Highly Available Kubernetes Clusters

Karan Goel
@karangoel

Meaghan Kjelland
@meaghnk

Google Cloud

High Availability?

Goals

The primary goals of this project are...

- to have a highly-available home Internet setup, with no SPOF (Single Point of Failure)
- to learn and have fun.

ISPs

- CenturyLink Fiber, gigabit, primary ISP.
- Atlas Networks, my backup ISP: a ~100Mbps radio link, using a Unifi NanoBeam AC.

Google Cloud

https://github.com/bradfitz/homelab

Planet Scale

“Designed on the same principles that allows Google to run billions of containers a week, Kubernetes can scale without increasing your ops team.”

- kubernetes.io

Google Cloud

Planet Scale... is meaningless without availability!

“Designed on the same principles that allows Google to run billions of containers a week, Kubernetes can scale without increasing your ops team.”

- kubernetes.io

Google Cloud

Karan Goel

Software Engineer, GKE On-Prem

@karangoel

Meaghan Kjelland

Software Engineer, GKE On-Prem

@meaghnk

Google Cloud

High Availability?

Goals

The primary goals of this project are...

- to have a highly-available home Internet setup, with no SPOF (Single Point of Failure)
- to learn and have fun.

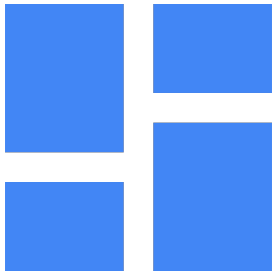
ISPs

- CenturyLink Fiber, gigabit, primary ISP.
- Atlas Networks, my backup ISP: a ~100Mbps radio link, using a Unifi NanoBeam AC.

Google Cloud

https://github.com/bradfitz/homelab

High Availability? Or Multi-Master?



