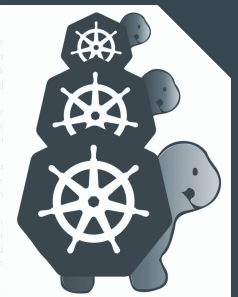
k8gb meets Cluster API

...

Jirka Kremser



Jirka Kremser

jkremser

@JirkaKremser

jiri.kremser@gmail.com

web: kremser.dev

previously:

- Red Hat
- Oracle
- GiantSwarm.io

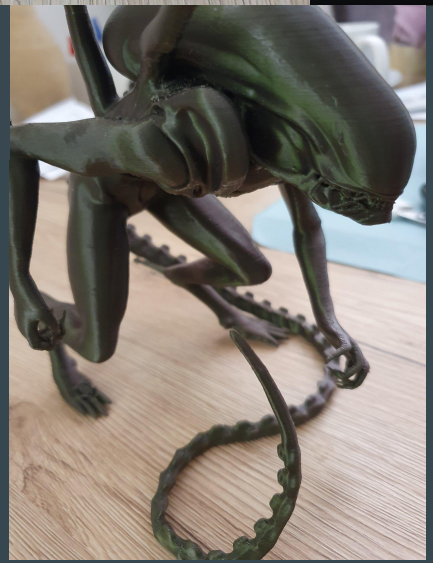
kedify.io (now)

pronounced as */yeerka/*

This slide deck: bit.ly/k8gb-capi

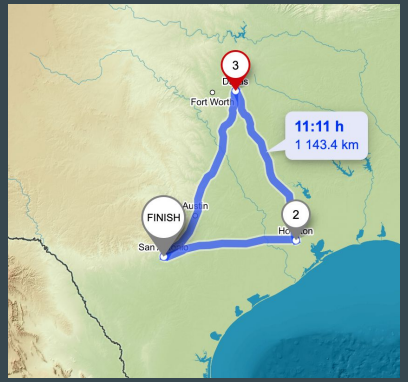
jkremser & life, slide #1

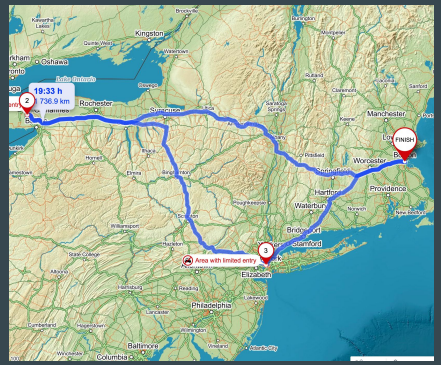
3D Printing

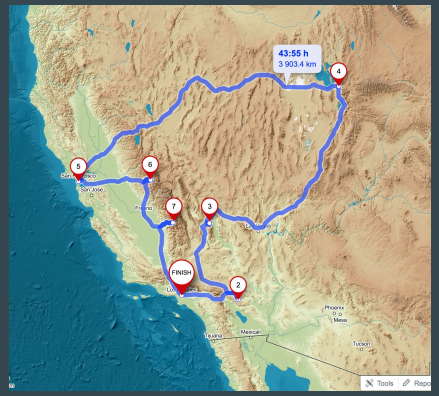


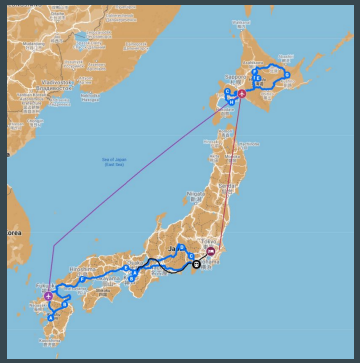
jkremser & life, slide #2

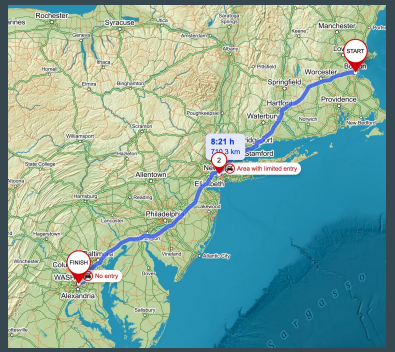
road trippin'

