



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

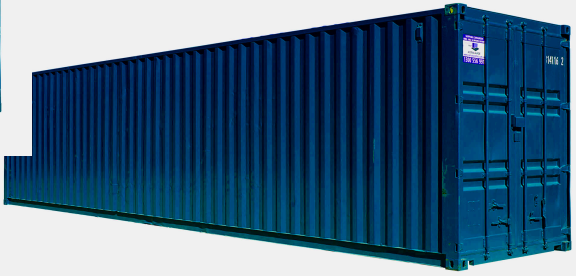
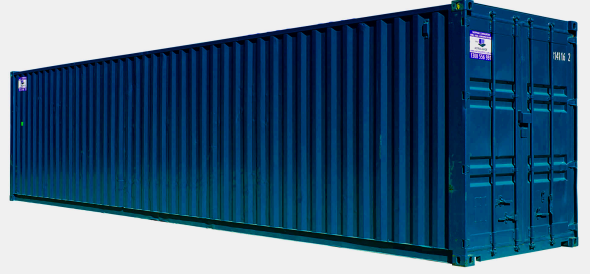
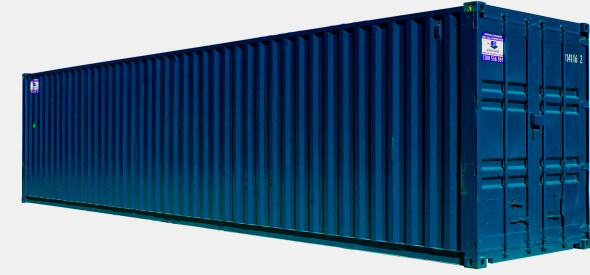
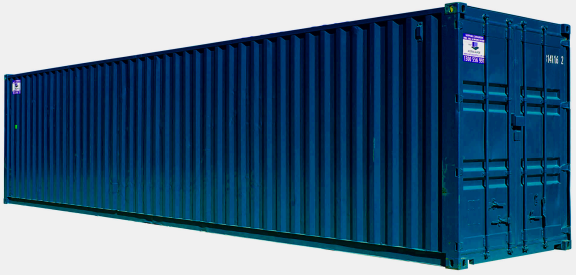
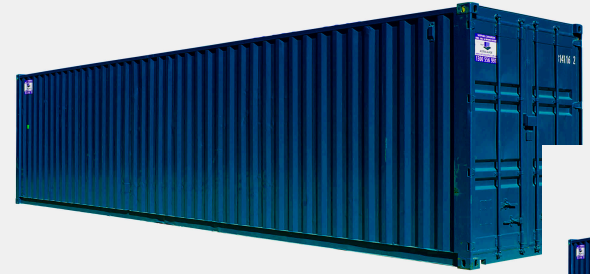
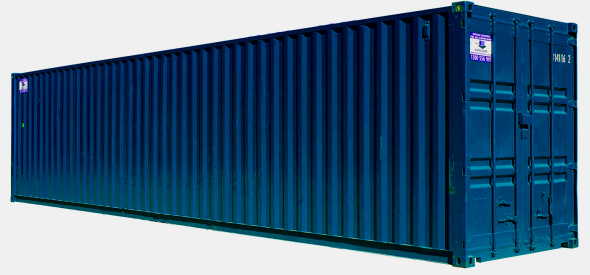
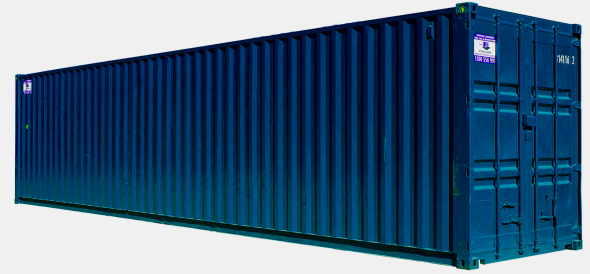
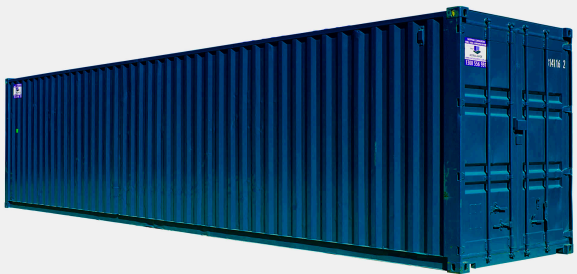
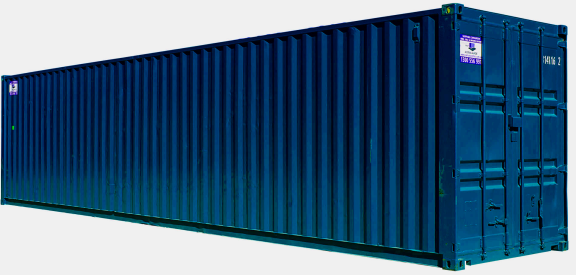
# Controllers at Chaos

*Kesavan Subramanian & Gaurav Gupta, SAP*

*The Basics*

A close-up photograph of a hand holding a white chalk marker, writing the words "The Basics" in a cursive script on a dark, textured chalkboard. The hand is positioned on the right side of the frame, with the marker tip touching the end of the word "Basics". The lighting is soft, highlighting the texture of the chalkboard and the smooth surface of the hand.