Multi-Master is not enough

Eliminating every single point of failure in each layer of the stack

Control plane

Networking

Application

Persistence

Application

Virtual Machines

Physical Machines

Storage

Electric & Power

Hypervisor

Network Partitions

Cooling Systems

GKE Failure Domains - Zones, Regions

Google Cloud

01

Application

webapp

02

Control Plane

controller-manager

scheduler

apiserver

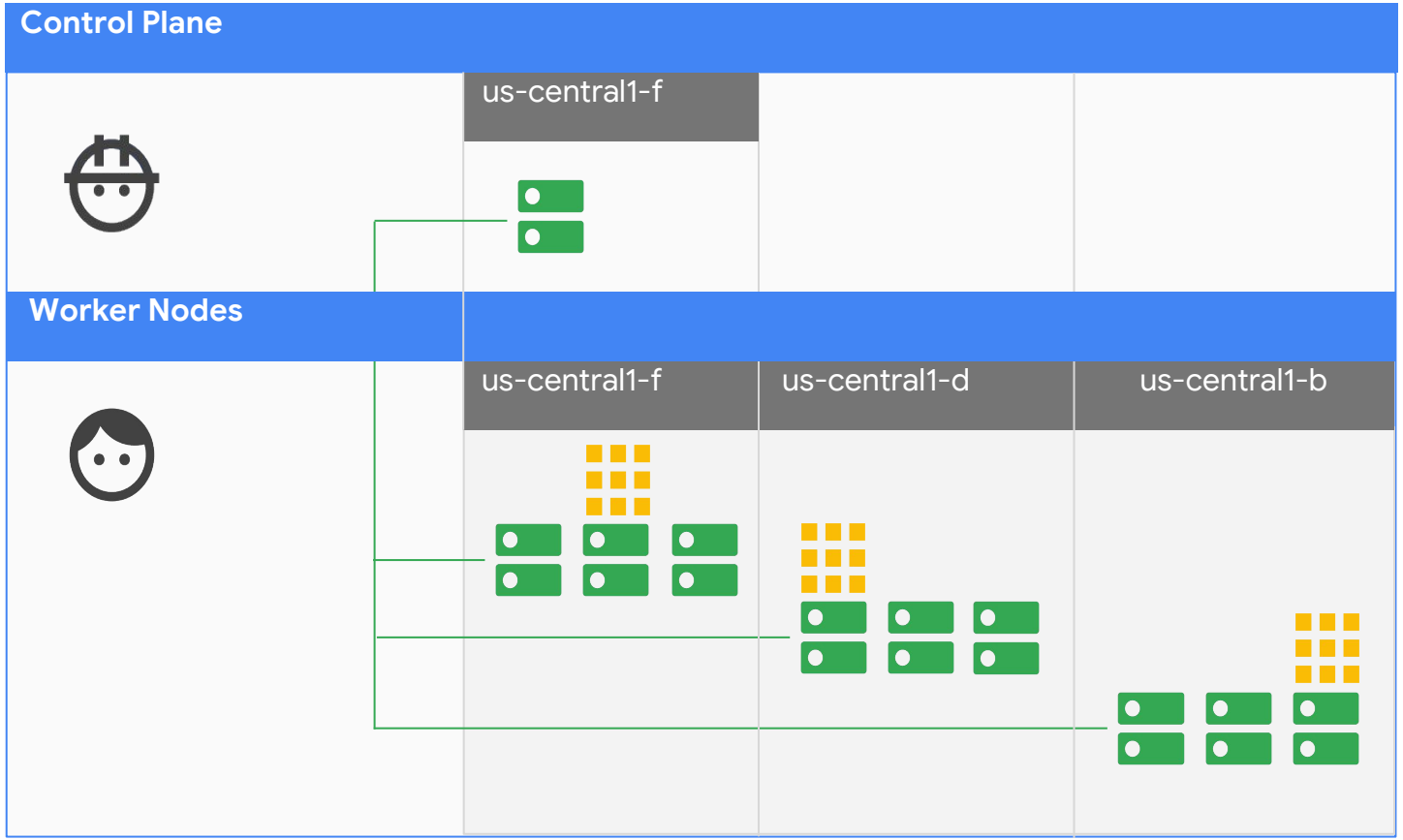
03

Data Plane

etcd

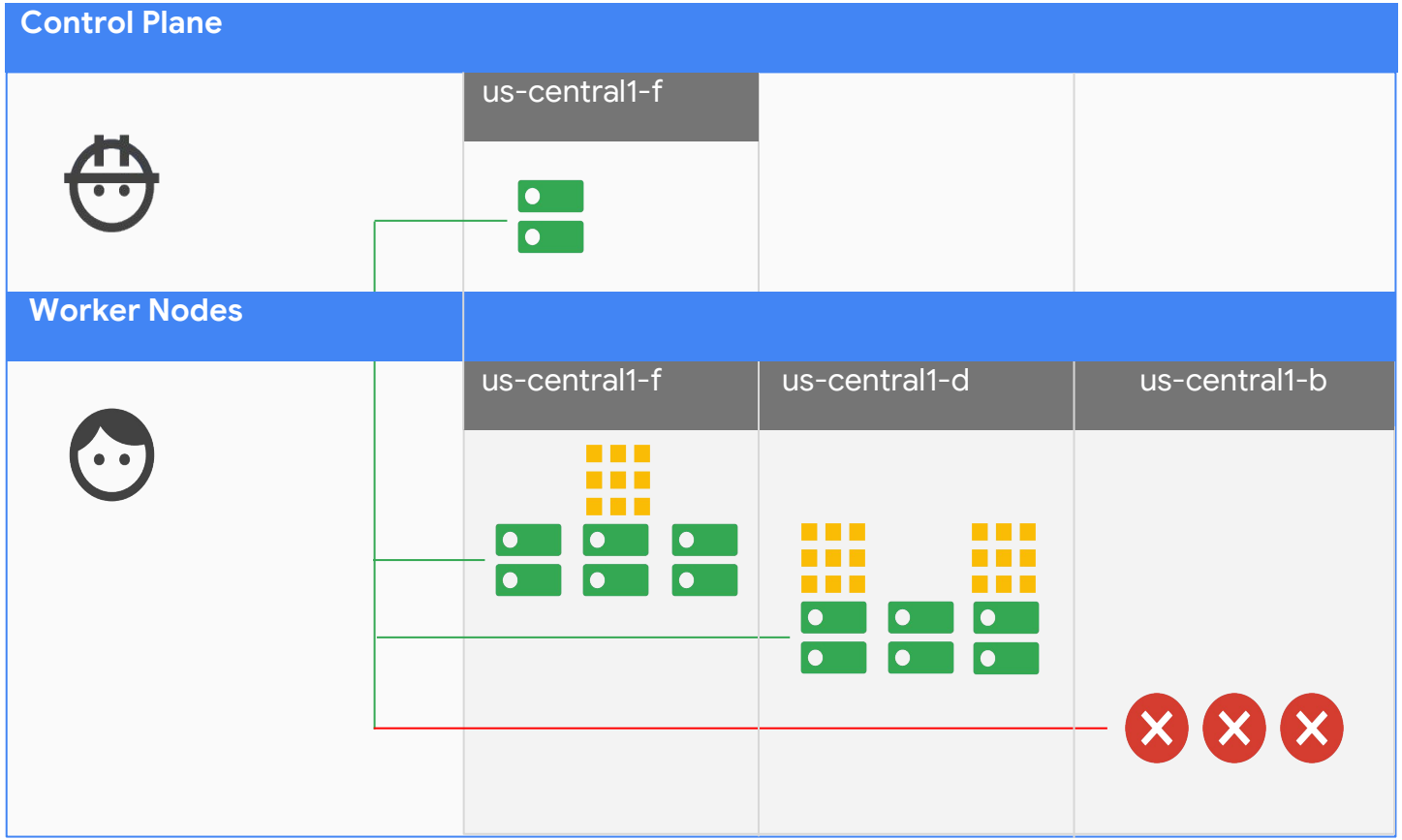
Application HA

Google Cloud



Google Cloud

Legend | Virtual Machine | Container



Google Cloud

Legend | Virtual Machine | Container

Deployments

- Run pods in a Deployment
  - Configure number of **replicas**
  - Rolling **updates**
  - Failure recovery
- StatefulSets for stateful applications
  - Stable storage
  - Unique network identity per pod

```
kind: Deployment
metadata:
 name: component
spec:
  replicas: 3
  updateStrategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 0
  ...
```

Setting Zones

- Pods scheduled across “zones”
- Add label to Nodes
- Added by Cloud Providers

```
kind: Deployment
metadata:
 name: component
spec:
  affinity:
    podAntiAffinity:

preferredDuringSchedulingIgnoredDuringExecu
tion:
      - labelSelector:
        matchExpressions:
        - key: component
          operator: In
          values:
          - component
         topologyKey:

         "failure-domain.beta.kubernetes.io/
         zone"
  ...
```