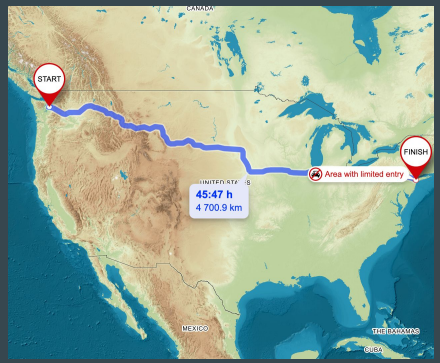


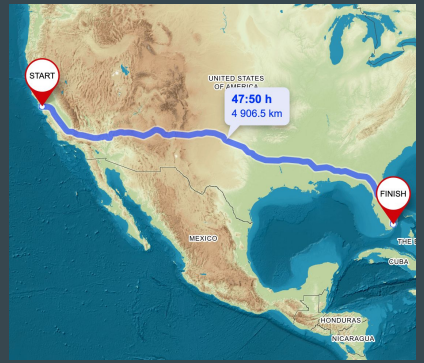


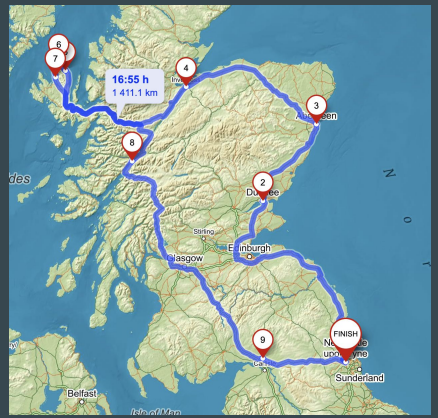


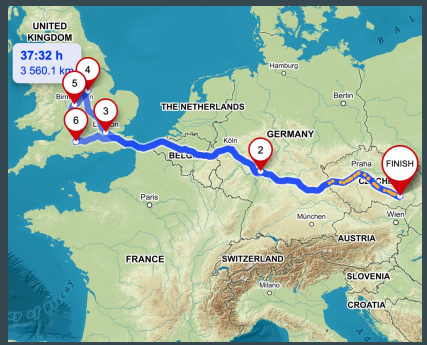












jkremser & life, slide #3

- 2 kids (boys, 8 and 10)
- drone “pilot” - youtu.be/1c0VaR70rDk

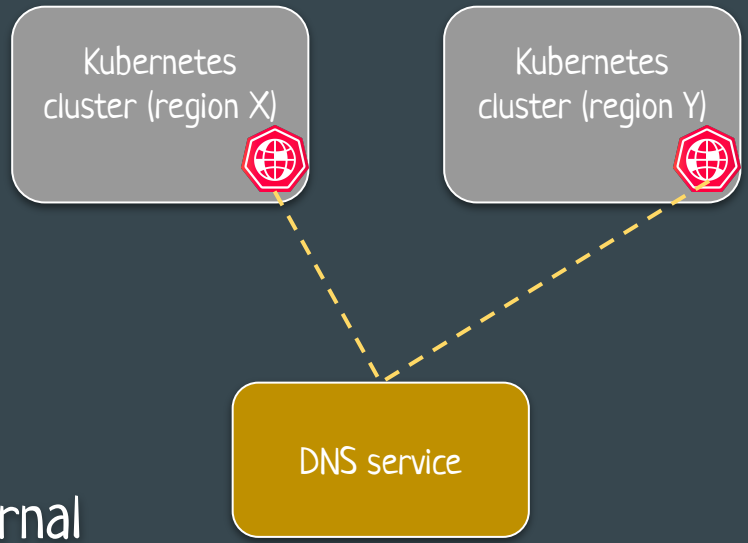
Motivation

"High availability of services as code"

- no vendor lock-in
- no external health checks
- no webui clicking / cloud provider cli tweaks

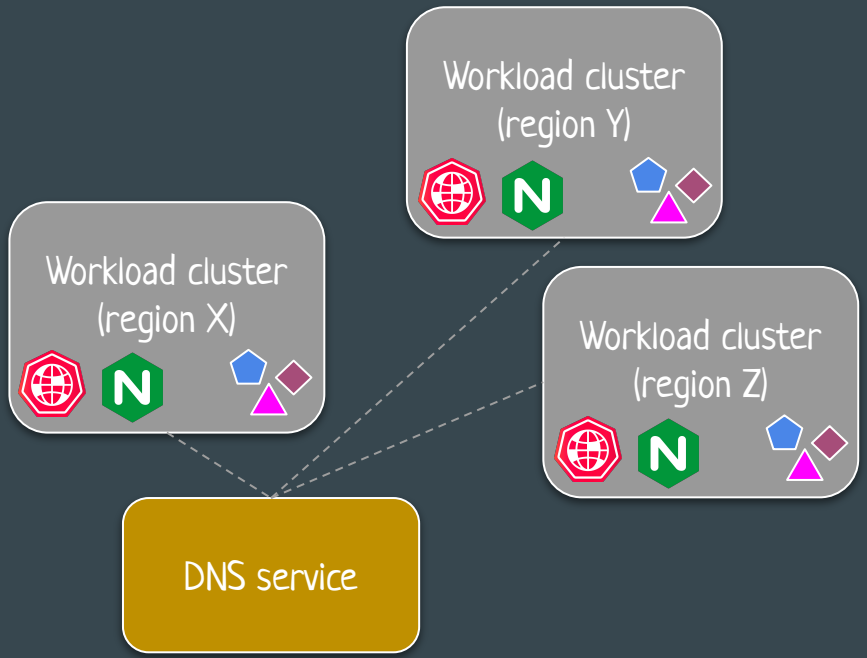
Part 1 - k8gb

- dns based global load balancer
- relies on k8s readiness probes
- decentralized - no SPoF
- CRD or annotation based
- ships internally own coredns and external dns components



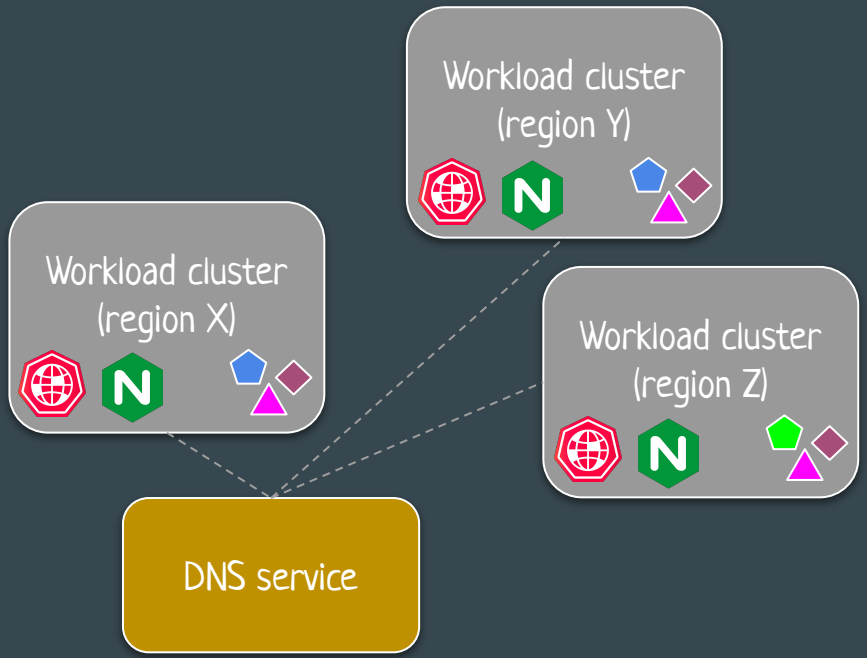
/k8gb-io/k8gb-manim/blob/master/example/k8gb.gif

Topology



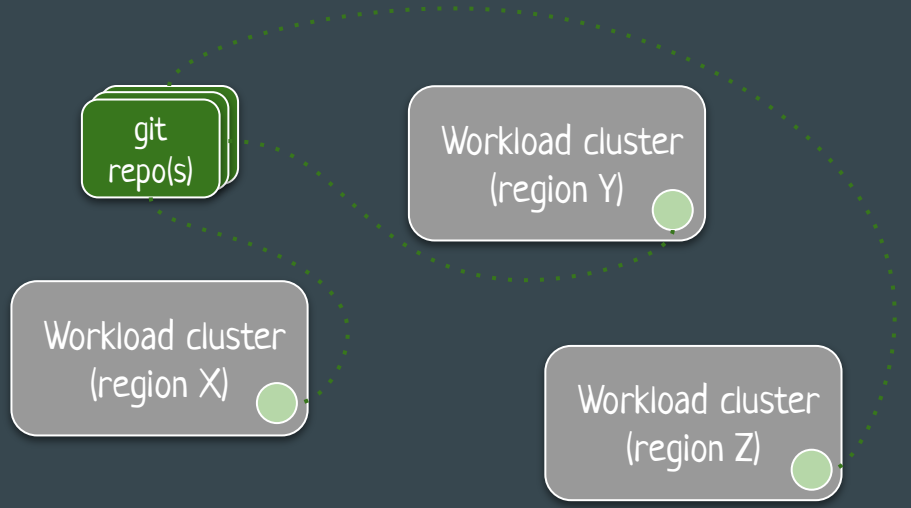
Clusters share common configuration and same workload applications. K8gb is not opinionated on how the config is delivered to the cluster.