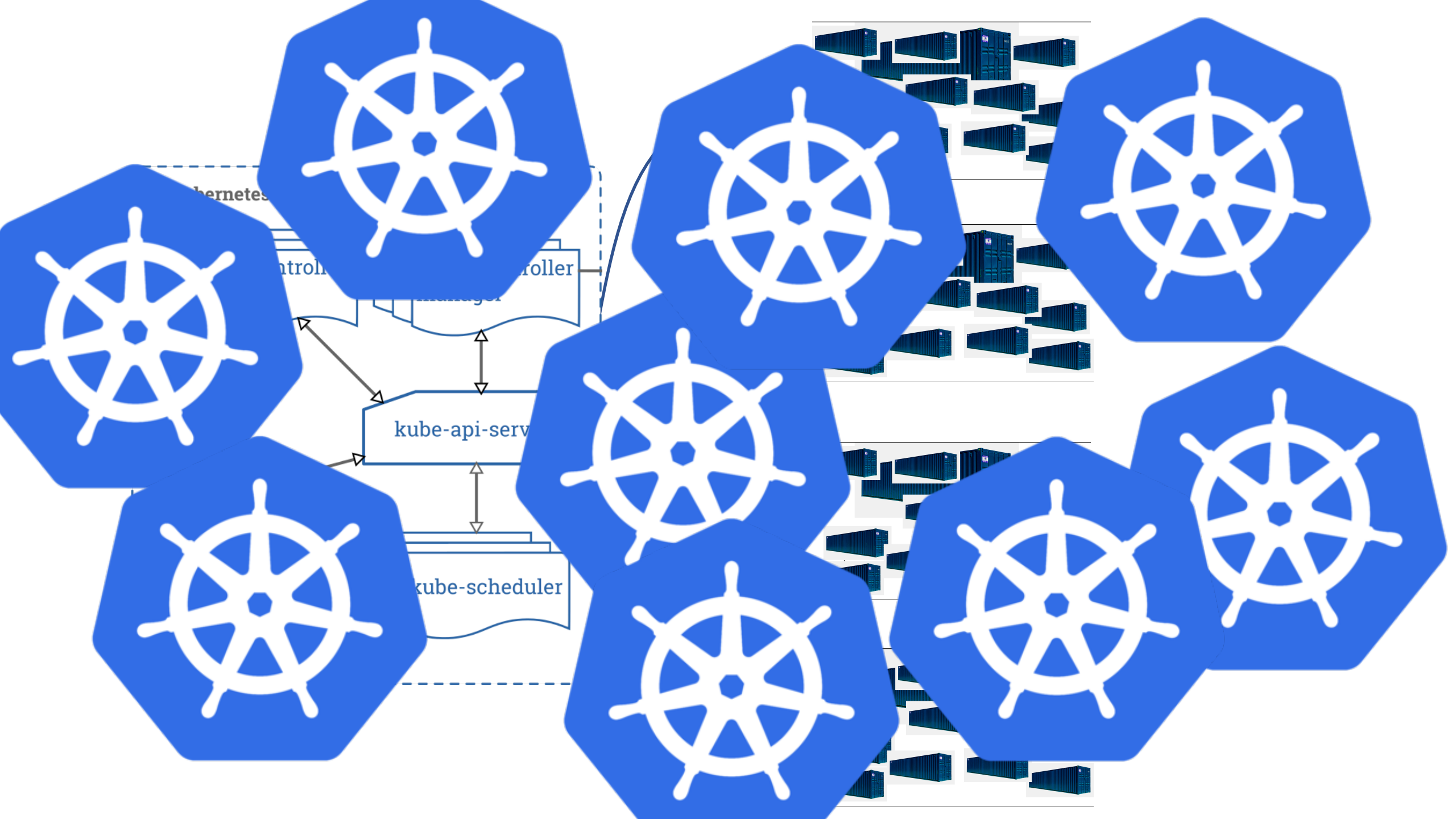


kubernetes.io







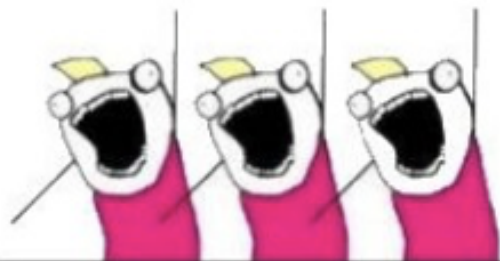




**WHAT DO WE WANT?**



**KUBERNETES CLUSTERS**



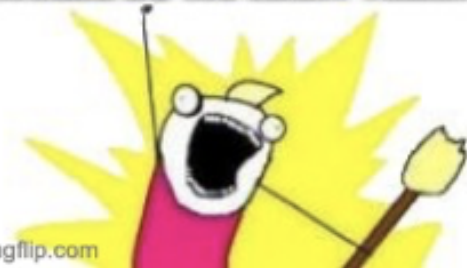
**HOW MANY DO WE WANT?**



**1000'S**



**WHERE DO WE WANT THEM?**



**GCP AWS AZURE...  
PRIVATE CLOUD**









vSphere

OpenStack

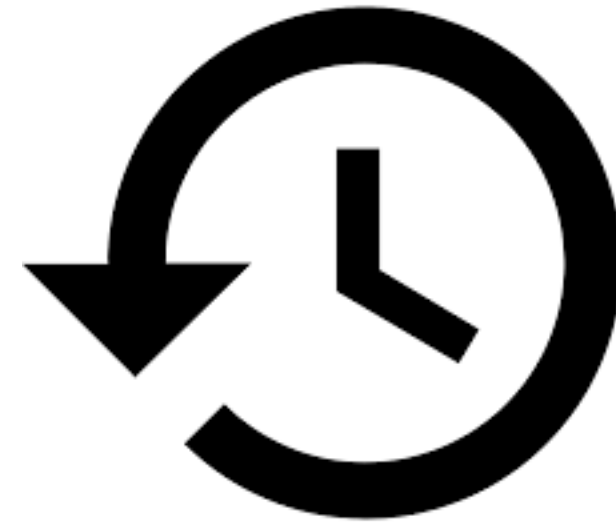
AWS

GCP

AliCloud

Azure





Backup and Restore

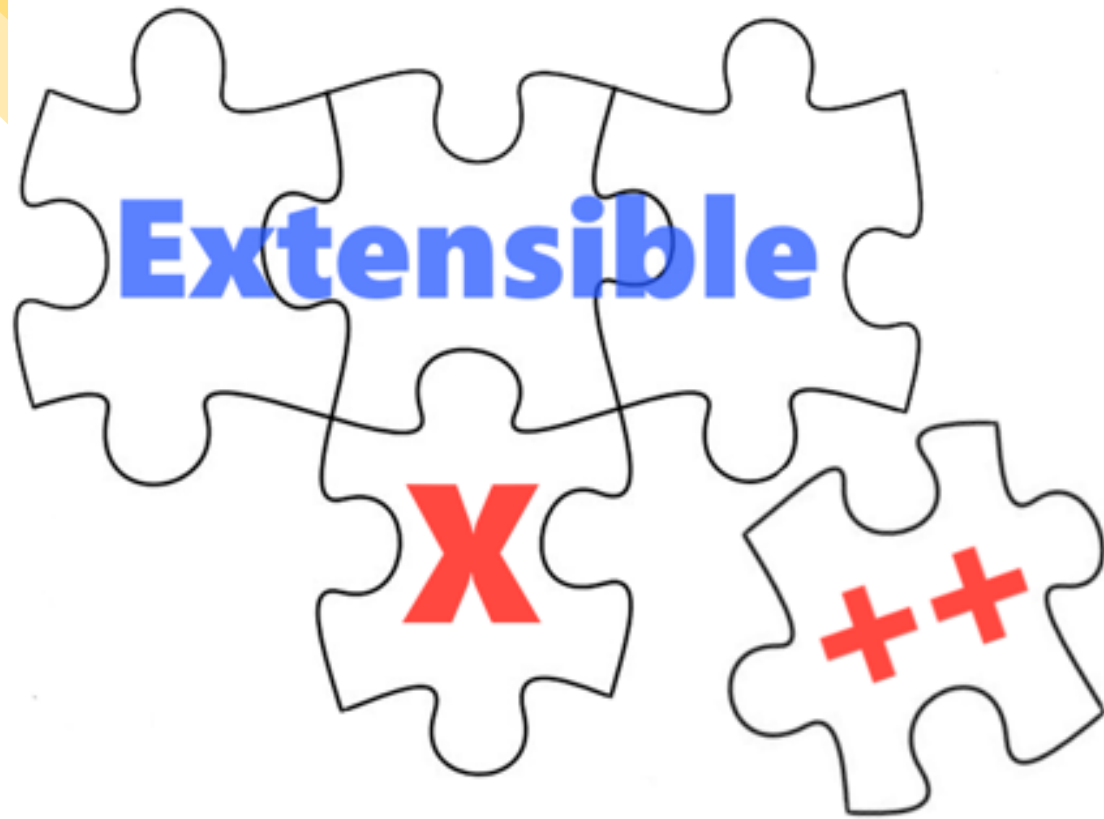


```
kind: my-cluster
apiVersion: api/version
metadata:
  name: my-k8s-cluster
spec:
  kubernetes:
    version: 1.18.5
  networking:
    type: calico
    pods: 100.96.0.0/11
    nodes: 10.250.0.0/19
    services: 100.64.0.0/13
  provider:
    type: gcp
  workers:
    - name: worker-nlsg6
      machine:
        type: n1-standard-2
        image:
          name: coreos
          version: 2512.3.0
        maximum: 10
        minimum: 1
        maxSurge: 1
        maxUnavailable: 0
        volume:
          type: pd-standard
          size: 50Gi
      zones:
        - europe-west1-d
  region: europe-west1
```

```
<-----
|
| >- Networking
|
<-----
cloud provider
<-----
Machine type
<-----
OS image
<-----
Region
```