Google Cloud

Node Upgrades

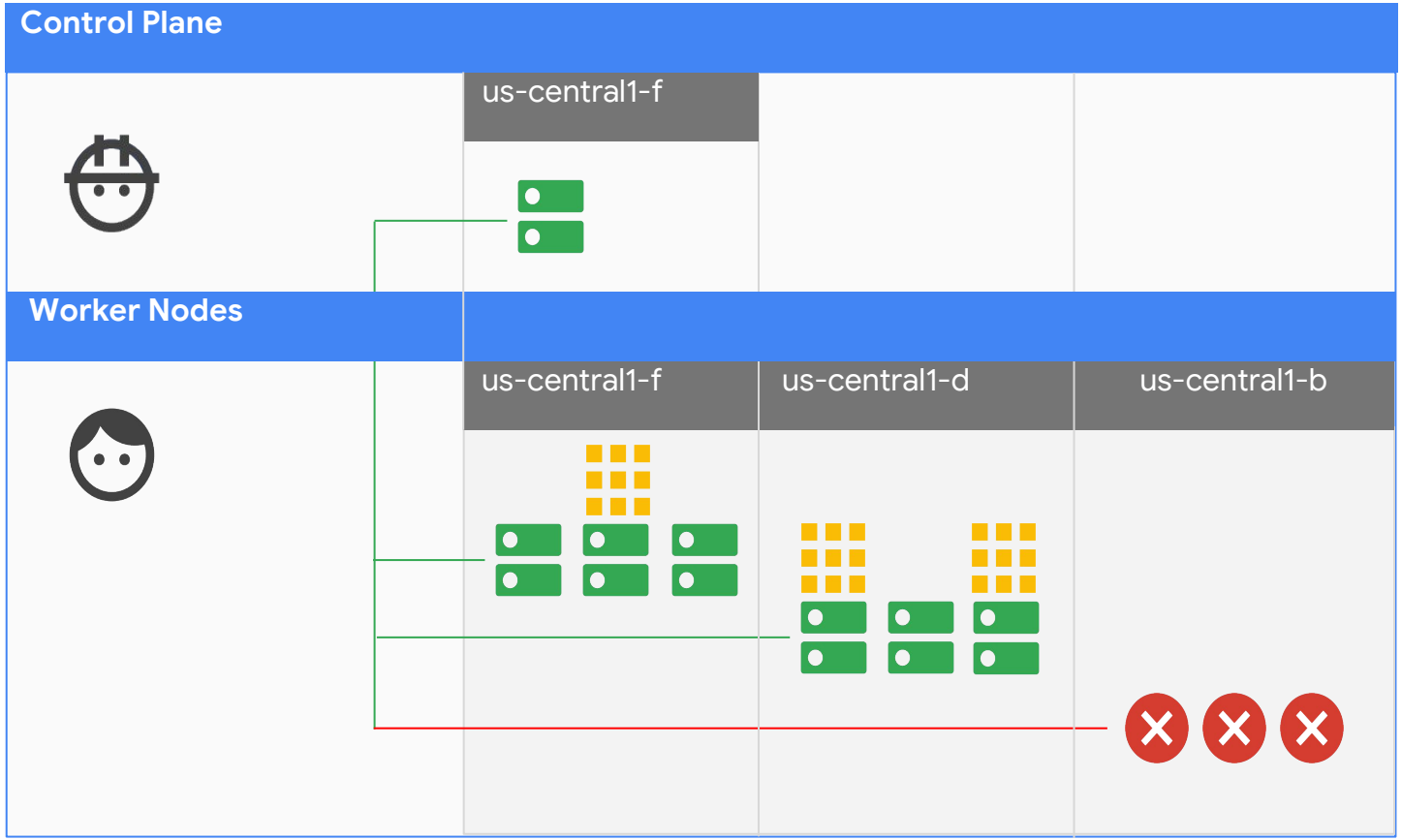
- Voluntary disruptions
- PodDisruptionBudget + kubectl drain
- Pod Eviction API rejects calls