Topology



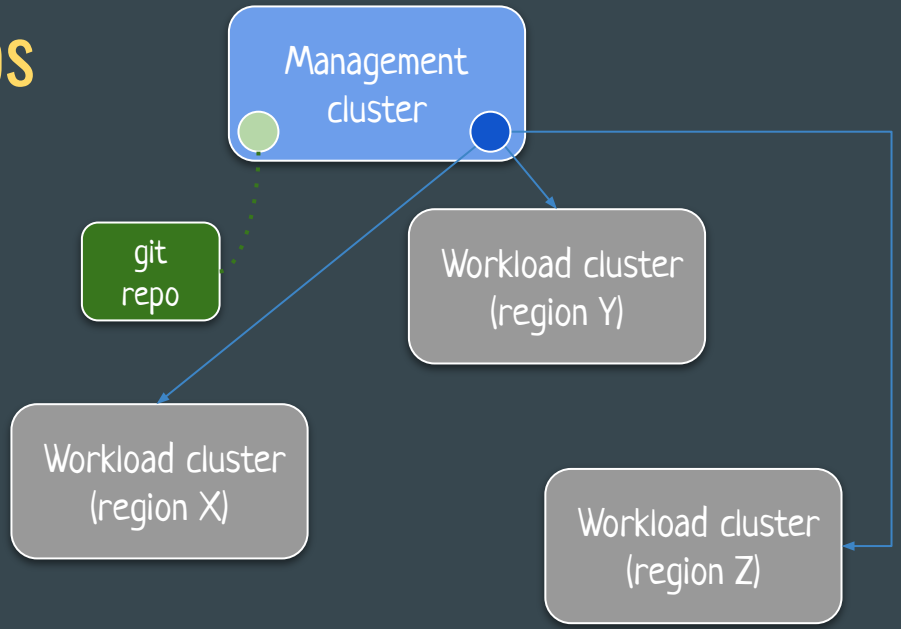
Clusters share common configuration and same workload applications. K8gb is not opinionated on how the config is delivered to the cluster.

GitOps



How to address configuration drift?

GitOps



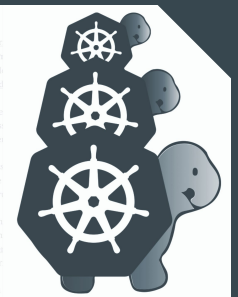
What if we want also k8s clusters as a code?

- flux/argocd
- something that can create and manage k8s clusters

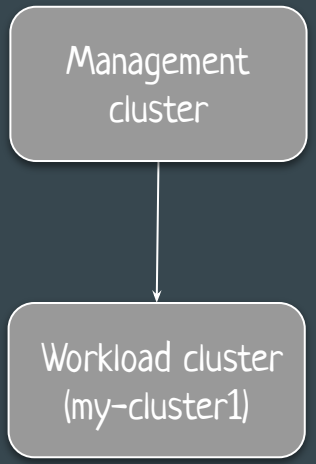
Part 2 - Cluster API “●“

```
clusterctl generate cluster my-cluster \
   --kubernetes-version v1.24.11 \
   --control-plane-machine-count=1 \
   --worker-machine-count=3 | kubectl apply -f -
```

```
~/w/example-capi-clusters master ?16 λ clusterctl describe cluster -n org-giantswarm gcapeverde
NAME                                                    READY  SEVERITY  REASON  SINCE  MESSAGE
Cluster/gcapeverde                                      True                     3d19h
├─ClusterInfrastructure - VSphereCluster/gcapeverde     True                     3d19h
├─ControlPlane - KubeadmControlPlane/gcapeverde         True                     3d19h
│ └─Machine/gcapeverde-4pl9j                            True                     3d19h
└─Workers
  └─MachineDeployment/gcapeverde-worker                 True                     3d18h
    └─3 Machines...                                     True                     3d19h  See gcapever
```

