
KubeCon
CloudNativeCon
Europe 2023
TiKV

Cluster API Providers

Intro, Deep Dive and Community

Who are we?

Ankita Swamy

Senior MTS @ VMware

Maintainer of Cluster API Provider AWS

Ashutosh Kumar

Engineer @ VMware

Maintainer of Cluster API Provider Azure

Richard Case

Principal Engineer @ SUSE

Maintainer of Cluster API Providers AWS, GCP, Microvm, RKE2

What we will be covering

- Cluster API Intro - very quick
- Cluster API Providers
- Cluster API Provider Updates
- Wrap up

What is Cluster API?

Built on the premise that “**Cluster lifecycle management is difficult**”.....especially cross cloud with a **consistent operational model**

What is Cluster API?

The Cluster API project uses **Kubernetes-style APIs** and patterns to **automate cluster lifecycle management** for platform operators

What is Cluster API?

The **supporting infrastructure**, like virtual machines, networks, load balancers, and VPCs, as well as the **Kubernetes cluster configuration** are all **defined in the same way that application developers operate deploying and managing their workloads**.

What is Cluster API?

KubeCon CloudNativeCon Europe 2023

Extensibility is core to Cluster API……..

……designed around interchangeable components via “providers”

What is Cluster API?

Establishes building blocks for higher order functions and day 2 operations:

- Cluster templating & “classes” of clusters
- Hooking into the wider provisioning landscape

What is Cluster API?

- Community calls every week on Wednesday @ 10am PT

- For a walkthrough of CAPI see the “let’s talk about…” series by Stefan & Fabrizio:
  - https://github.com/kubernetes-sigs/cluster-api/discussions/6106

What is a Cluster API provider?

A Kubernetes **operator** that implements infrastructure / operating environment specific functionality that is utilized by core Cluster API when managing the lifecycle of a K8s cluster.

The operator implements a contract via its custom resources (i.e. CRDs) depending on the type of provider, which enables interaction between core CAPI and the provider.

Provider Types

- **Infrastructure -** used to provision any infrastructure that is required to create and run a Kubernetes cluster. For example, networking, security groups, virtual or physical host machines

- **Bootstrap -** used to create the “user-data” that is passed to the infrastructure machines that contains the instructions to bootstrap a Kubernetes node on that machine. 2 parts to it:
  - Action: how Kubernetes is bootstrapped (e.g. invoking kubeadm)
  - Format: how the action is encoded and passed to the machine (e.g. cloud-init, ignition)

- **Control plane -** used to control the creation & lifecycle of the Kubernetes control plane. It can utilize resources created by bootstrap and infrastructure providers.
  - Kubeadm control plane is the original
  - Managed Kubernetes (i.e. EKS, AKS) implementations - no nodes

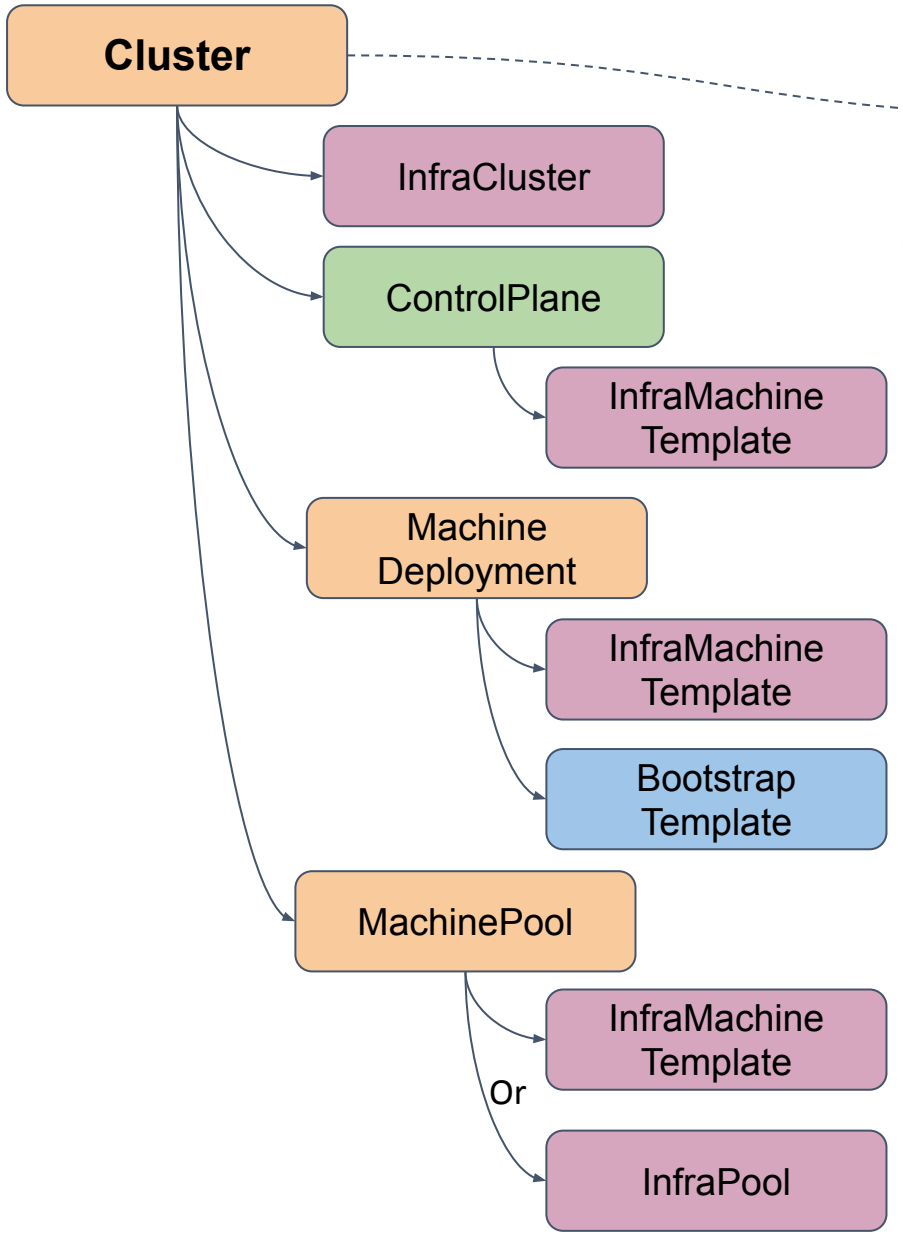
New Provider Types

- **Addon -** used to manage the lifecycle of workloads on the cluster after initial provisioning. This will ultimately succeed “ClusterResourceSet”. Currently, only 1 provider for Helm: https://github.com/kubernetes-sigs/cluster-api-addon-provider-helm