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: my-app-pdb
spec:
  minAvailable: 2
  selector:
    matchLabels:
      app: my-app
```

Google Cloud

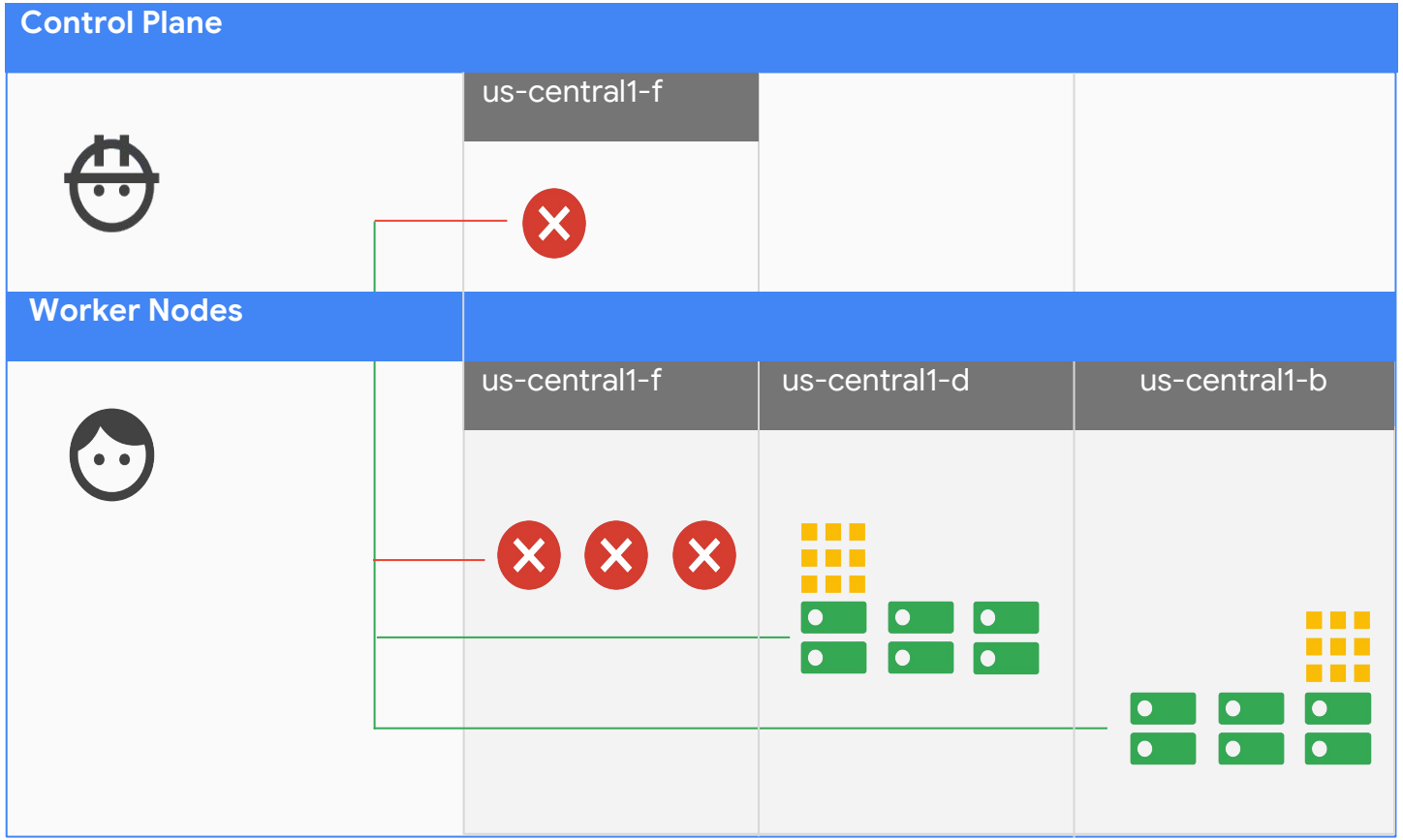