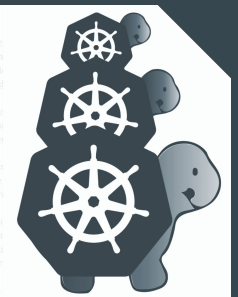


Topology

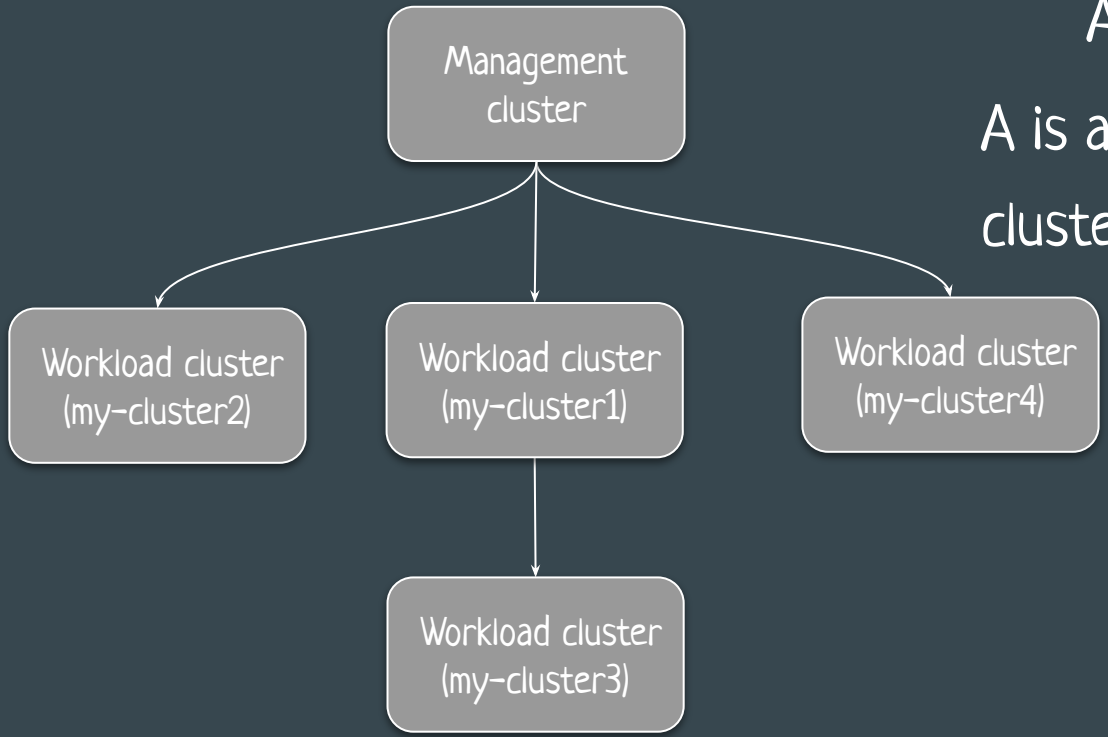


A ⟶ B

A is a management cluster for cluster B

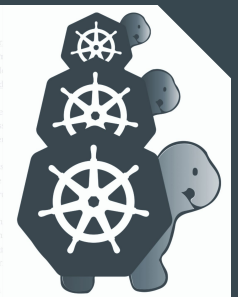


Topology

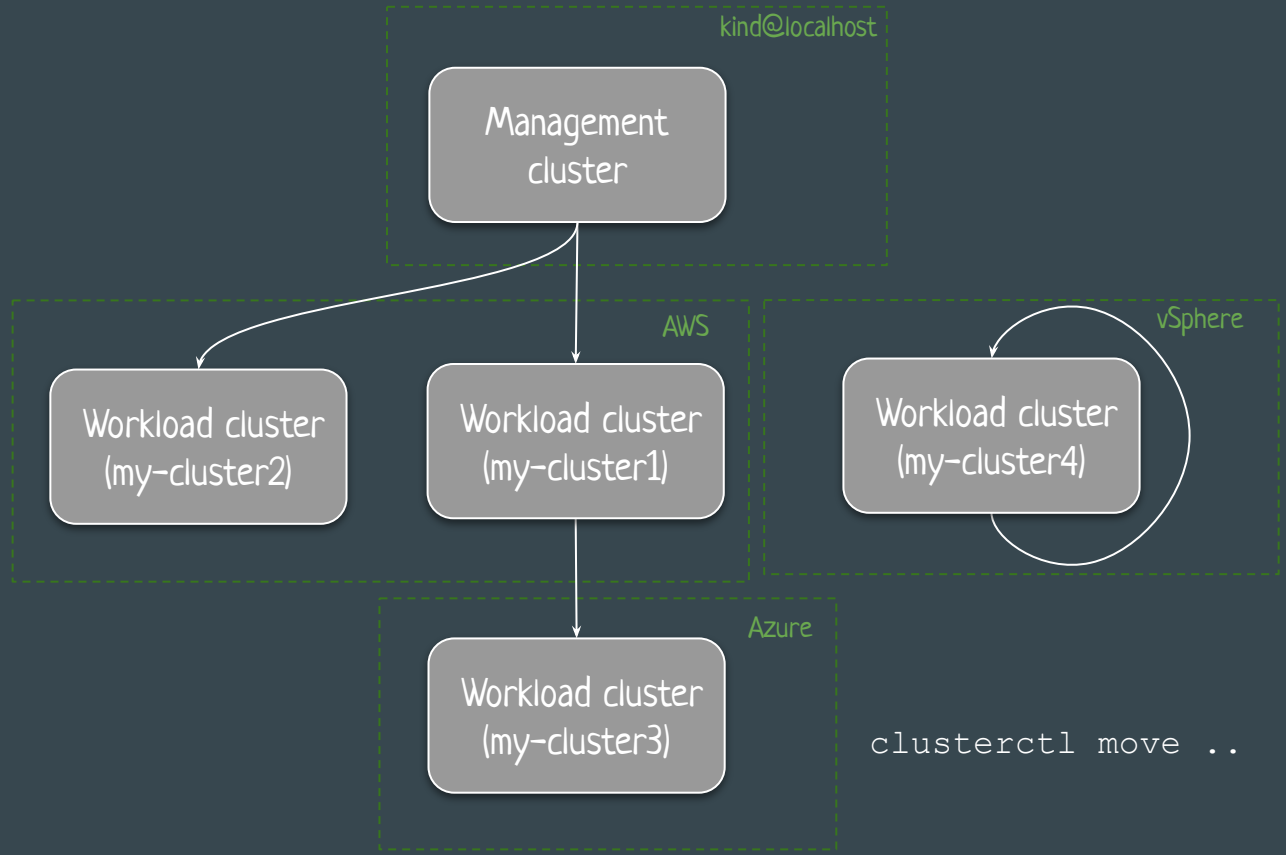


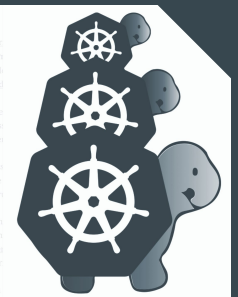
A ⟶ B

A is a management cluster for cluster B



Topology


kind@localhost
Management cluster
AWS
Workload cluster (my-cluster2)
Workload cluster (my-cluster1)
vSphere
Workload cluster (my-cluster4)
Azure
Workload cluster (my-cluster3)
clusterctl move ..



Components

- Core
- Infrastructure
- Bootstrap
- Control Plane

Components

- Core
- Infrastructure
- Bootstrap
- Control Plane

controllers	CRDs
capi-controller-manager	Cluster MachineDeployment MachineHealthCheck MachinePool Machine MachineSet ClusterClass

Components

- Core
- Infrastructure
- Bootstrap
- Control Plane

VMs, Networking, security groups (aws)

let Foo ∈ {AWS, Azure, GCP, VSphere,..}

controllers	CRDs
cap{a, z, g, v, vcd, d, ..}-controller-manager	FooCluster FooMachineTemplate FooMachine FooClusterIdentity FooControlPlane FooMachinePool ...

