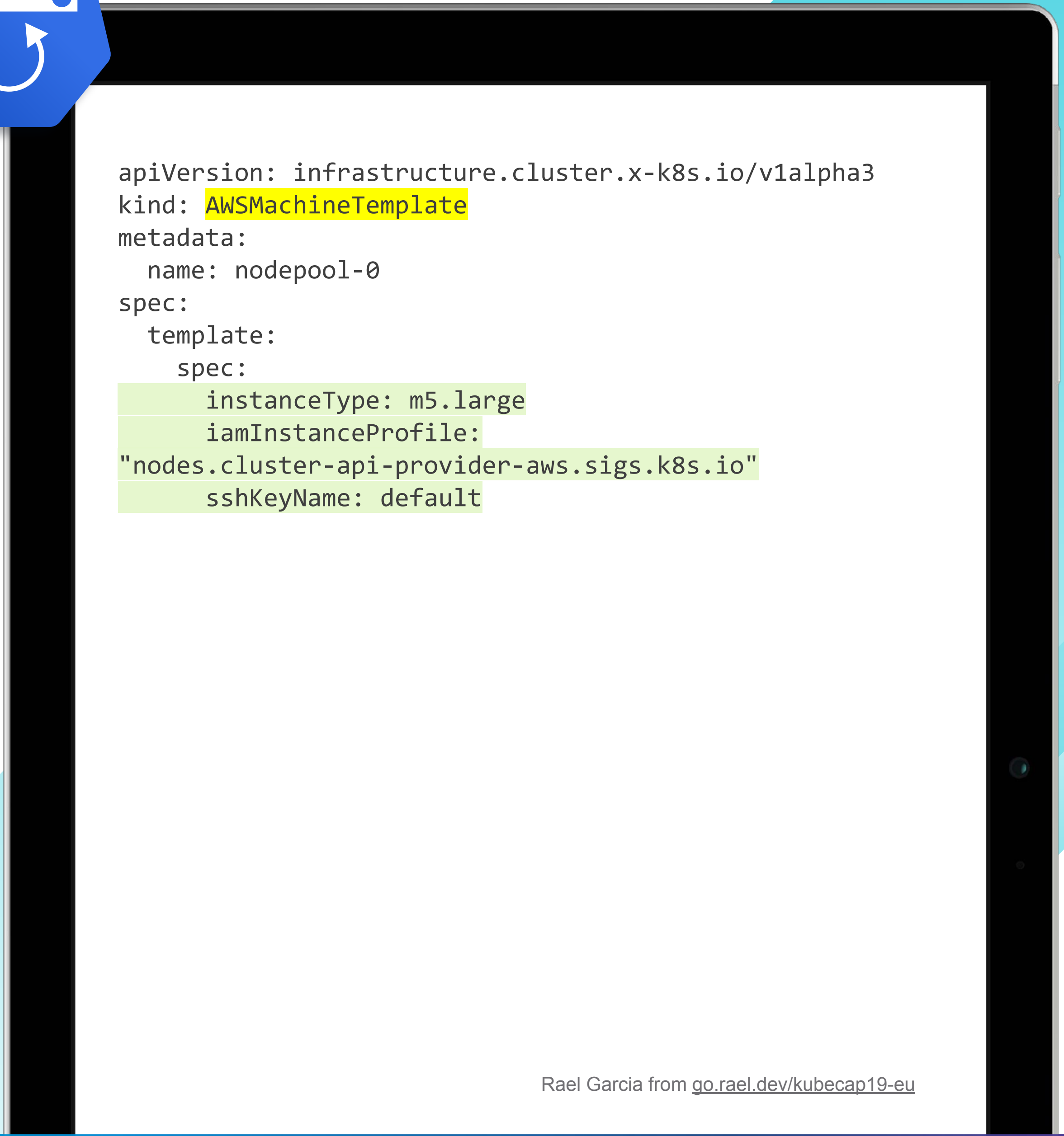
```yaml
apiVersion: cluster.x-k8s.io/v1alpha3
kind: MachineDeployment
metadata:
  name: nodepool-0
  labels: {cluster.k8s.io/cluster-name: cluster-api-demo}
spec:
  replicas: 3
  selector:
    matchLabels:
      cluster.x-k8s.io/cluster-name: cluster-api-demo
      nodepool: nodepool-0
  template:
    metadata:
      labels:
        cluster.x-k8s.io/cluster-name: cluster-api-demo
        nodepool: nodepool-0
    spec:
      version: v1.17.4
      bootstrap:
        configRef:
          name: nodepool-0
          apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
          kind: KubeadmConfigTemplate
      infrastructureRef:
        name: nodepool-0
        apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
        kind: AWSMachineTemplate
```

Rael Garcia from go.rael.dev/kubecap19-eu

InfraMachineTemplate Example

InfraMachineTemplate (AWSMachineTemplate)

Template used for creating
Provider-specific InfraMachine instances
for the MachineDeployment