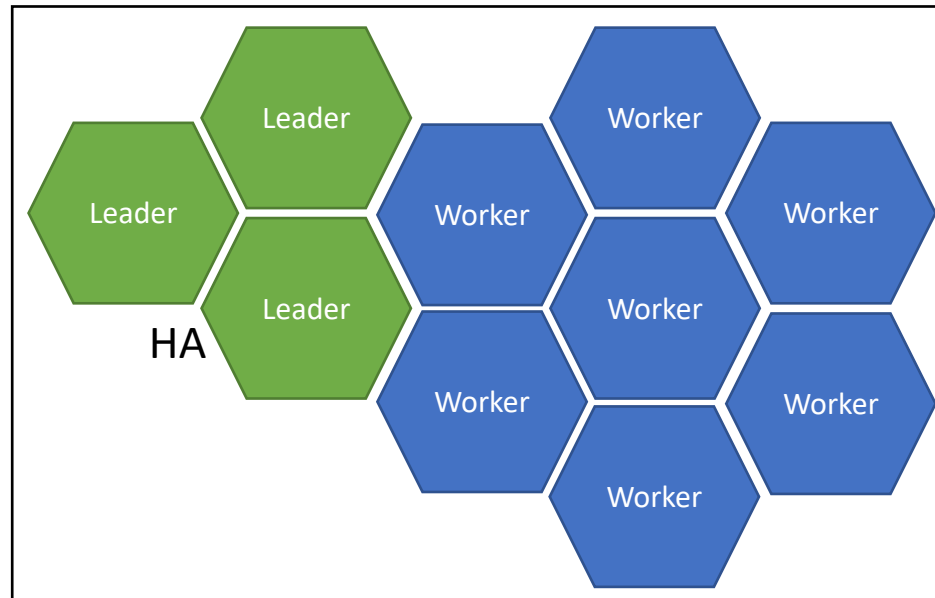
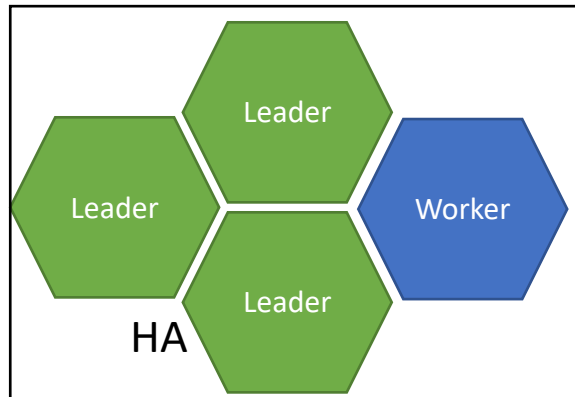
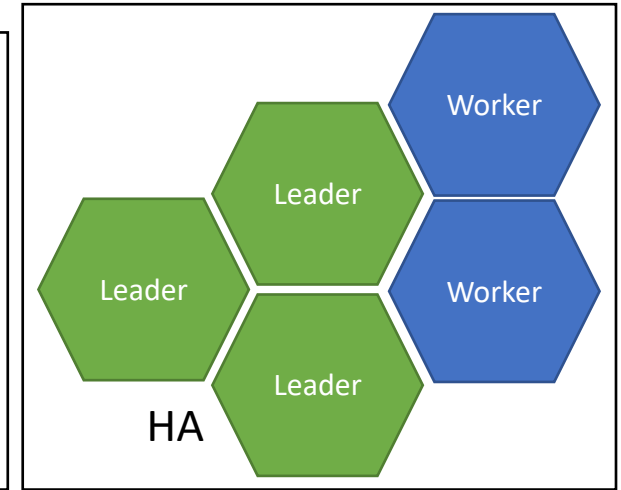
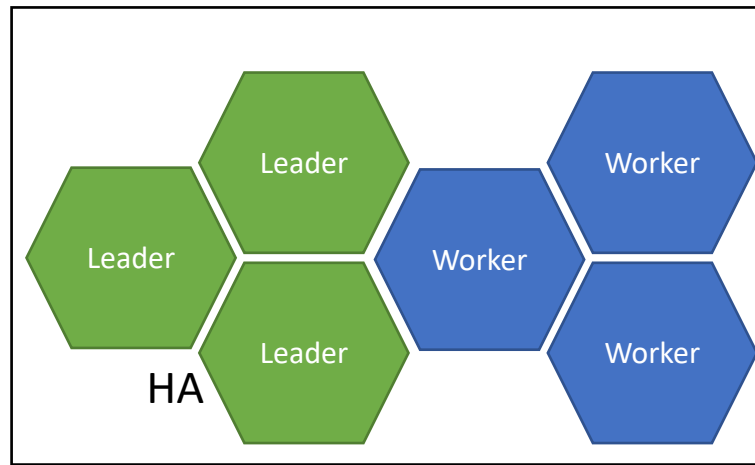
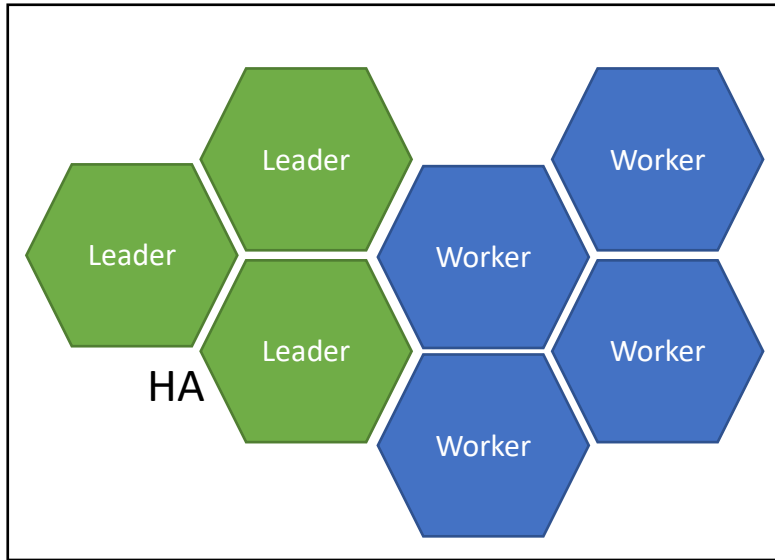




Bring your own Cloud




# Common k8s cluster setup





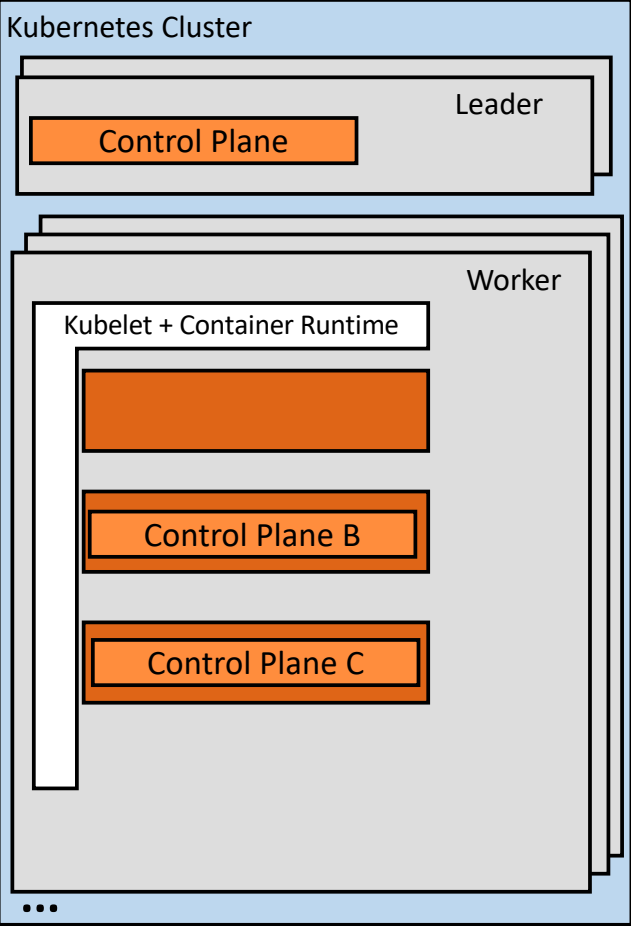
Gardener cluster

Seed cluster

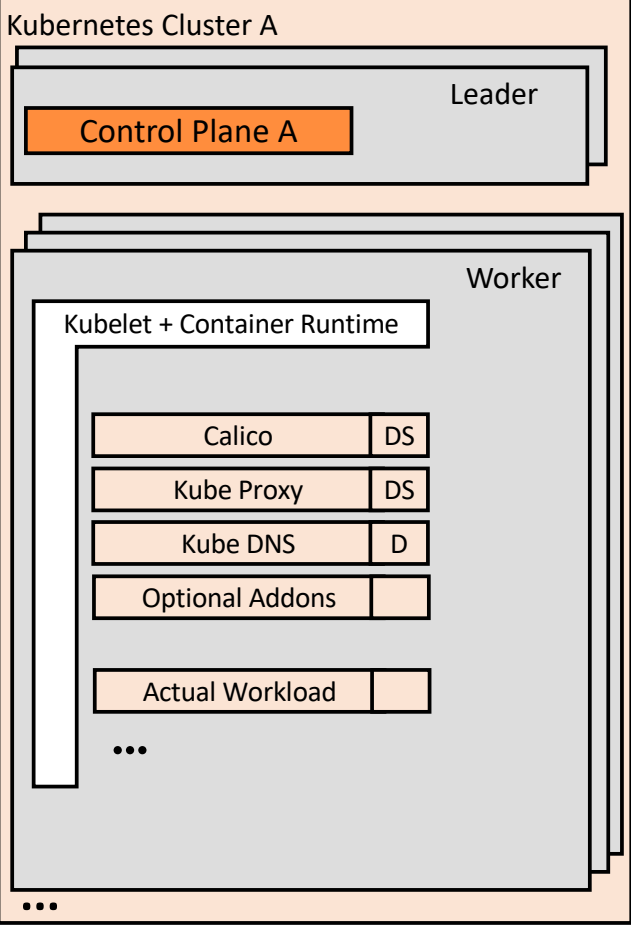
End-User Cluster (Shoot Cluster) 