Components

- Core
- Infrastructure
- Bootstrap
- Control Plane

controllers	CRDs
capi-kubeadm-bootstrap-controller-manager	KubeadmConfigTemplate KubeadmConfigs

Controller converts KubeadmConfig bootstrap object into a cloud-init or ignition script that is going to turn a Machine into a Kubernetes Node using kubeadm
(also MicroK8s impl, EKS, Talos)

Components

- Core
- Infrastructure
- Bootstrap
- Control Plane

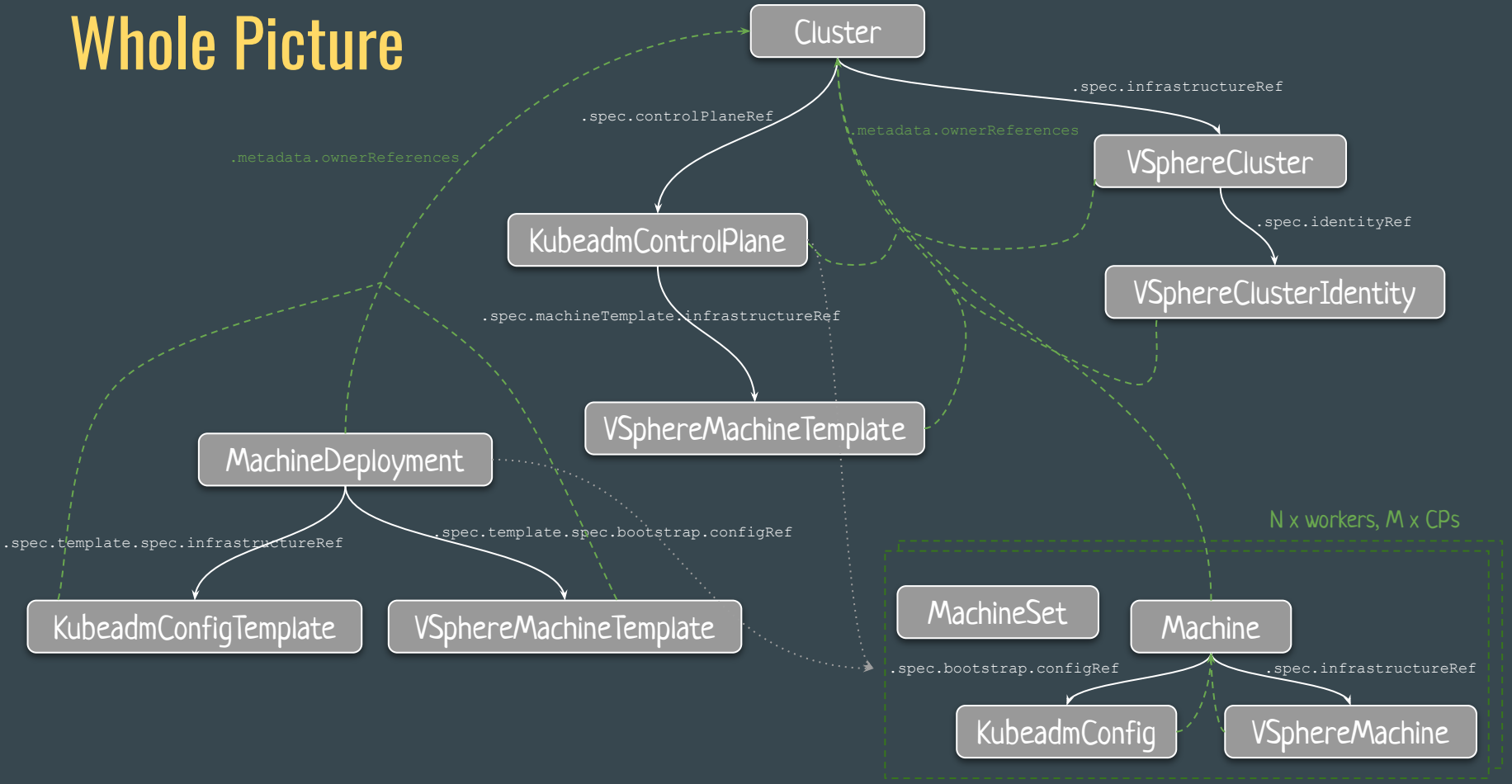
controllers	CRDs
capi-kubeadm-control-plane-controller-manager	KubeadmControlPlaneTemplate KubeadmControlPlane

etcd, coredns, image repo

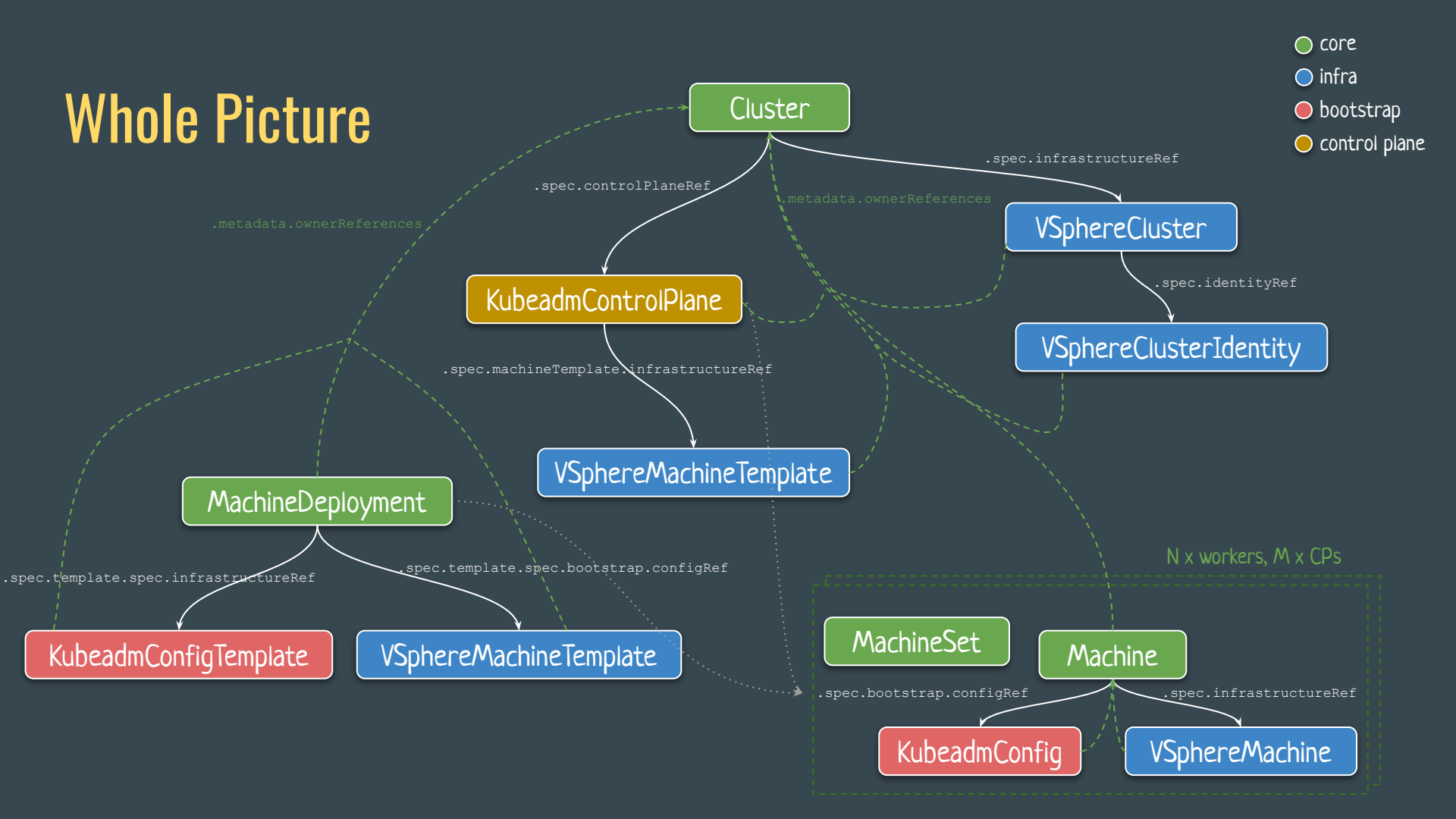
KubeadmControlPlane CR has also KubeadmConfig included