- **IPAM -** used to claim IP address from an IPAM solution. Reference implementation: https://github.com/telekom/cluster-api-ipam-provider-in-cluster

```yaml
apiVersion: ipam.cluster.x-k8s.io/v1alpha1
kind: InClusterIPPool
metadata:
  name: inclusterippool-sample
spec:
  first: 10.0.0.10
  last: 10.10.0.42
  prefix: 24
  gateway: 10.0.0.1
```

CRDs & Spec



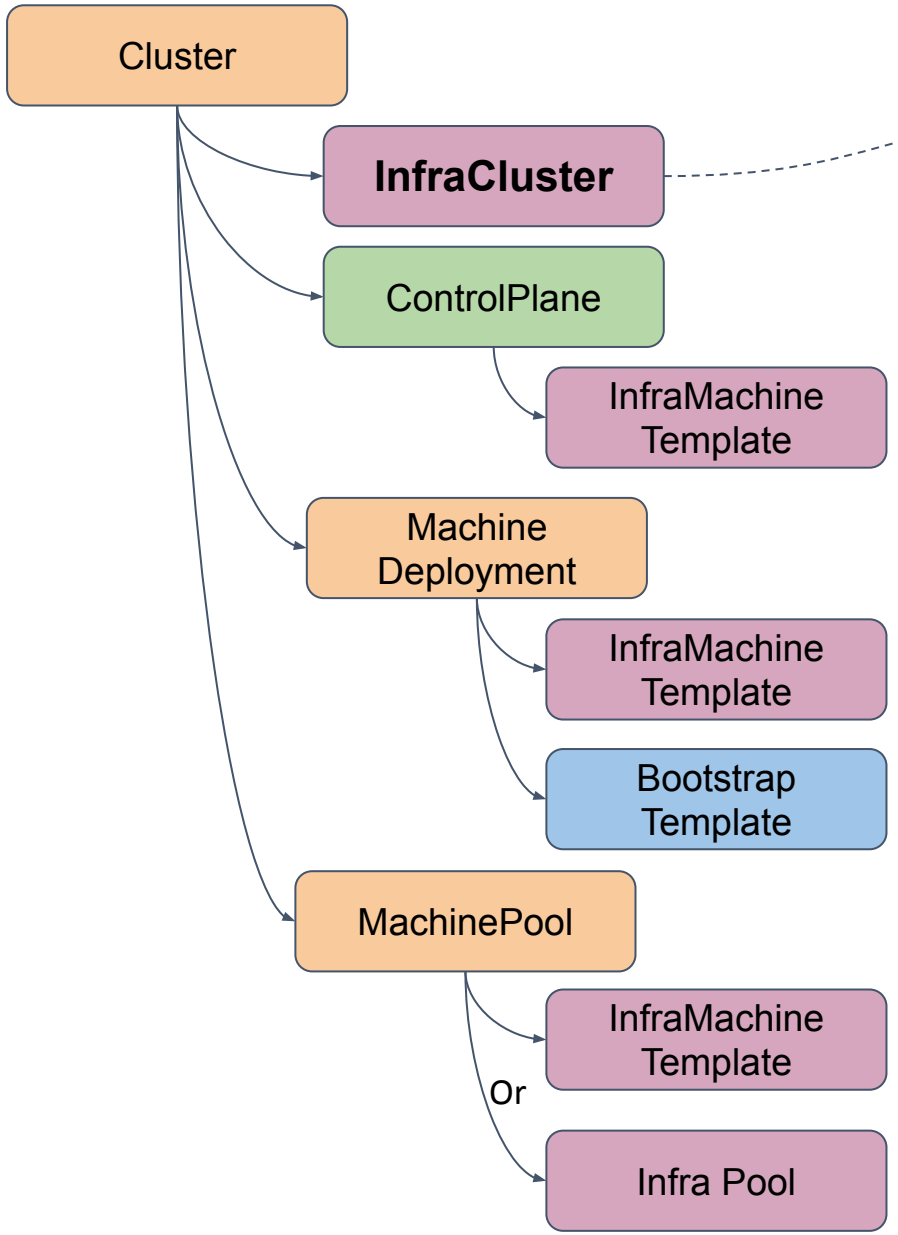
Cluster

Logically represents the cluster as a whole.

Contains general configuration like pod cidr.

```yaml
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: "example"
spec:
  clusterNetwork:
    pods:
      cidrBlocks: ["192.168.0.0/16"]
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
    kind: GCPCluster
    name: "example"
  controlPlaneRef:
    kind: KubeadmControlPlane
    apiVersion: controlplane.cluster.x-k8s.io/v1beta1
    name: "example-control-plane"
```