Target IaaS/Account



Target IaaS/Account

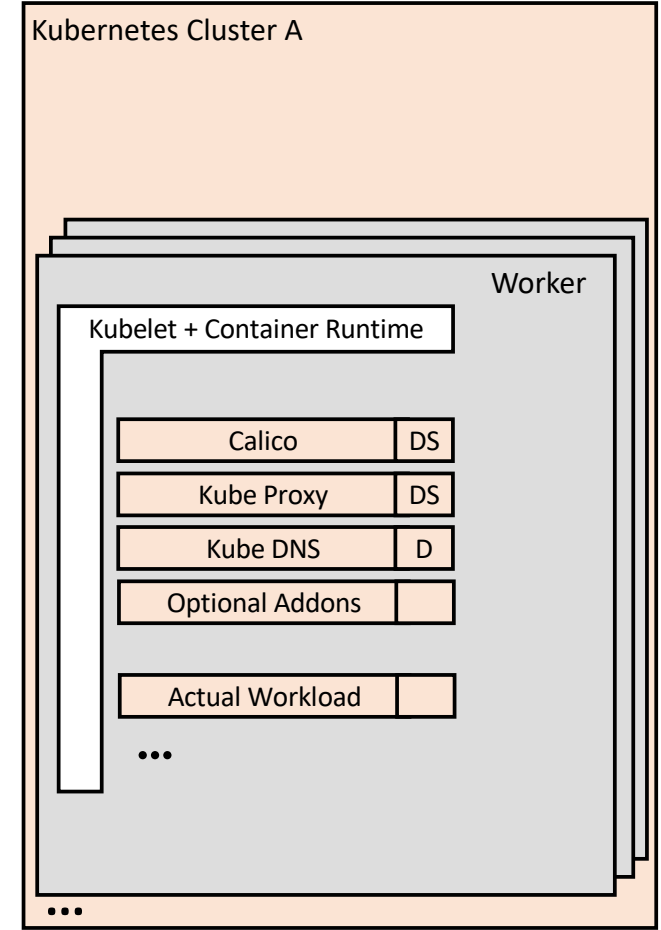
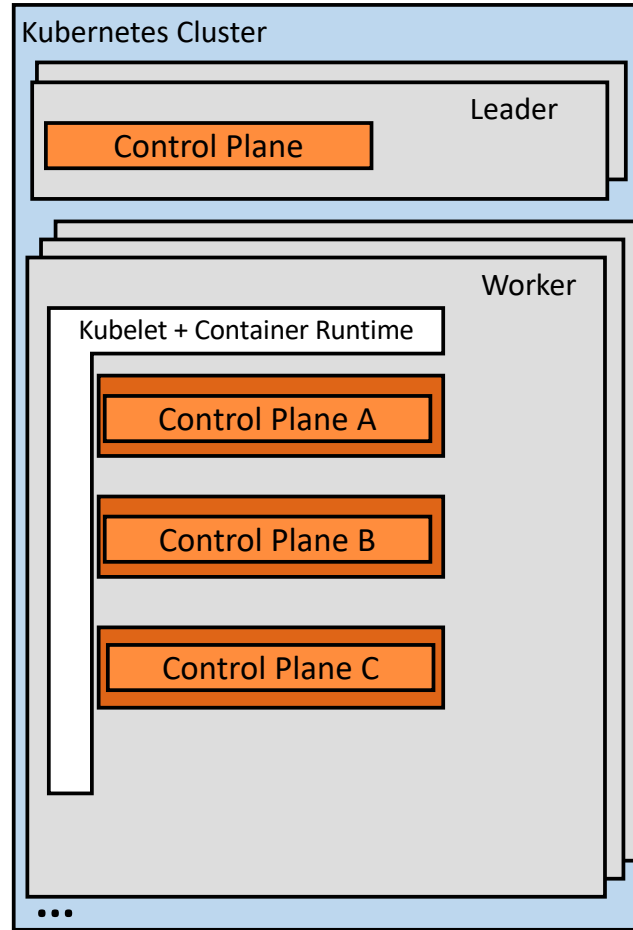
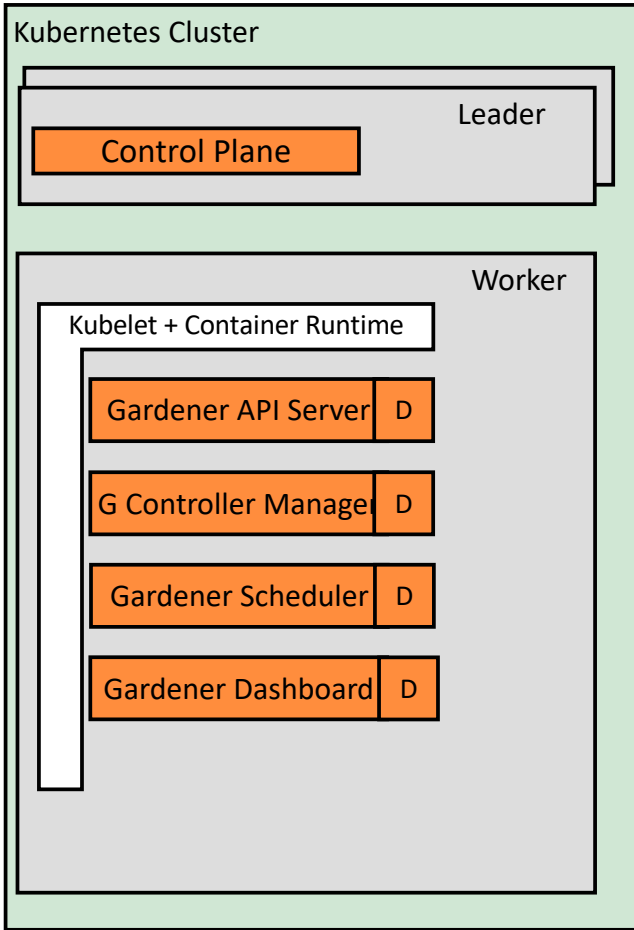


Target IaaS/Account

Gardener cluster

Seed cluster

End-User Cluster (Shoot Cluster)



Target IaaS/Account

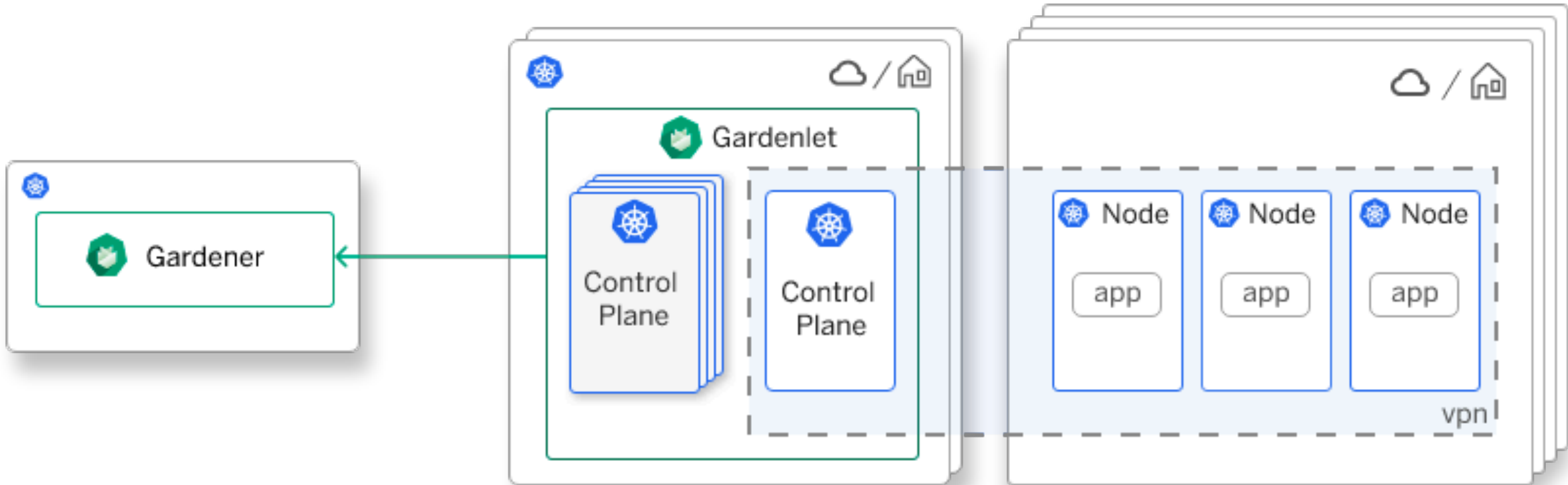
IaaS/Account

Target IaaS/Account



## Make It All About Kubernetes Again

Gardener abstracts environment specifics to deliver the same homogeneous Kubernetes-native DevOps experience everywhere