(each CP is also a node)

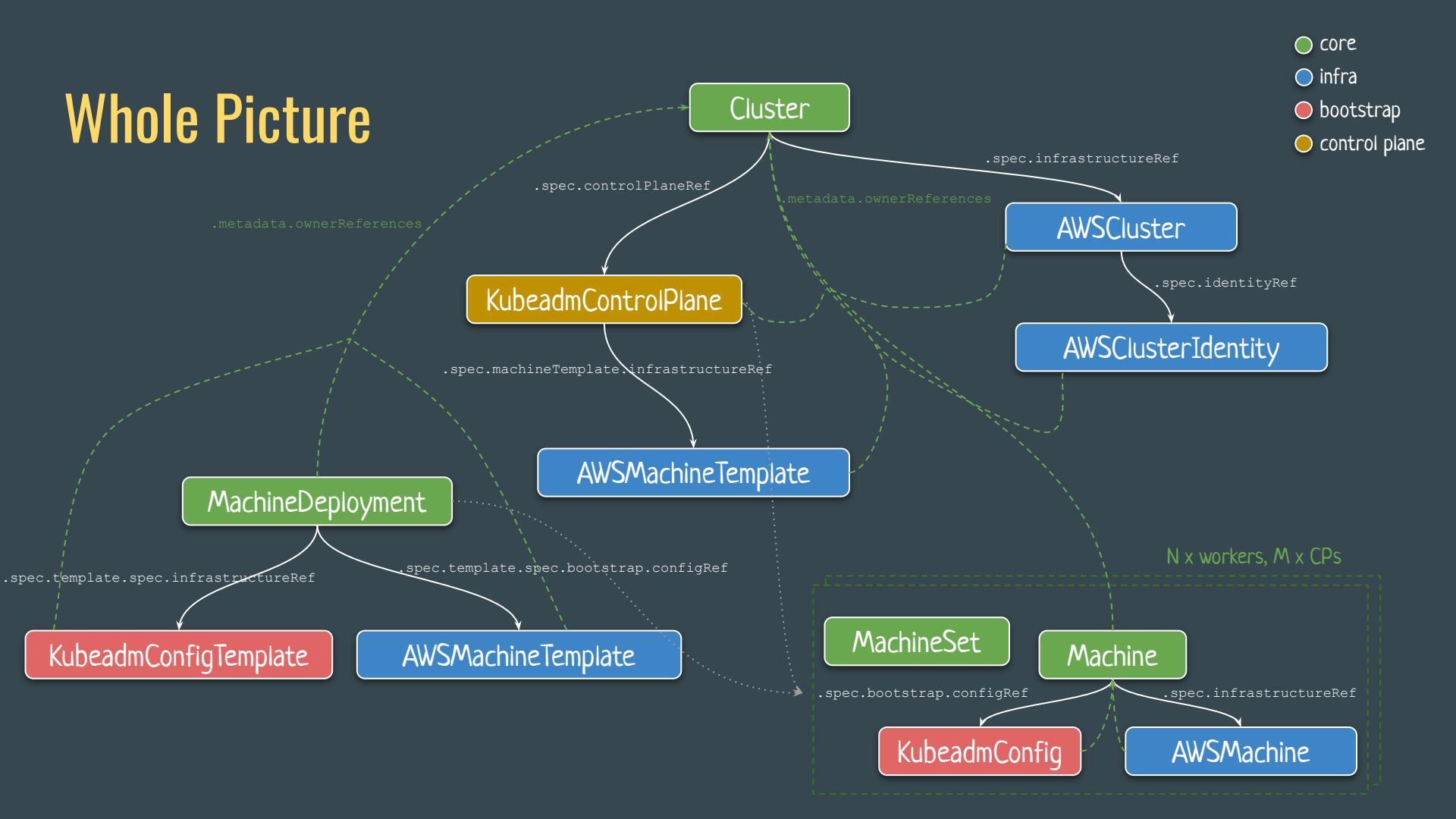
Whole Picture



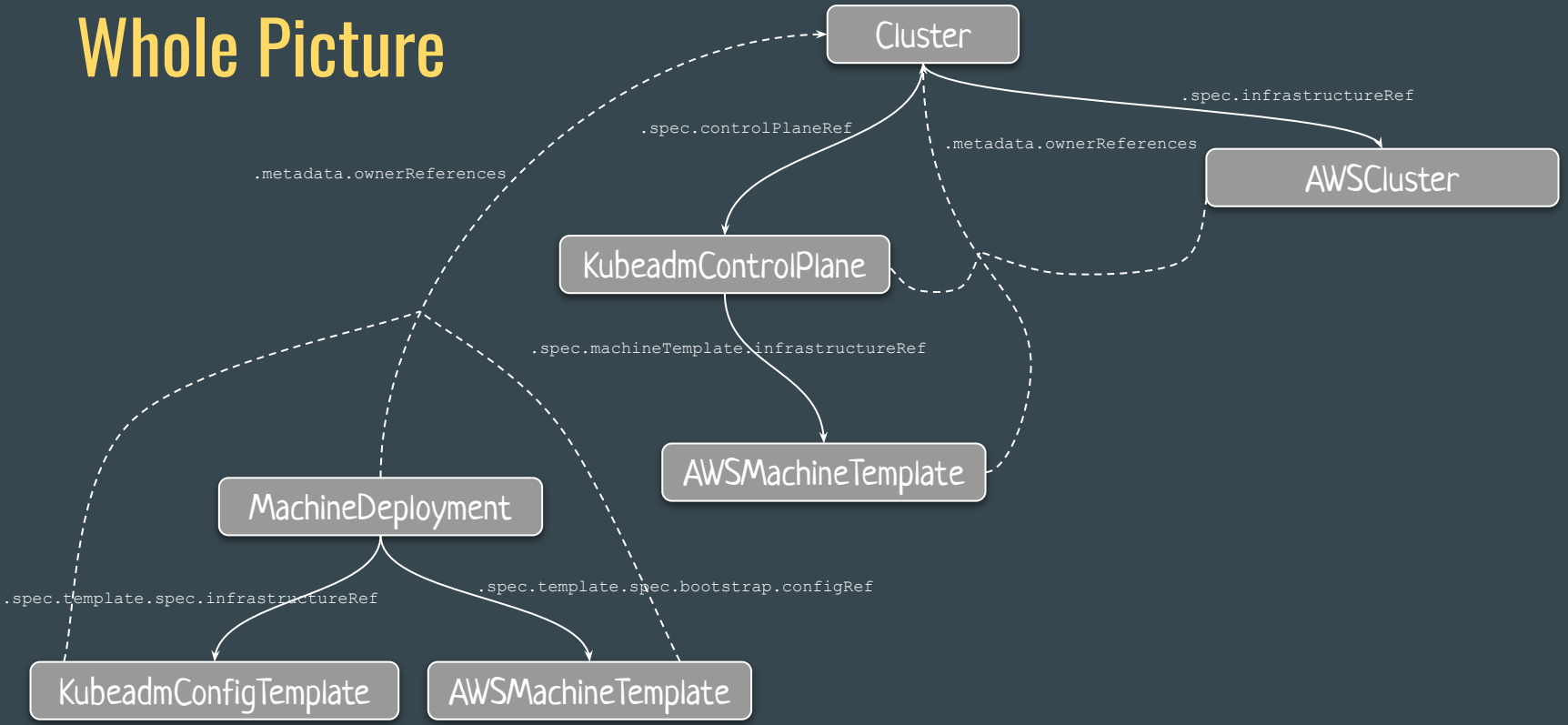
Whole Picture



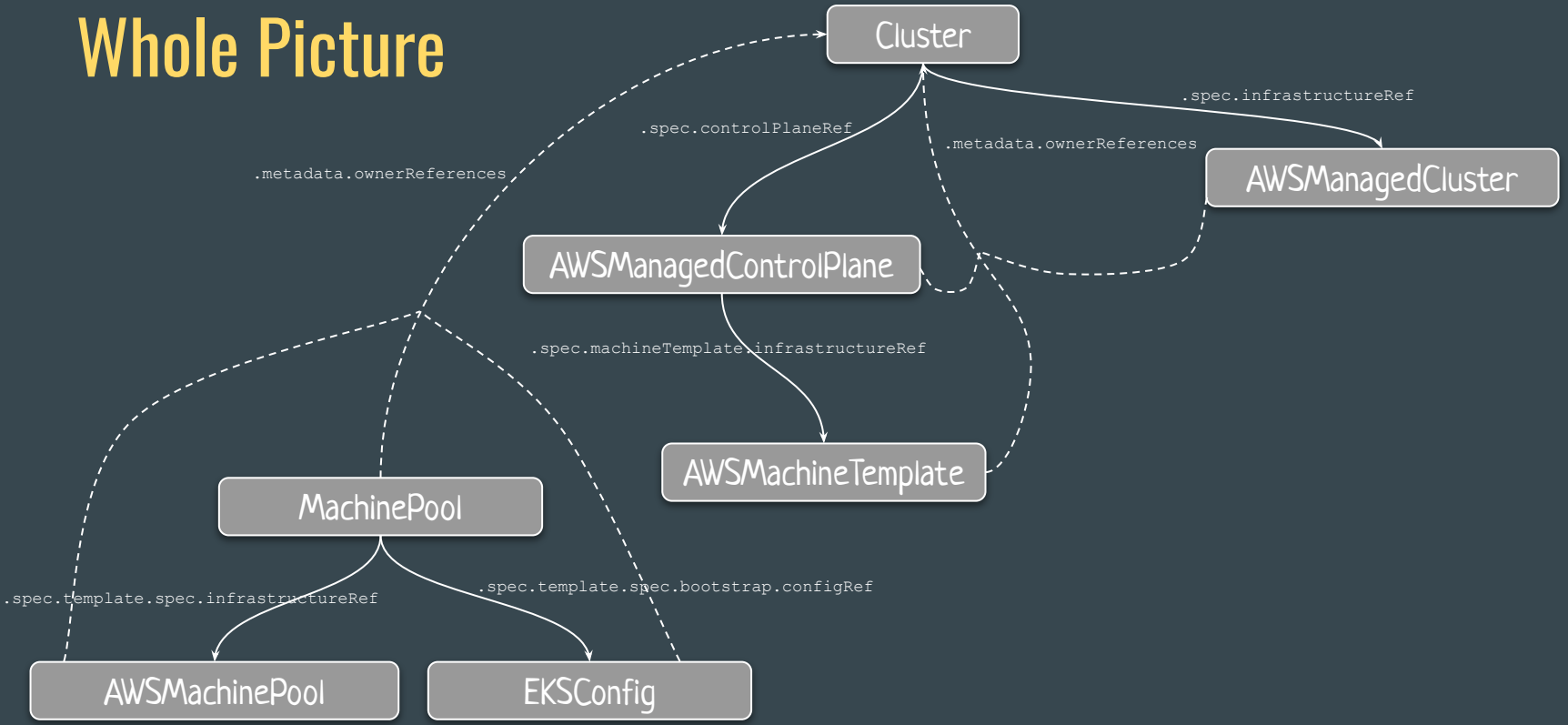
Whole Picture



Whole Picture

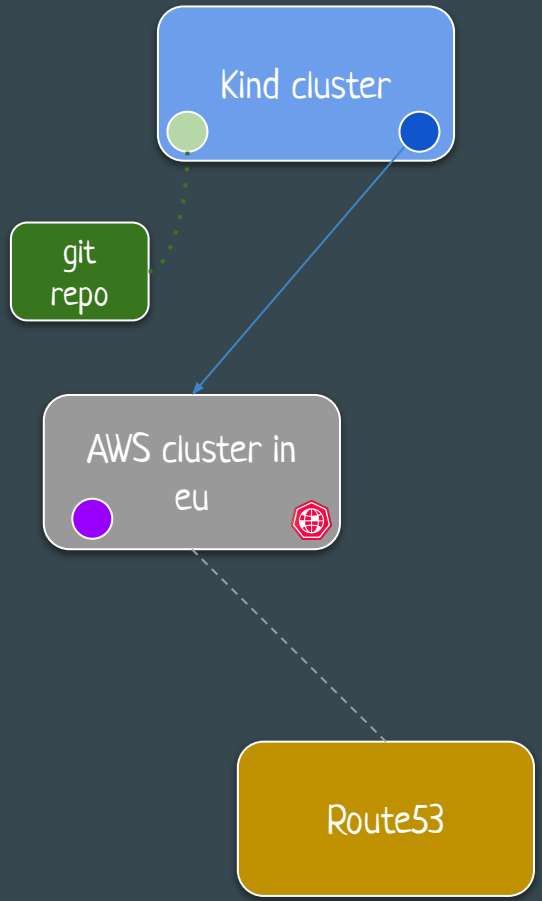


Whole Picture



Demo

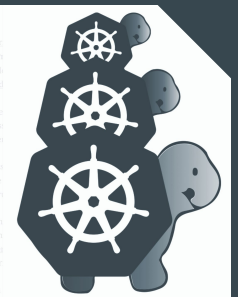
Demo



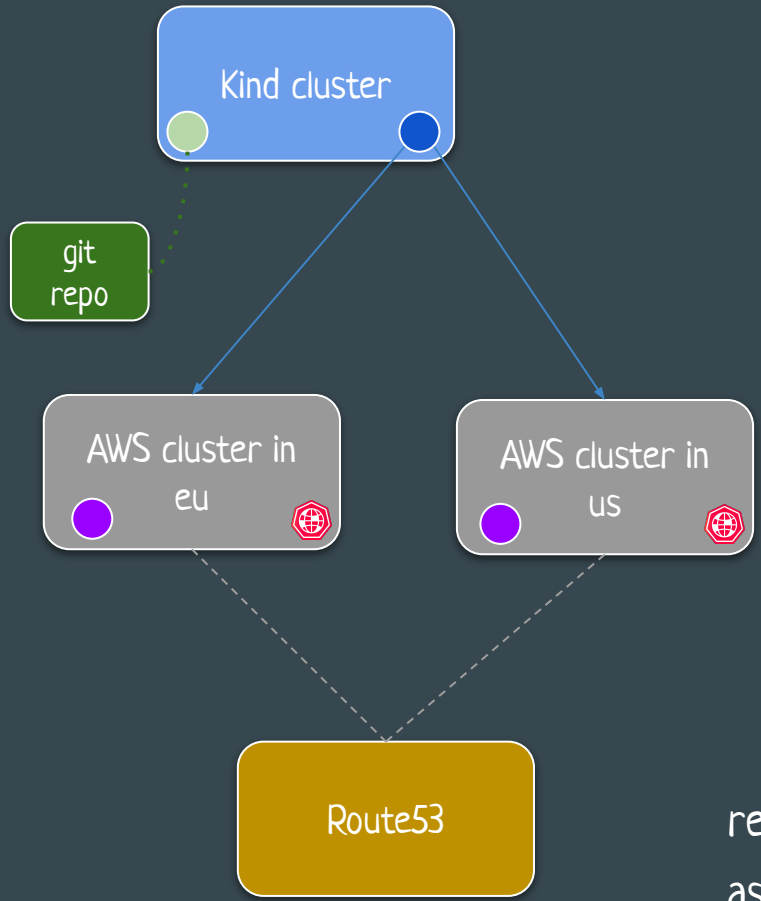
Setup

- Argo CD
- ClusterAPI
- example app
- k8gb

..then we merged a pr



Demo

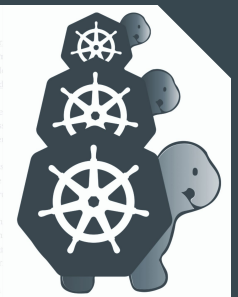


What happened?

1. Merged PR with new cluster
2. k8s cluster was created
3. c{n,s,p}i was installed
4. k8gb was installed and configured