CRDs & Spec



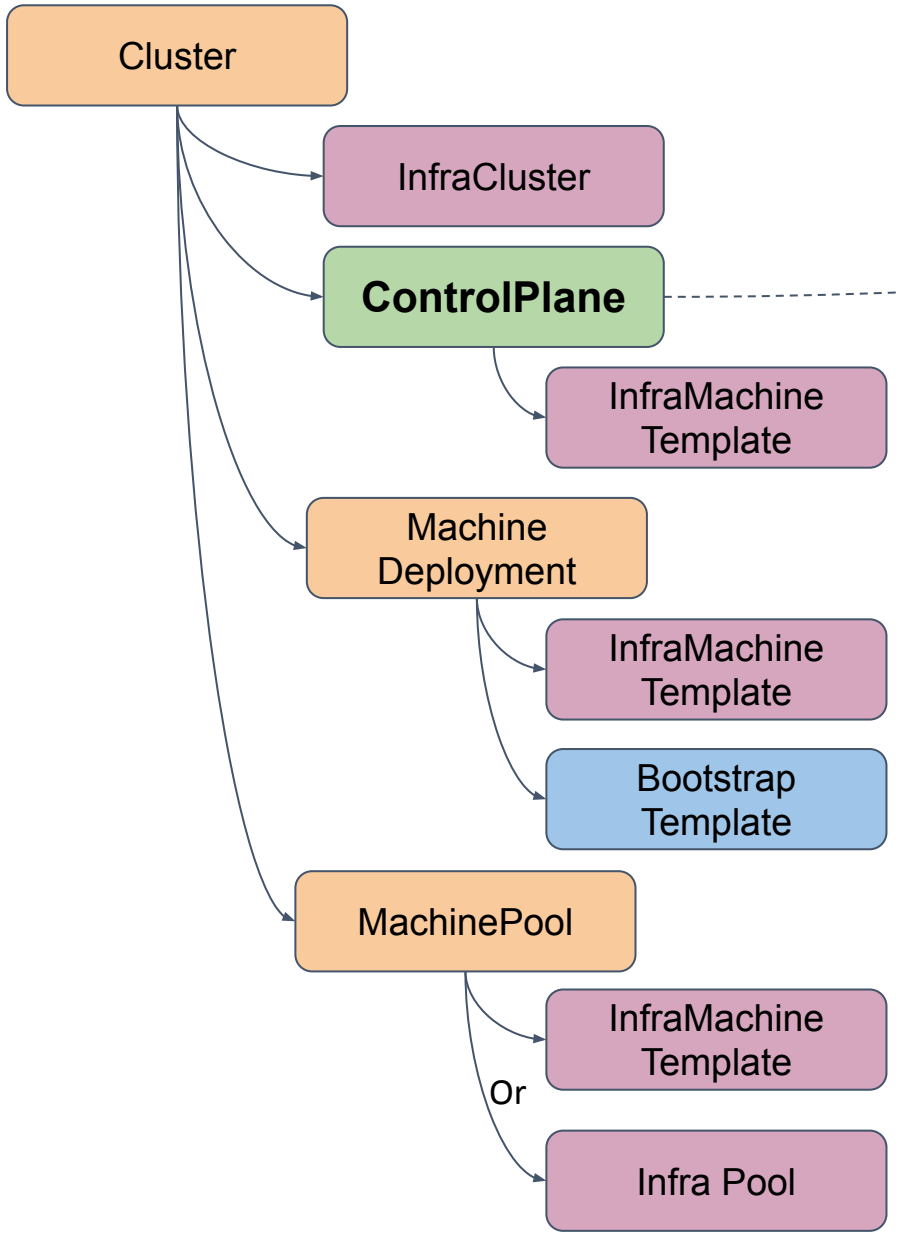
InfraCluster

Represents base infrastructure that is required for the cluster.

Examples: AzureCluster, AWSCluster,

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPCluster
metadata:
  name: "example"
spec:
  project: "myproject"
  region: "europe-west2""
  network:
    name: "default"
```

CRDs & Spec



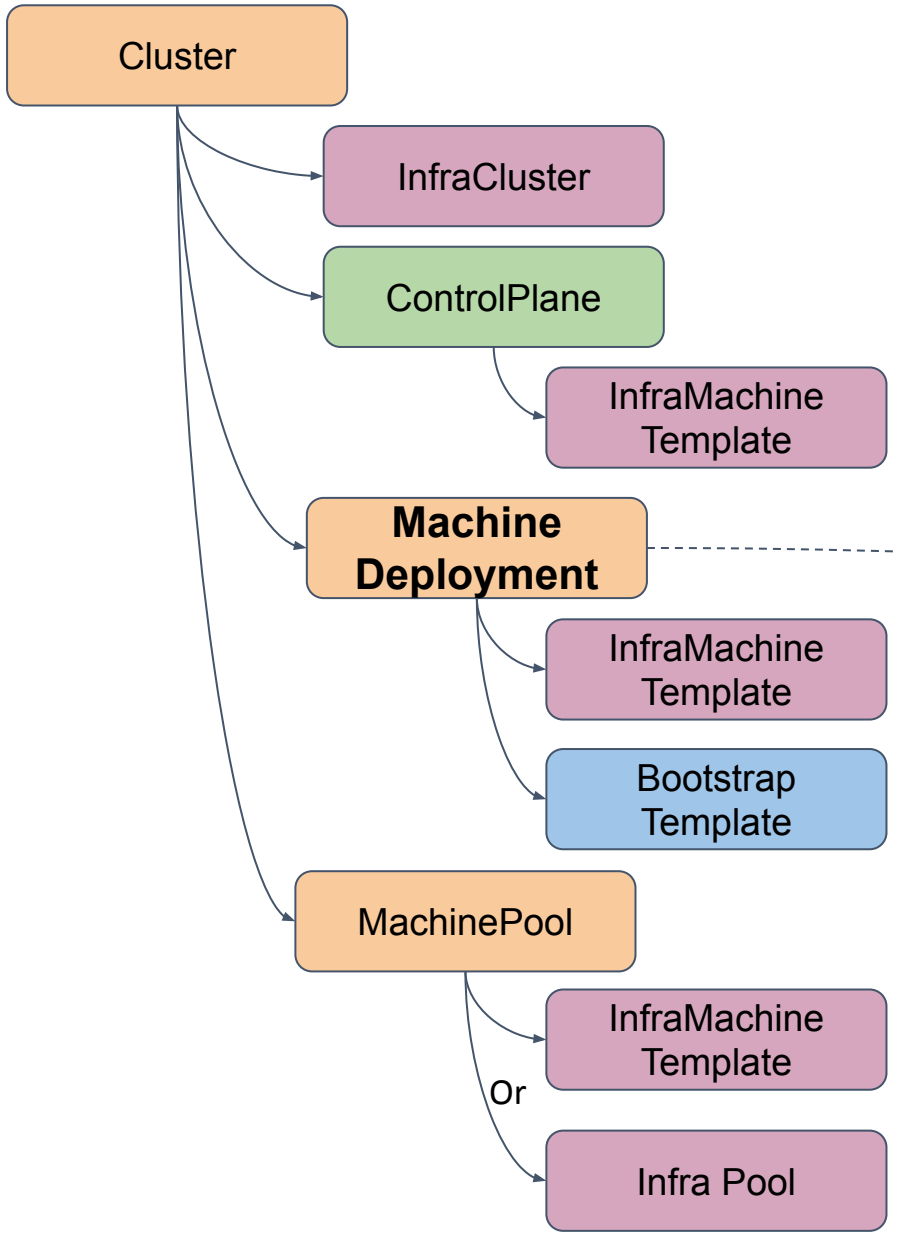
ControlPlane

Represents the Kubernetes control plane.
Specify replicas, kubernetes version

Examples: KubeadmControlPlane, EKSManagedControlPlane