Isn't a Simple Architecture

Let's Add chaos in it

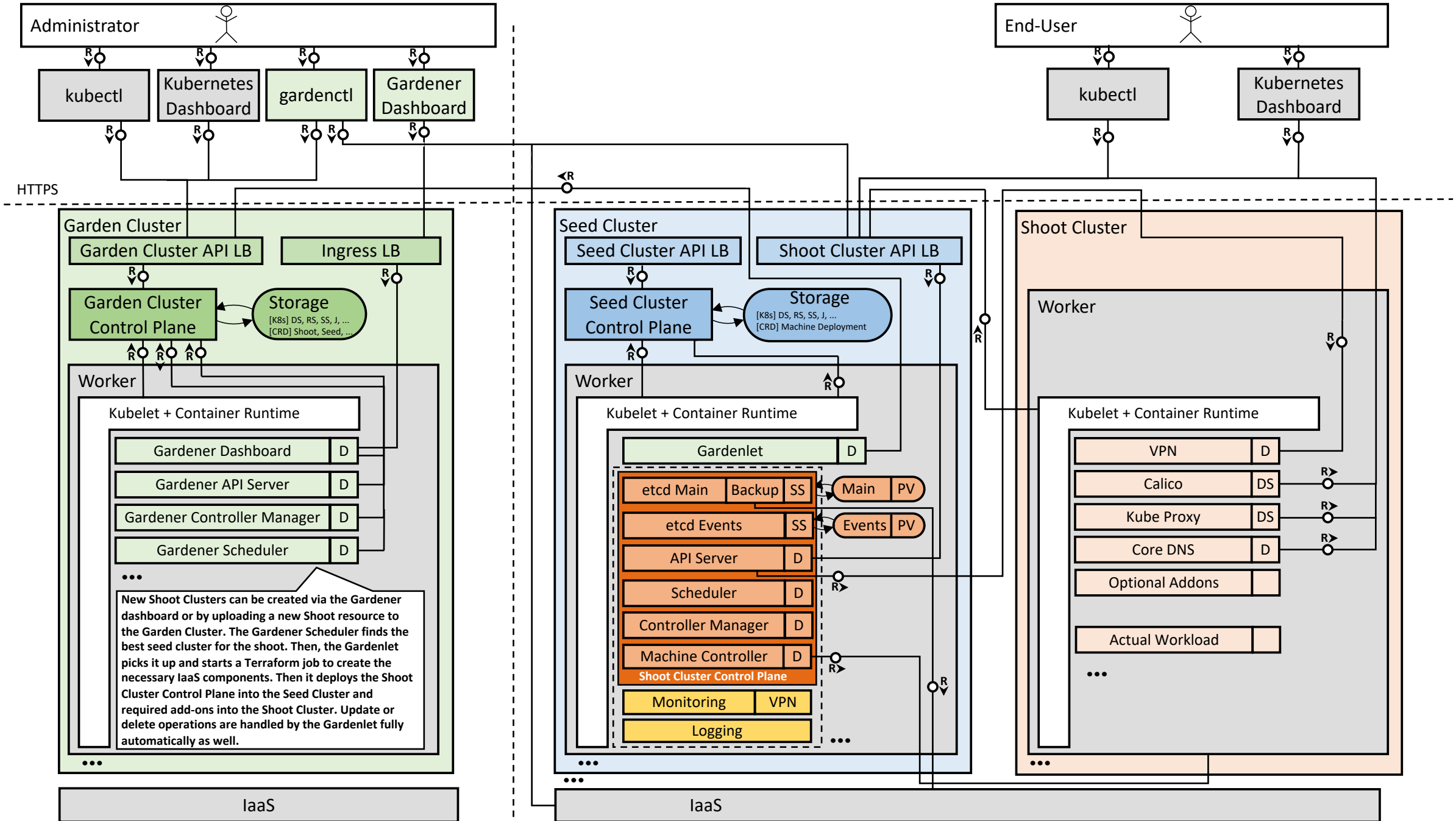




# Garden Cluster

# Seed Cluster

# Shoot Cluster



Let's speak K8s

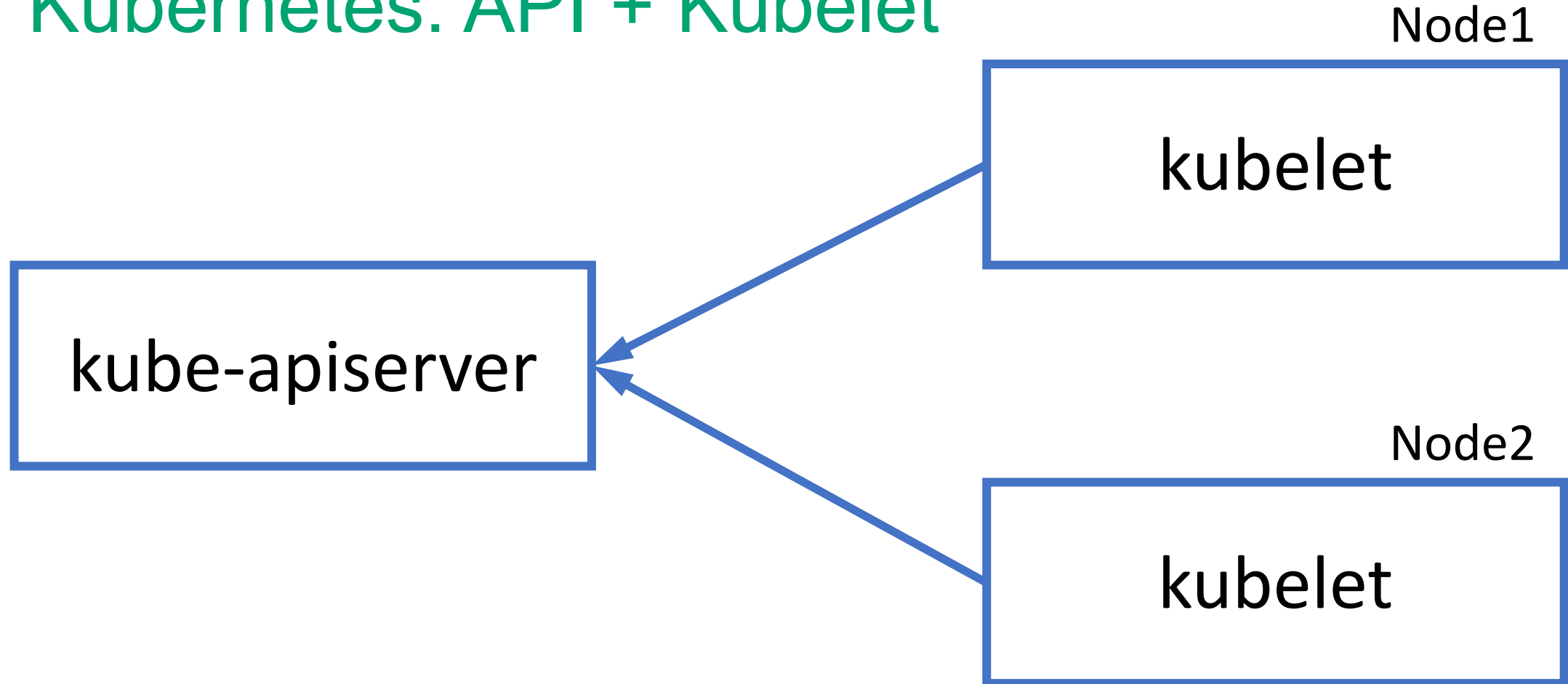


## Primary **Gardener** Design Principle

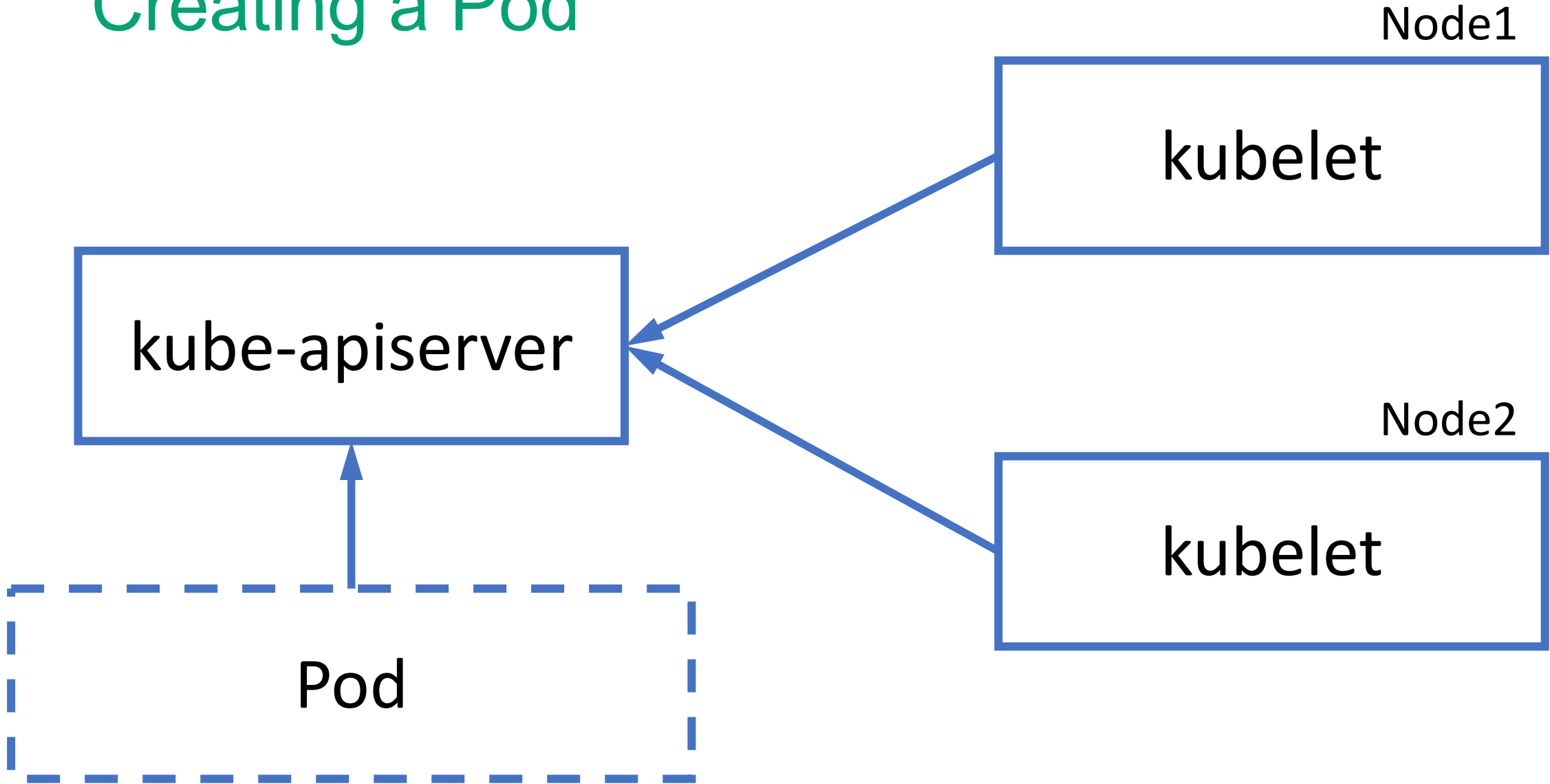
Do not reinvent the wheel and learn one concept and apply it uniformly ...