5. example app and nginx was installed

repo: jkremser/fosdemo-clusters

ascii recording



What's there (by CAPI)

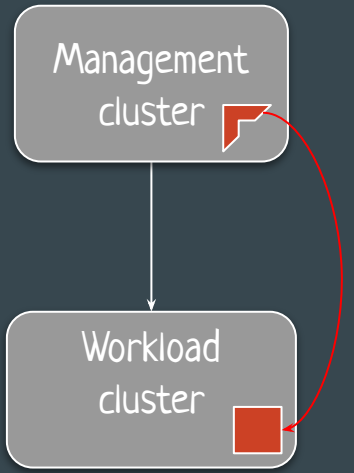
- control plane for other clusters
- scaling (nodes)
- cluster updates (k8s version)
- infrastructure as code

What's also there (bring your own YAMLs)

- CNI (Container Network Interface) - Cilium
- CSI (Container Storage Interface) - aws-ebs-csi-driver
- CPI (Cloud Provider Interface) - AWS' cloud-controller-manager
- global load balancing using k8gb
- nginx as ingress controller
- app

Content Delivery

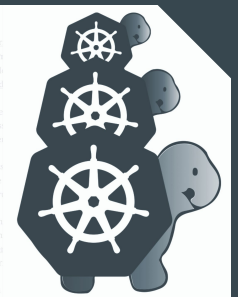
- CAPI + ClusterResourceSet
- cluster-api-addon-provider-helm + HelmChartProxy
- Flux + HelmRelease
- secrets? (SOPS, Vault CSI / agents)
- operator in MC doing something to WCs -> custom



How to start

- CAPD
  a. create local cluster with Kind
  b. `clusterctl init --infrastructure docker`
  c. `clusterctl generate cluster foo | k apply -f -`

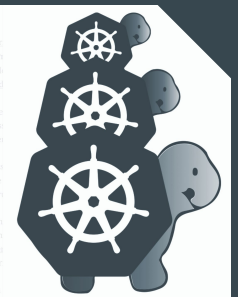
- k8gb - local playground setup described at k8gb.io



Takeaways

- k8gb and other distributed systems that require similar configuration across multiple clusters fits well with CAPI

- we achieved real high availability as code



Thank You!

Q&A

bit.ly/k8gb-capi

jiri.kremser@gmail.com