```yaml
apiVersion: controlplane.cluster.x-k8s.io/v1beta1
kind: KubeadmControlPlane
metadata:
  name: "example-control-plane"
spec:
  replicas: 3
  version: "v1.25.0"
  machineTemplate:
    infrastructureRef:
      kind: GCPMachineTemplate
      apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
      name: "example-control-plane"
  kubeadmConfigSpec:
    initConfiguration:
      ...
    clusterConfiguration:
      ...
    joinConfiguration:
      ...
```

CRDs & Spec

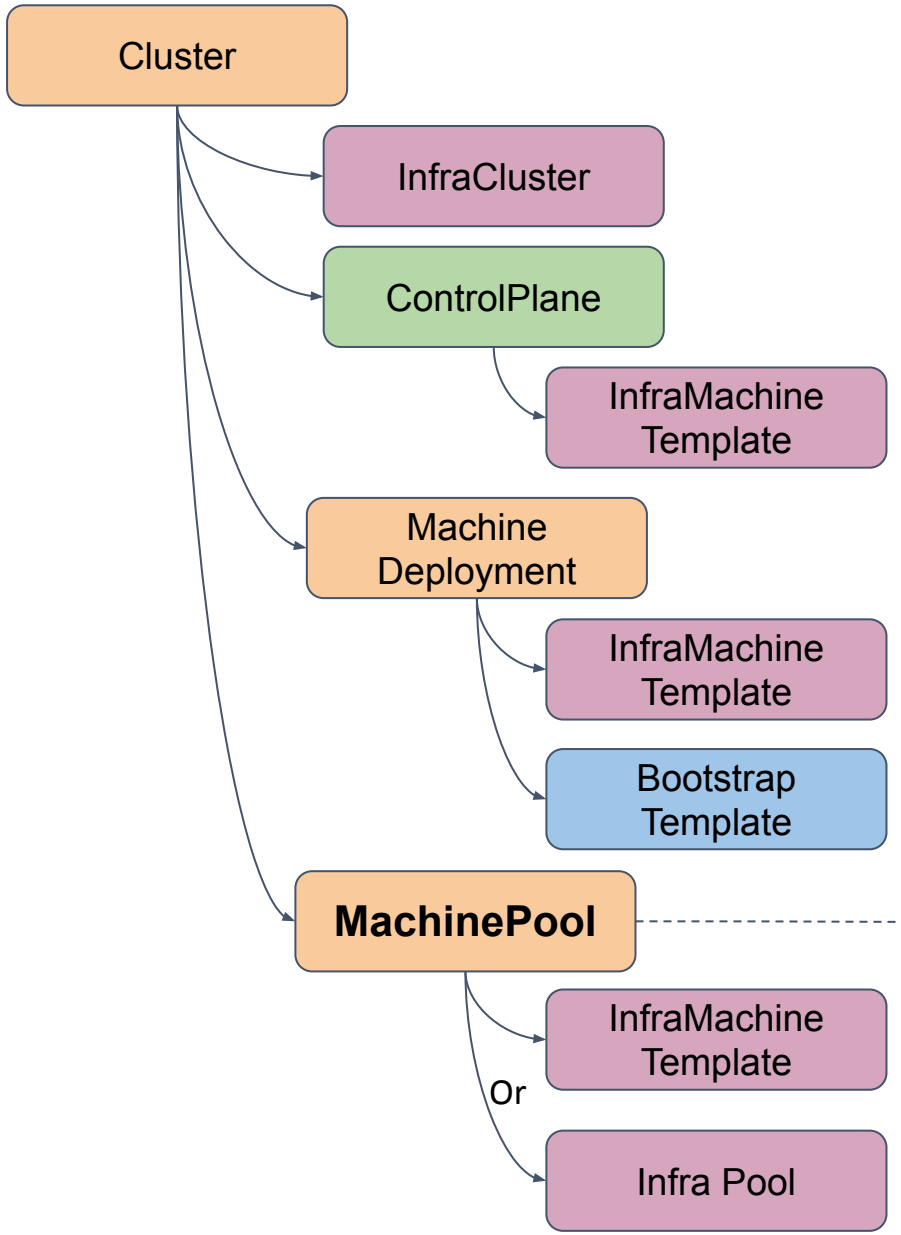


MachineDeployment

Represents the worker nodes of the cluster.
Manages lifecycle of nodes, including k8s version upgrades.