Control Plane HA

Google Cloud



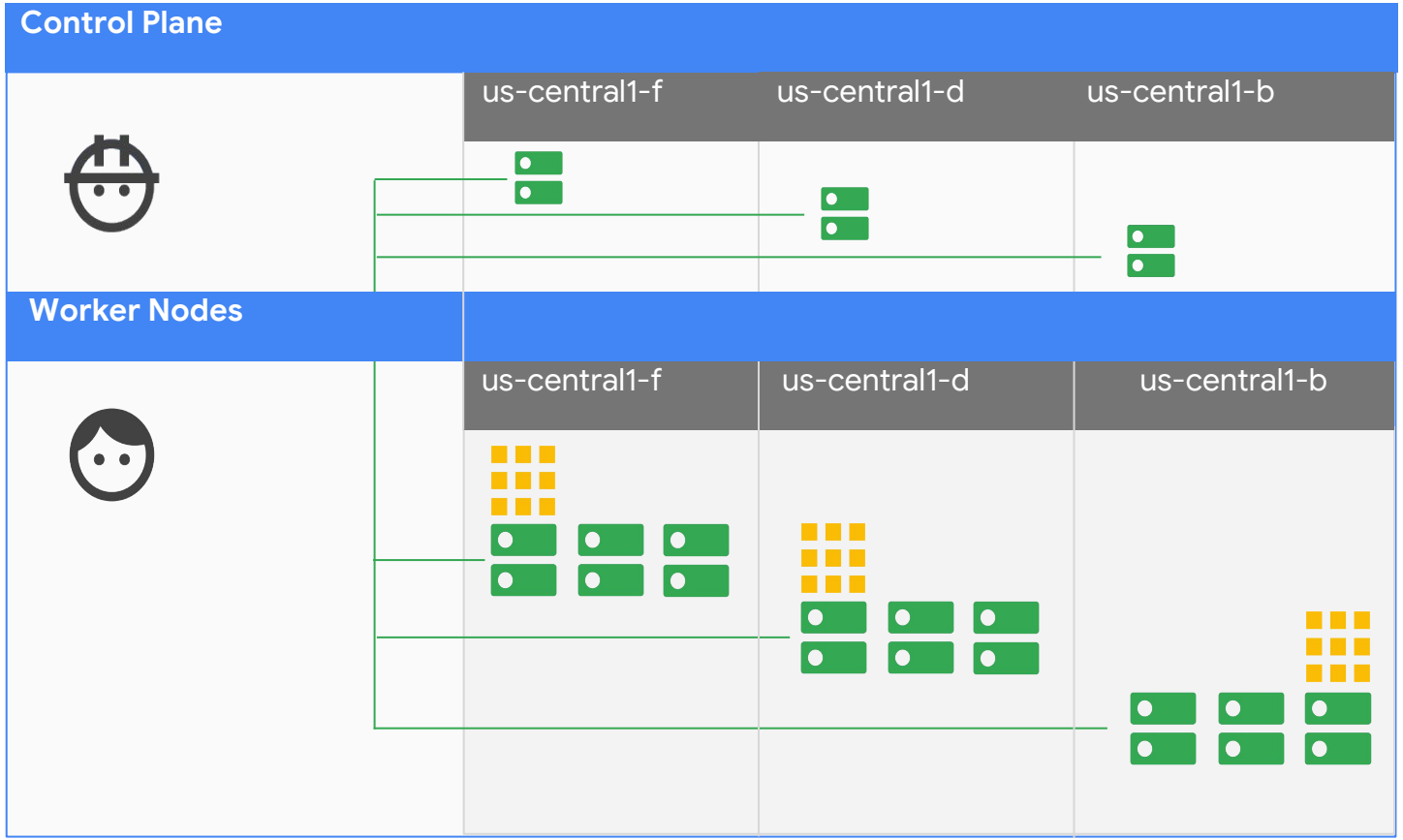
Google Cloud

Legend | Virtual Machine | Container



Google Cloud

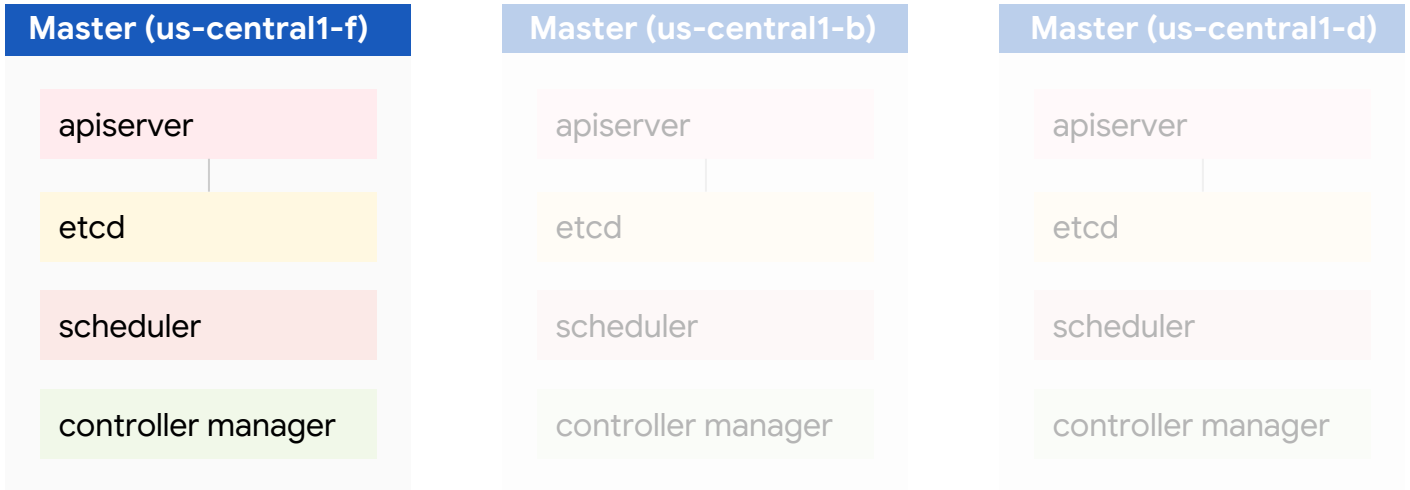