Let Kubernetes drive the design of  
the **Gardener**.

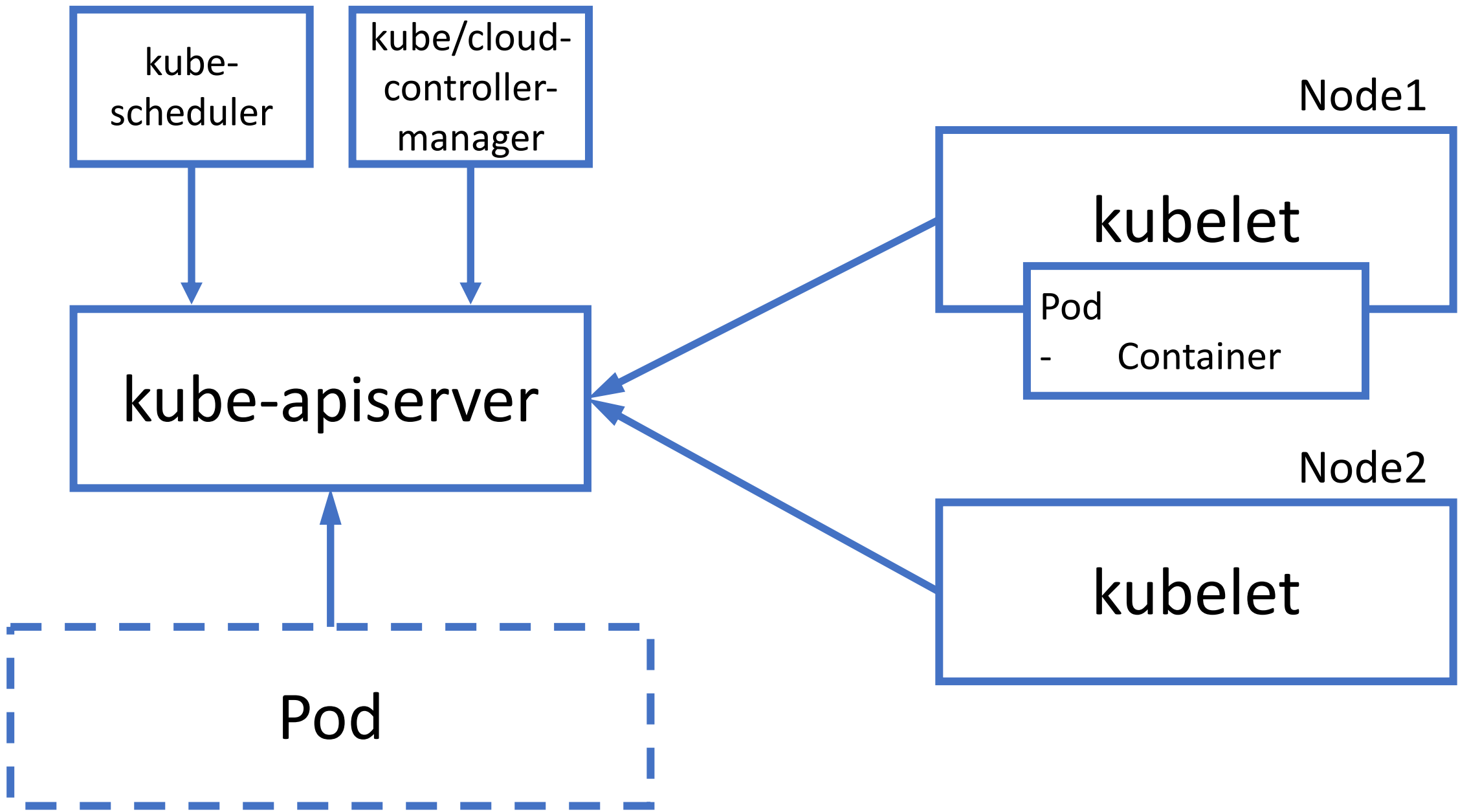
# Kubernetes: API + Kubelet



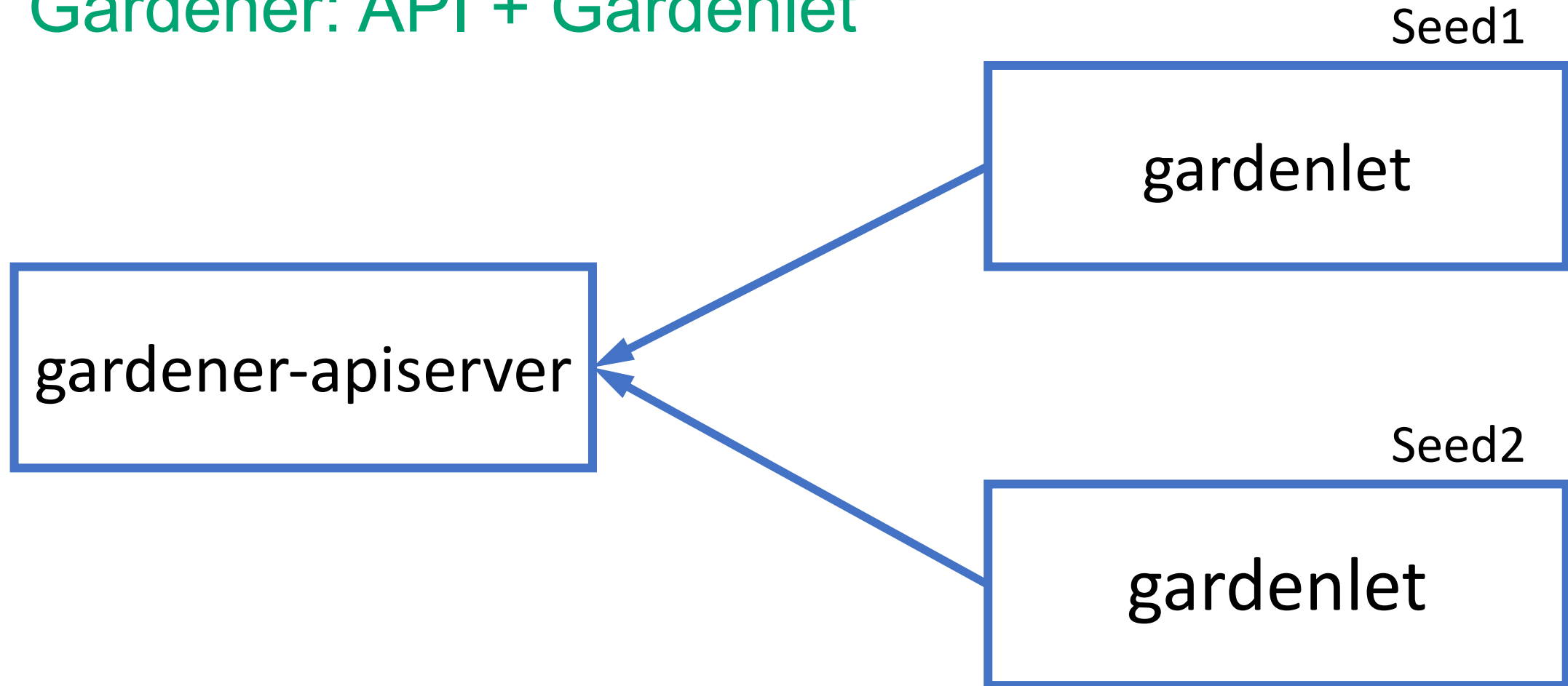
# Creating a Pod



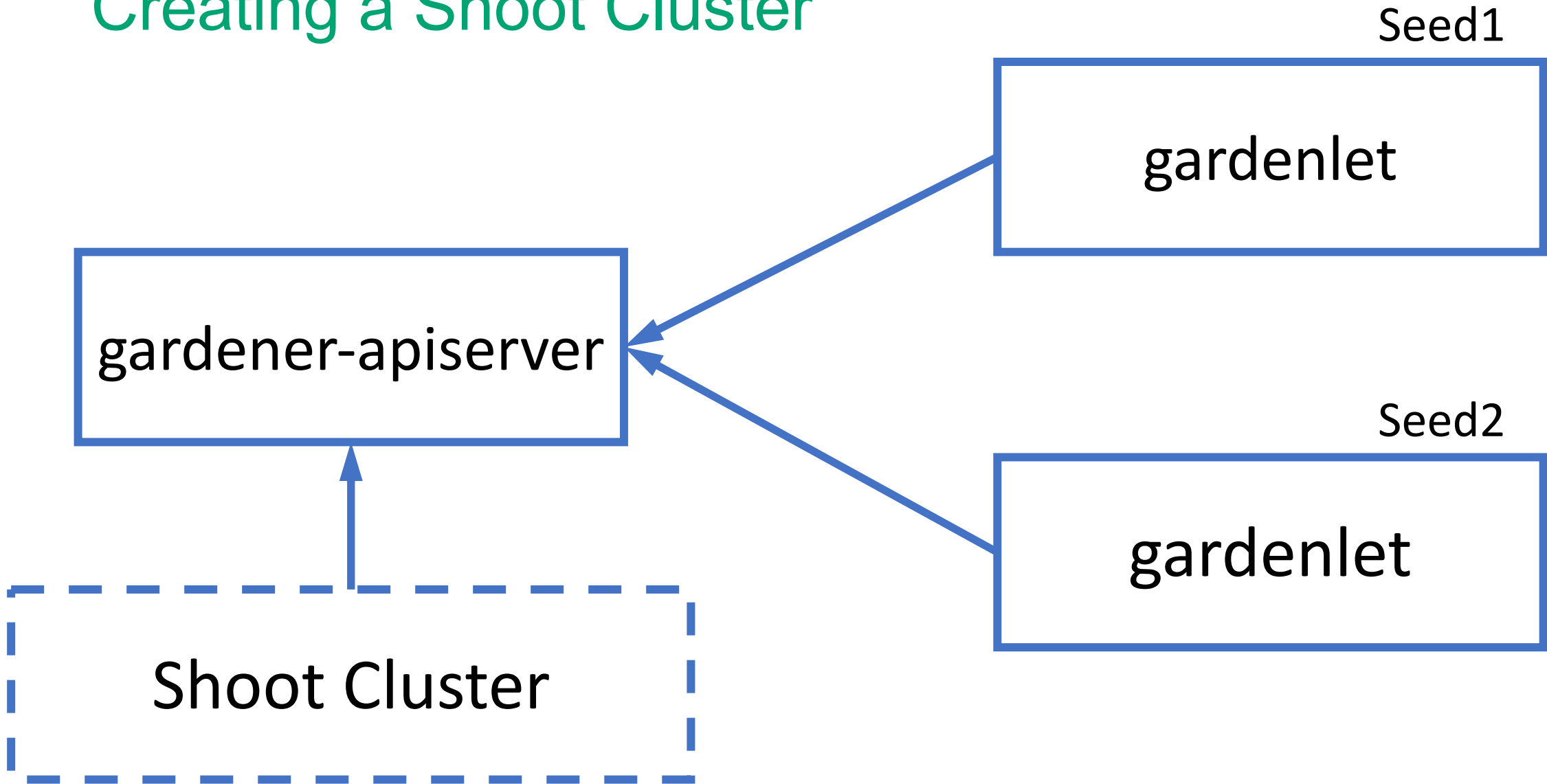




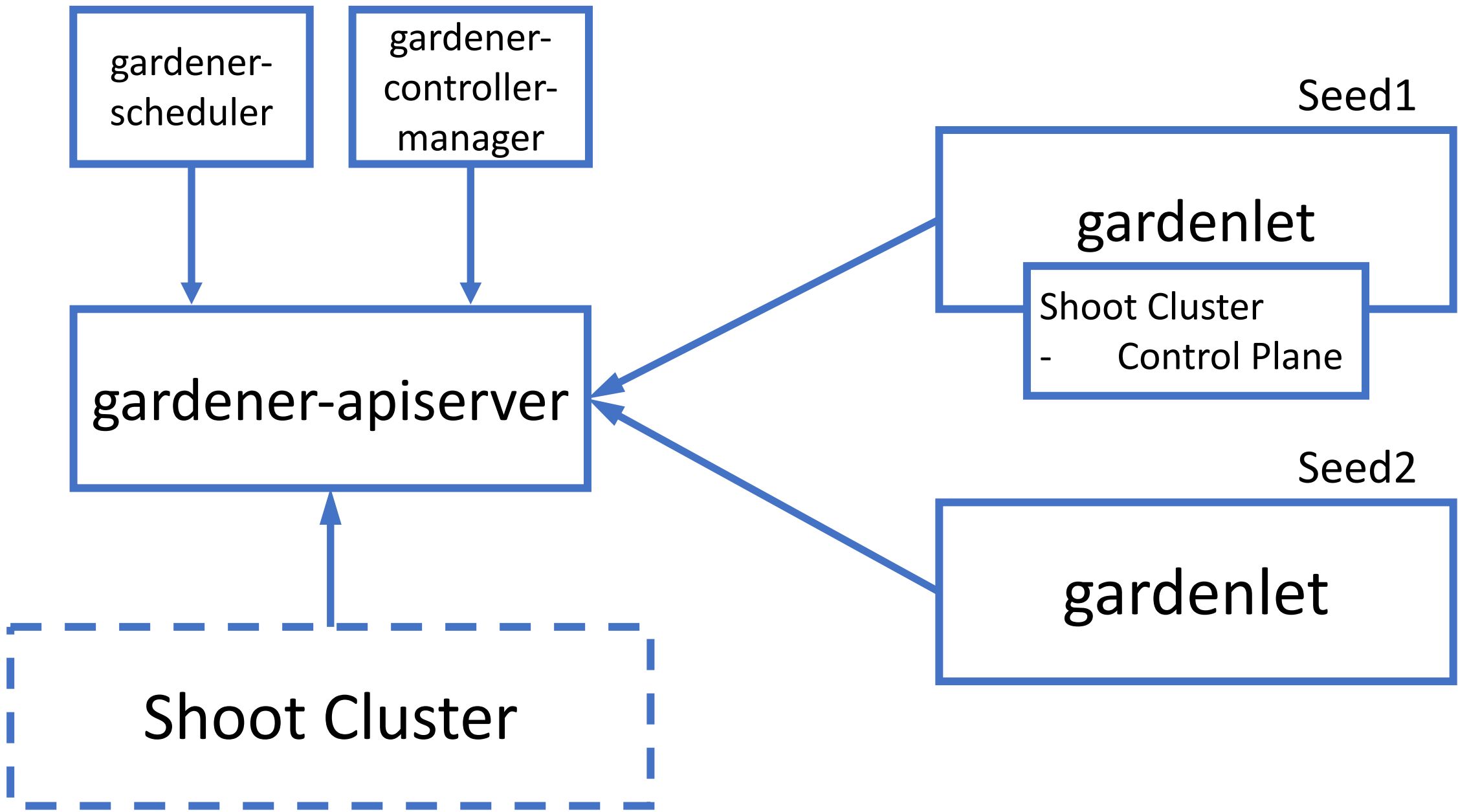
# Gardener: API + Gardenlet



# Creating a Shoot Cluster

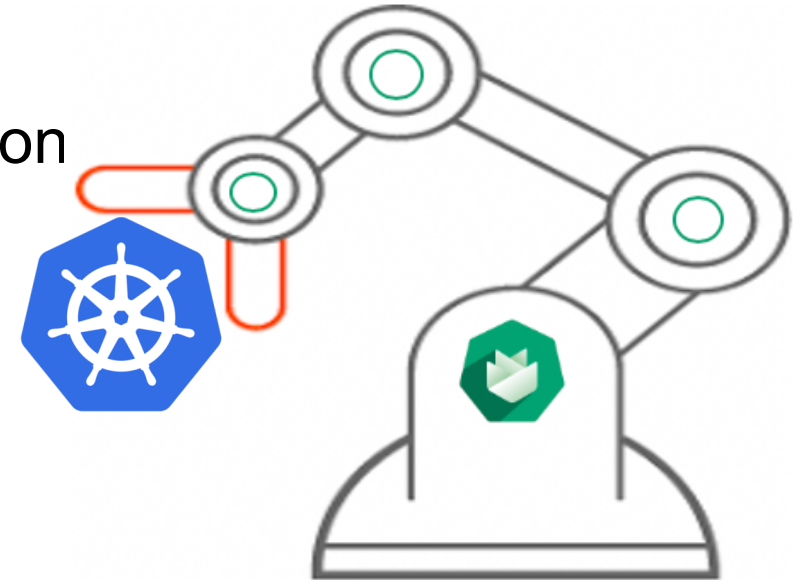






# Operations at Scale

- Gardener Controller Manager
  - Single Controller Manager to manage all shoots on the seed clusters
  - Suits to operate thousands of clusters.
- True Scalability
  - Beyond the capacity of a single controller-manager
  - Distribute Controller logic to work independently