```yaml
apiVersion: cluster.x-k8s.io/v1beta1
kind: MachineDeployment
metadata:
  name: "example-md-0"
spec:
  clusterName: "example"
  replicas: 1
  template:
    spec:
      clusterName: "example"
      failureDomain: "europe-west2-a"
      version: "1.22.9"
      bootstrap:
        configRef:
          name: "example-md-0"
          apiVersion: bootstrap.cluster.x-k8s.io/v1beta1
          kind: KubeadmConfigTemplate
      infrastructureRef:
        name: "example-md-0"
        apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
        kind: GCPMachineTemplate
```

CRDs & Spec

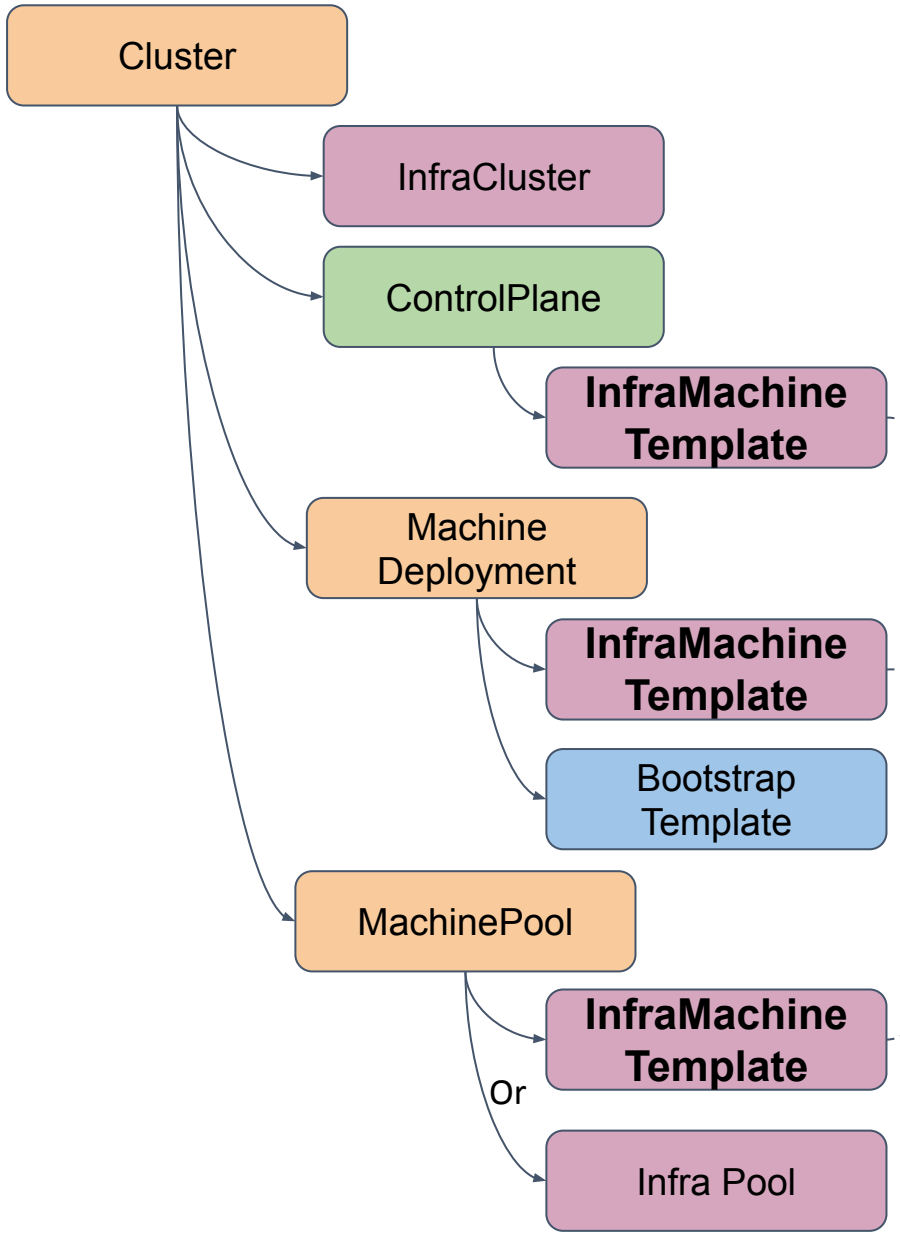


MachinePool

Represents a pool of worker nodes of the cluster. Implementations generally backed by an infra service that support scaling (i.e. ASG, VMSS)

```yaml
apiVersion: cluster.x-k8s.io/v1beta1
kind: MachinePool
metadata:
  name: example-mp-0
spec:
  clusterName: example
  replicas: 3
  template:
    spec:
      bootstrap:
        dataSecretName: ""
      clusterName: example
      infrastructureRef:
        apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
        kind: GCPManagedMachinePool
        name: example-mp-0
```

CRDs & Spec



InfraMachine Template

Represents a template for an individual machine.
Instance of Machine and InfraMachine created from it

Examples: AWSMachineTemplate, DockerMachineTemplate

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPMachineTemplate
metadata:
  name: "example-control-plane"
spec:
  template:
    spec:
      instanceType: "n1-standard-2"
      image: "cluster-api-ubuntu-2004-gpu"
```

Cluster API Provider Azure (CAPZ)

- It is again a Kubernetes operator (a.k.a controller manager)
  - The operator reconciles to provision infrastructure resources on Azure

- It also consists of webhooks that validates and/or put sane defaults on the CRs

Cluster API Provider Azure (CAPZ)

AzureClusterIdentity

AzureCluster

AzureMachineTemplate

AzureMachine

AzureMachinePool

Cluster API Provider Azure (CAPZ)

- AzureClusterIdentity has the details which can be used by CAPZ for authentication to Azure APIs.
  - AzureCluster references AzureClusterIdentity and enables multi tenancy.

- AzureMachineTemplate consists information related to VM e.g dataDisks, OSDisk, VMSize etc.
  - MachineDeployment and KubeadmControlPlane has reference to AzureMachineTemplate

Cluster API Provider Azure (CAPZ)

AzureCluster

Group

PublicIP

LoadBalancer

VirtualNetwork

NatGateway

PrivateDNS

SecurityGroup

Subnet

BastionDNS

RouteTable

VnetPeering

PrivateEndpoint

Cluster API Provider Azure (CAPZ)

AzureMachine

PublicIP

InboundNatRule

NetworkInterface

AvailabilitySet

Disk

VirtualMachine

RoleAssignment

VMExtension

Tag

Cluster API Provider Azure (CAPZ)

- CAPZ implementation of Azure Managed Kubernetes has moved to graduation from experimental in v1.8.0 release.
- CAPZ uses out of tree cloud provider by default from v1.8.0 release.
- CAPZ enables you to create Kubernetes clusters using Flatcar Container Linux on Microsoft Azure.
- Support for VMSS flexible orchestration mode.
- Going forward CAPZ is going to support workload identity and is expected to be delivered in upcoming release.