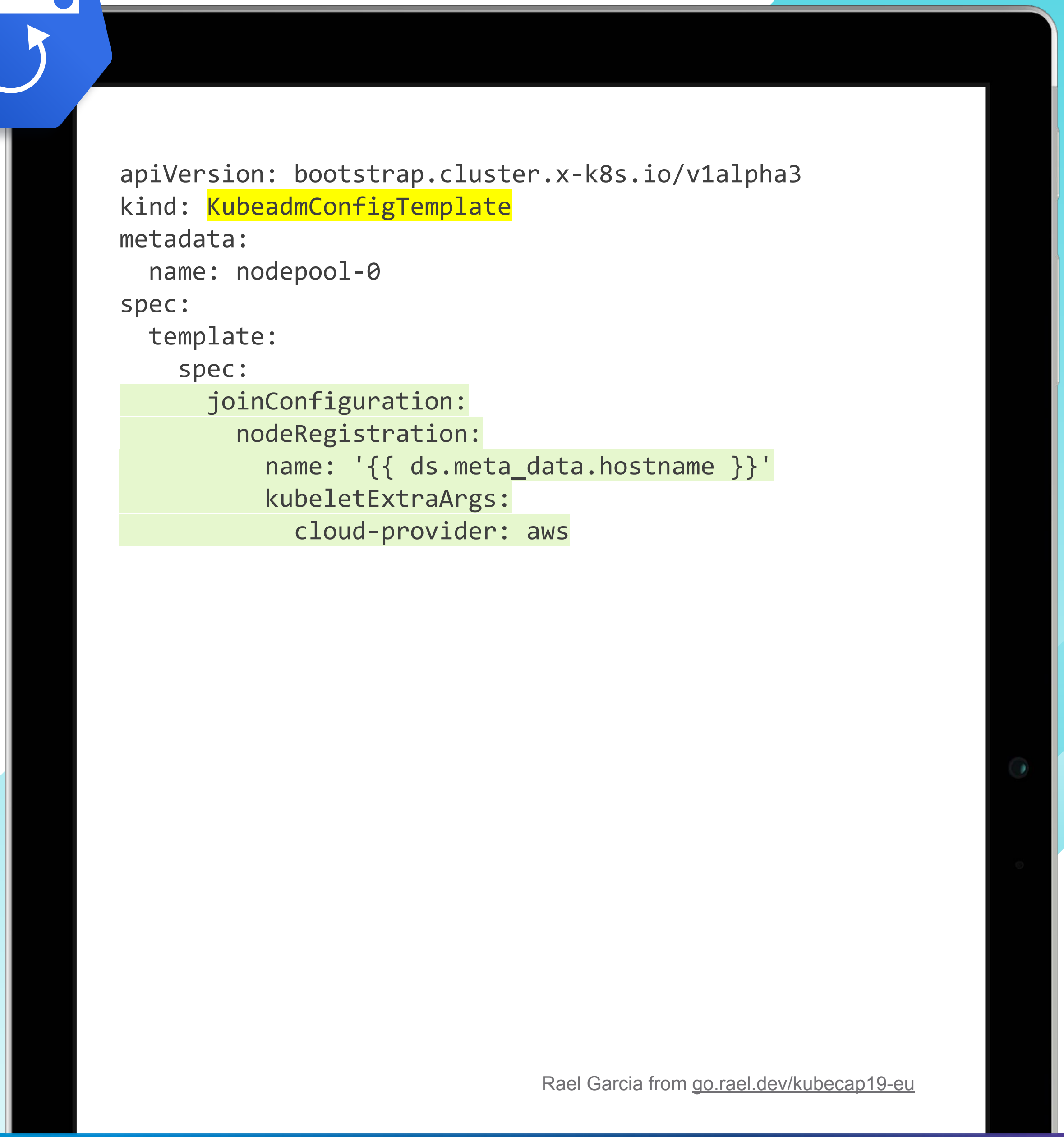
```
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSMachineTemplate
metadata:
  name: nodepool-0
spec:
  template:
    spec:
      instanceType: m5.large
      iamInstanceProfile:
"nodes.cluster-api-provider-aws.sigs.k8s.io"
      sshKeyName: default
```

Rael Garcia from go.rael.dev/kubecap19-eu

BootstrapConfigTemplate Example

BootstrapConfigTemplate (KubeadmConfigTemplate)

Template used for creating BootstrapConfig objects for configuring the InfraMachine instances for the MachineDeployment



```yaml
apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
kind: KubeadmConfigTemplate
metadata:
  name: nodepool-0
spec:
  template:
    spec:
      joinConfiguration:
        nodeRegistration:
          name: '{{ ds.meta_data.hostname }}'
          kubeletExtraArgs:
            cloud-provider: aws
```

Rael Garcia from go.rael.dev/kubecap19-eu

Cluster API Deployment Topology

5 separate deployments

Deployment 1

Common Cluster API controllers

Cluster
Machine

MachineSet

MachineDeployment

Deployment 2

Cluster API webhooks

Deployment 3

Control Plane Provider -specific controllers KubeadmControlPlane

KubeadmControlPlane

Deployment 4

Infrastructure Provider -specific controllers

InfraCluster

InfraMachine

Deployment 5

Bootstrap Provider -specific controllers

BootstrapConfig