# Kubelet

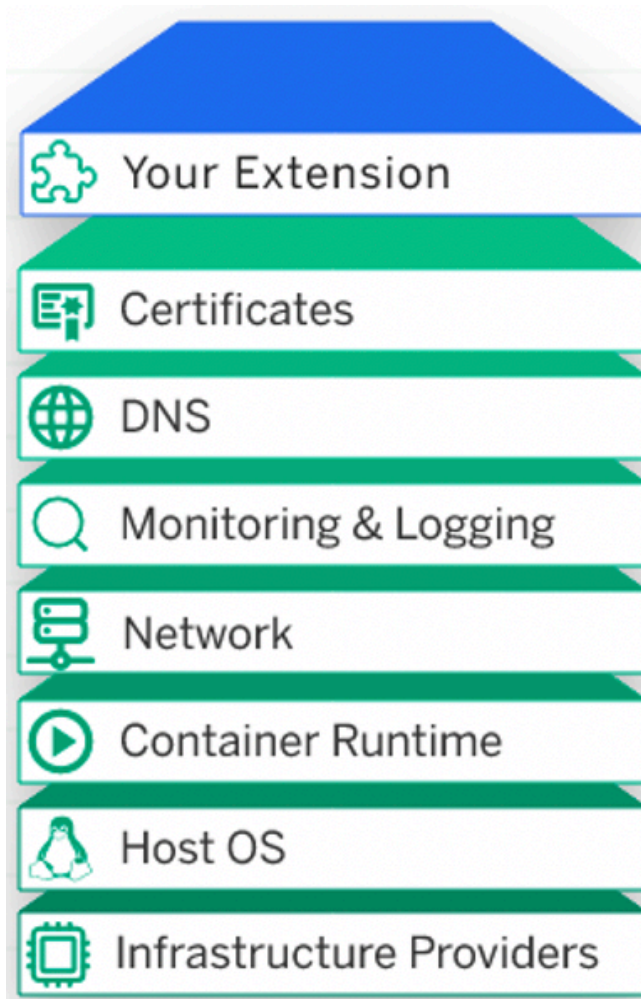
Well known story of kubelet

*“Primary node agent runs on each node, responsible for managing pods and containers in particular”*



# Gardenlet

- Agent on every seed cluster to manage shoot clusters in that seed
- Takes over the job from Gardener Controller manager in reconciling the shoot clusters
- Similar way of using lease objects for node heartbeats, gardenlet uses lease objects for seed heart beats
- Pave the ways to grow and operate as many shoot clusters
- Not necessary to run inside the Seed cluster as long as it can talk to the seed's API server
- Opens up doors in placing shoot clusters behind firewalls



# Extension Controllers



Bring your own  
Cloud



Plug and Play

# Machine Controller Manager

- Machine Controller Manager (MCM) manages VMs as another Kubernetes custom resource
- Provides a declarative way to manage VMs

Machine Deployment Controller

Deployment

Machine Deployment





# Machine Controller Manager

- Machine Controller Manager (MCM) manages VMs as another Kubernetes custom resource
- Provides a declarative way to manage VMs

Machine Set Controller

ReplicaSet

Machine Set



# Machine Controller Manager

- Machine Controller Manager (MCM) manages VMs as another Kubernetes custom resource
- Provides a declarative way to manage VMs

Machine Controller

Pods

Machines



# Dependency Watchdog

- If etcd is down, apiserver & controllers can go in CrashLoopBackOff
- Deletes pods in CrashLoopBackOff
  - New pods start as soon as apiserver is up



# Cluster Autoscaler

- Forked and adapted to work with Machine Deployments
- Autoscales seed/shoot cluster worker pool

Can result in downtime if etcd is scheduled on a scaled down node

# Dedicated etcd worker pool

- Etcd is scheduled on dedicated worker pool
- Other control plane components are deployed separately

Now one set of worker pool can autoscale

# HVPA Controller

- Some components such as Kube apiserver needs both HPA and VPA
- Missing flexibility and Functionality
  - Configurable thresholds
  - Maintenance & Stabilization window
  - Scaling policies
- Reuse HPA and VPA components
- Weight based scaling

```
kind: Hvpa
metadata:
  name: hvpa-sample
spec:
  weightBasedScalingIntervals:
    - vpaWeight: 0
      startReplicaCount: 1
      lastReplicaCount: 3
    - vpaWeight: 0.6
      startReplicaCount: 4
      lastReplicaCount: 10
  hpa:
    .
    .
  template:
    .
    .
    spec:
      minReplicas: 1
      maxReplicas: 10
      metrics:
```

# Resilience / Disaster Recovery – Part I

In case a **shoot** cluster has issues...

- **Kubernetes** (brings back the shoot cluster control plane / resources)
- **Machine Controller** (brings back machines)
- **ETCD Backup & Restore** (brings back the persistence)
- **Gardener** reconciliation (brings back infrastructure, configuration,  
the very essence of what comprises a shoot cluster)

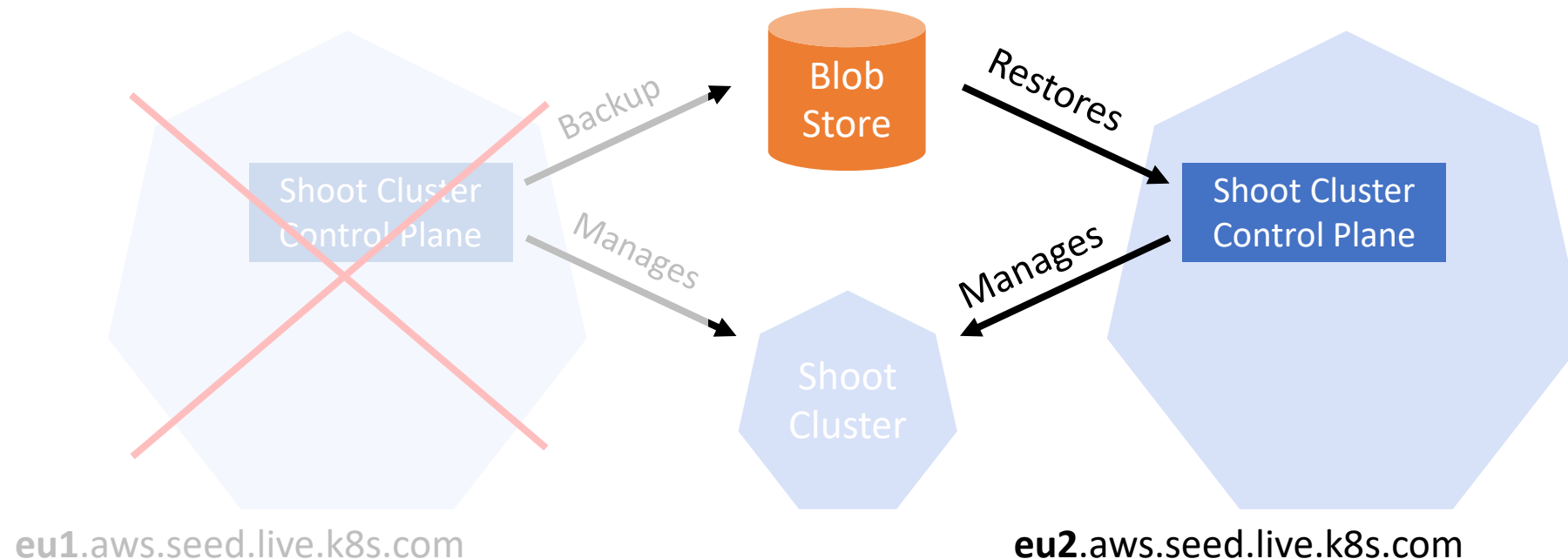
**Note: Workload not included** and must be handled by the end users.



# Resilience / Disaster Recovery – Part II

In case a **seed** cluster is lost...

- Even though a seed cluster is set up as a shoot cluster, regional problems may take it offline longer than we like, so we can **move control planes**



# Demo

The screenshot displays the Gardener Project Clusters interface. On the left is a dark sidebar with the Gardener logo and navigation menu. The main area shows a table of Kubernetes clusters under the heading 'Project Clusters'.

**Gardener**  
1.5.0-alpha.0+dev  
The Kubernetes Botanist

**Project Clusters**

**Kubernetes Clusters** Search

NAME ↑	INFRASTRUCTURE	PURPOSE	STATUS	READINESS	ACTIONS
<a href="#">lwqhpegv1i</a>	eu-de-1	PROD	✓	Control Nodes System	⊞ ⚙️ 🗑️
<a href="#">brdah5ppie</a>	europa-west1	EVAL	✓	Control Nodes System	⊞ ⚙️ 🗑️
<a href="#">js-1</a>	eu-west-1	EVAL	✓	Control Nodes System	⊞ ⚙️ 🗑️

SAP © 2018

# Take Away



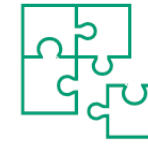
Everywhere



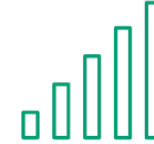
Homogeneous



Fully Managed



Customizable



Scalable



Alibaba  
Cloud



Amazon Web  
Services



Microsoft  
Azure



Google Cloud  
Platform



Metal-  
Stack



OpenStack



Packet Cloud



VMware  
vSphere

<https://gardener.cloud>

<https://github.com/gardener>

<https://kubernetes.slack.com/archives/CB57N0BFG>