Cluster API Provider AWS (CAPA)

- Kubernetes-native declarative infrastructure for AWS
- Manages bootstrapping of VPCs, gateways, security groups and EC2 instances
- Choice of OSes among Amazon Linux 2, CentOS 7, Ubuntu(18.04, 20.04) and Flatcar for instances

Cluster API Provider AWS (CAPA)

AWSCluster

VPC

Public IP

Elastic Load Balancer

Internet Gateways

NAT Gateways

Egress Only Internet Gateways

Security Groups

Subnets

Bastion

Route Tables

Secondary CIDR

S3

Cluster API Provider AWS (CAPA)

AWSMachine

PublicIP

Spot instances

Availability Zones

Security Groups

Volumes

Load balancer

Network Interfaces

EC2 instance

Tag

Cluster API Provider AWS (CAPA)

- EKS support in CAPA graduated from experimental to stable feature in v0.7.0
- Ignition support enabled for Flatcar OS for bootstrapping machines to the cluster in v1.4.0
- External Resource Garbage collection feature introduced in v1.5.0
- IPv6 support for EKS introduced in v2.0.0
- Support for NLBs as control plane load balancers introduced in v2.0.0
- Re-introduced AWSManagedCluster in v2.0.0

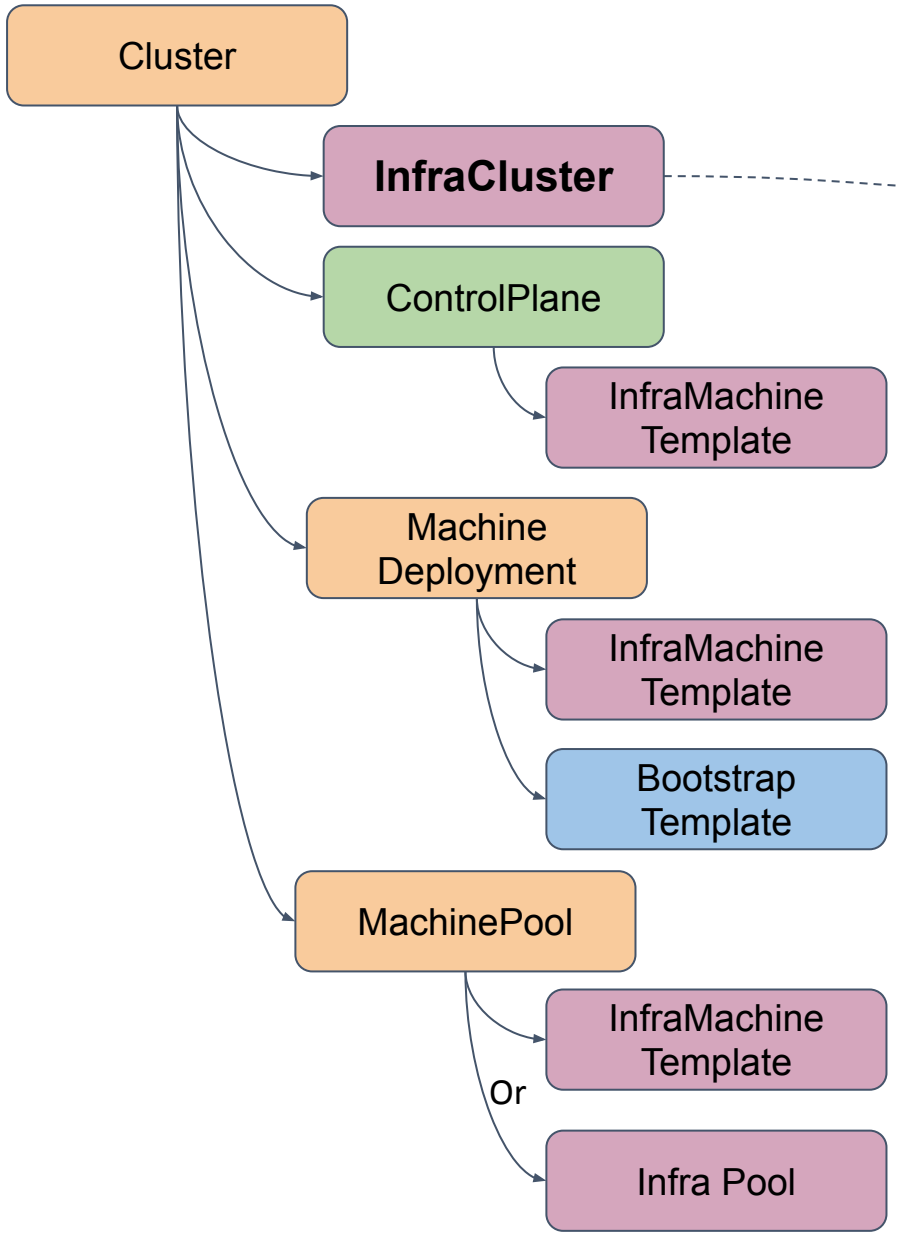
External Resource GC in CAPA

- External resources were sole responsibility of users
- External Resource GC ensures proper cleanup of AWS-specific resources, such as ELBs, NLBs, and security groups created by CCM
- Experimental feature
- Leverages clusterawsadm CLI tool to enable/disable feature
- Supported in both CAPA-managed and EKS-managed clusters

Cluster API Provider GCP (CAPG)

- Experimental support for GKE
  - Including MachinePool
  - Enabled via the **GKE** feature flag

Cluster API Provider GCP - GKE

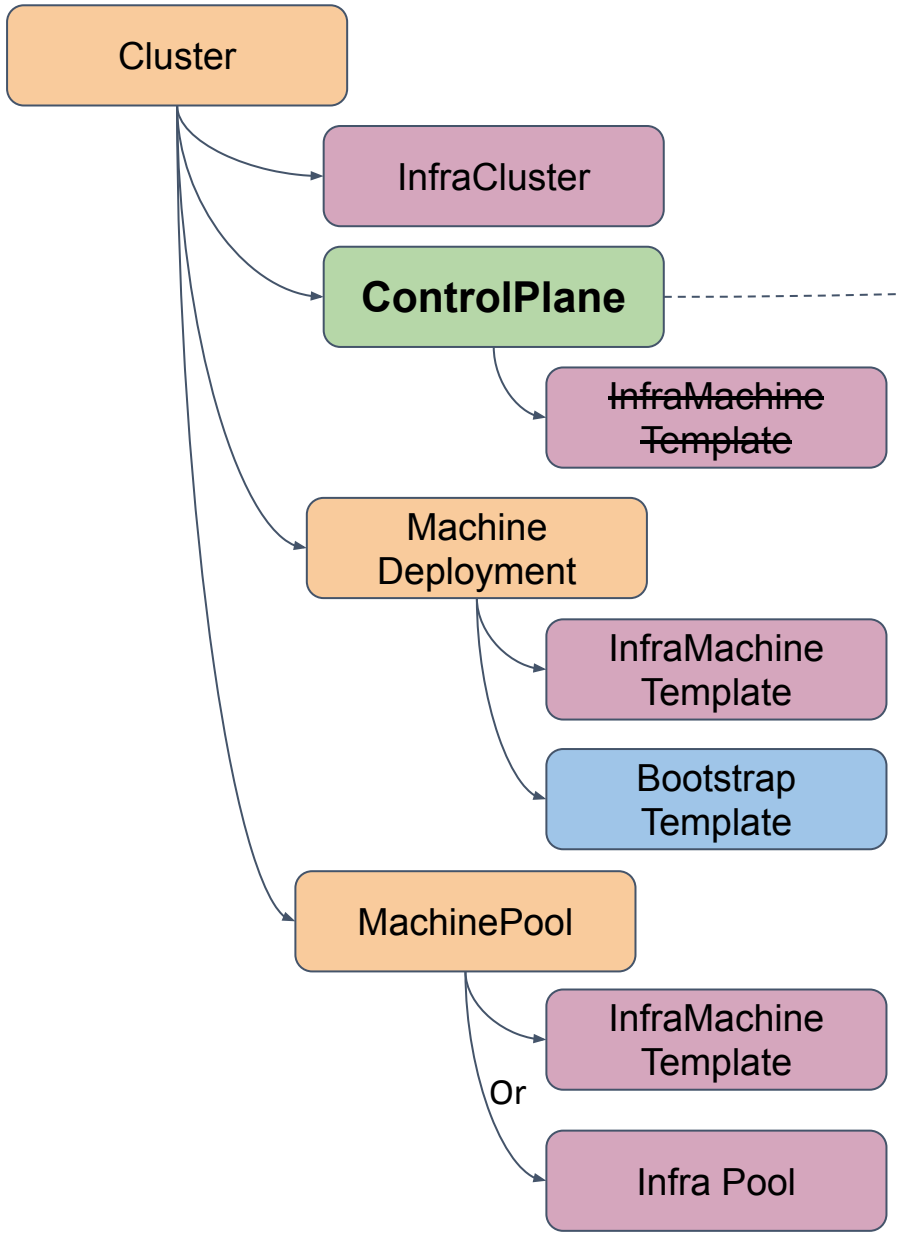


GCPManagedCluster

Creates base GCP infra required for the cluster.

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPManagedCluster
metadata:
  name: "example"
spec:
  project: "myproject"
  region: "europe-west2"
  network:
    name: "default"
```

Cluster API Provider GCP - GKE

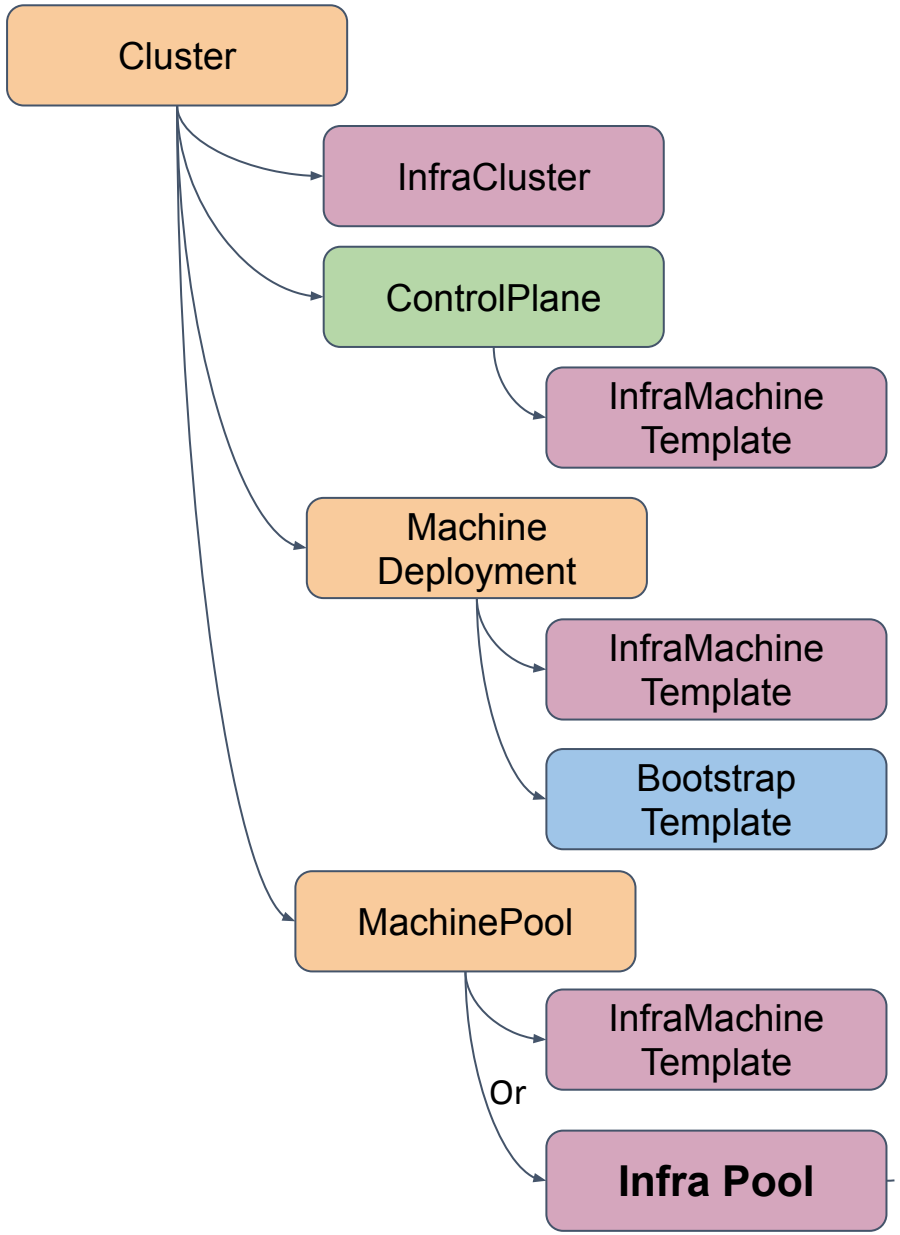


GCPManagedControlPlane

Creates & manages instance of the GKE service.
Autopilot supported

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPManagedControlPlane
metadata:
  name: "example-control-plane"
spec:
  project: "myproject"
  location: "europe-west2"
  releaseChannel: "regular"
```

Cluster API Provider GCP - GKE



GCPManagedMachinePool

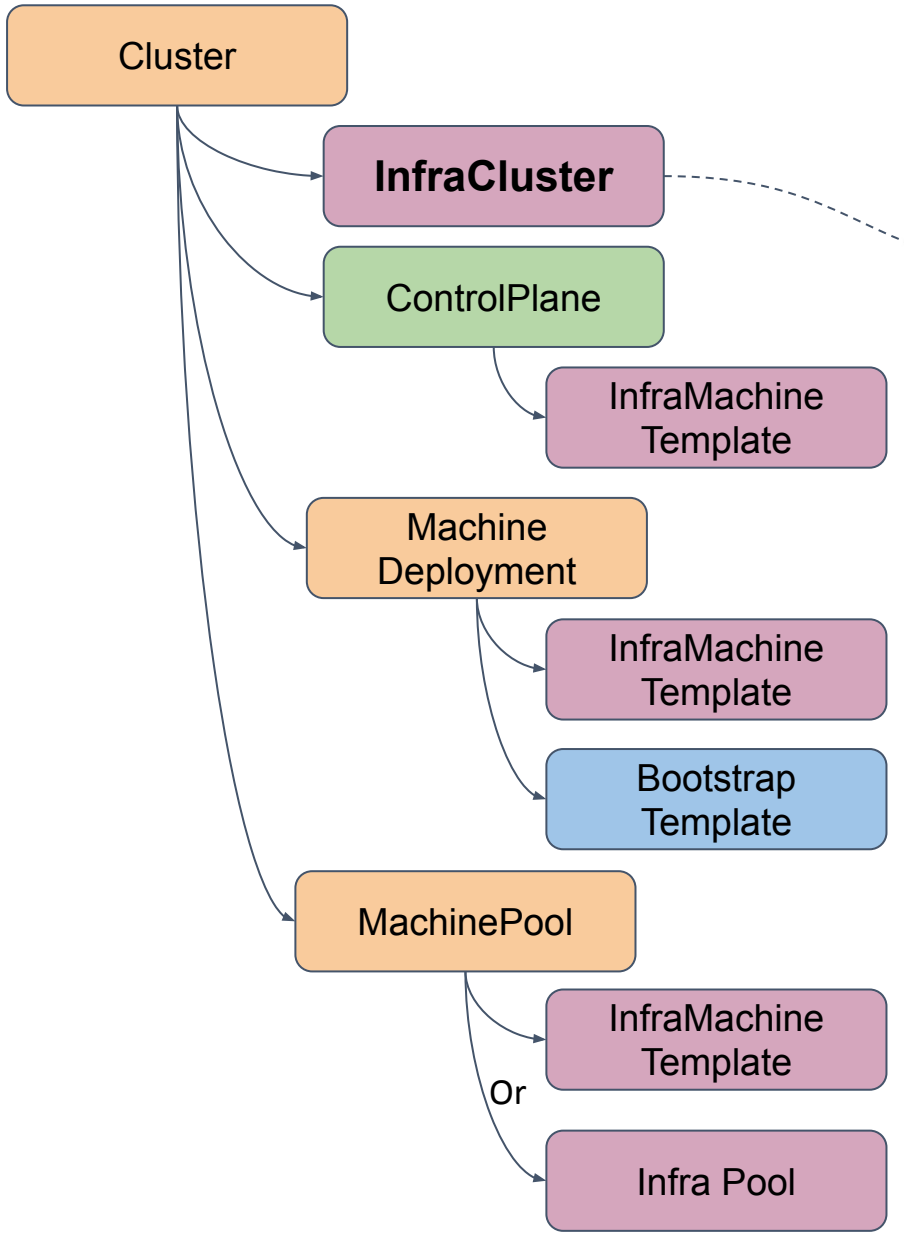
Creates and manages GKE node pool.
Scaling, labels, taints currently supported

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPManagedMachinePool
metadata:
  name: example-mp-0
spec:
  scaling:
    minCount: 3
    maxCount: 10
```

Cluster API Provider GCP (CAPG)

- Per cluster credentials
- LFX Mentee working on adding OpenTelemetary

Cluster API Provider GCP - Cluster Creds



GCPCluster/GCPManagedCluster

Optionally reference a secret containing the credentials

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: GCPCluster
metadata:
  name: "example"
spec:
  project: "myproj"
  region: "europe-west2"
  network:
    name: "default"
  credentialsRef:
    name: gcp-test-creds
    namespace: default
```

Other Updates

- CAAPH - addon provider for helm
  - Needs helps to carry on the great work already done
  - https://github.com/kubernetes-sigs/cluster-api-addon-provider-helm
- CAPRKE2 - new provider created for provisioning RKE2 based cluster
  - https://github.com/rancher-sandbox/cluster-api-provider-rke2

Wrap up

Wrap up

📝 Writing skills?

- Document our book: quick start, architecture diagrams, contracts, and so on!

🧬 Product skills?

- Gather use cases, compile user pulse surveys, draw roadmaps.
- Work with project's maintainers and the community to shape our product.
- Help with backlog grooming, maintain milestones.

👩‍💻 Coding skills?

- Review pull requests, become an approver.
- Search for help wanted, or good first issues across our repositories.

🙌 Other skills?

- Use CAPI and its providers then give feedback
- End user experience is the most valuable experience we can get

Questions?


KubeCon
CloudNativeCon
Europe 2023
Thank you!