Legend | Virtual Machine | Container



Google Cloud

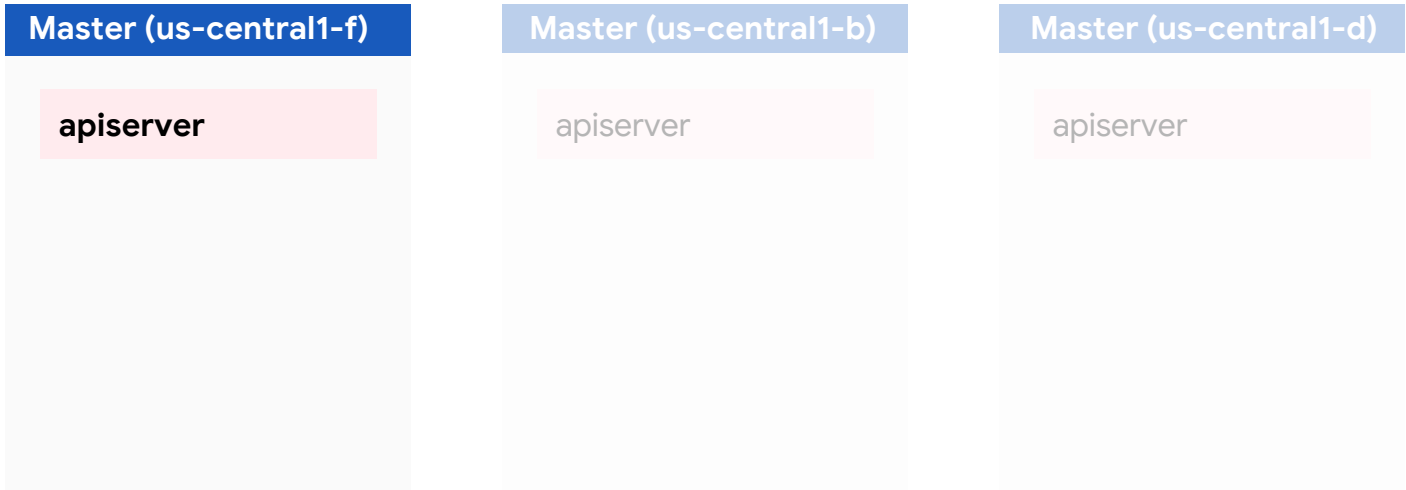
Legend | Virtual Machine | Container

Control Plane High Availability



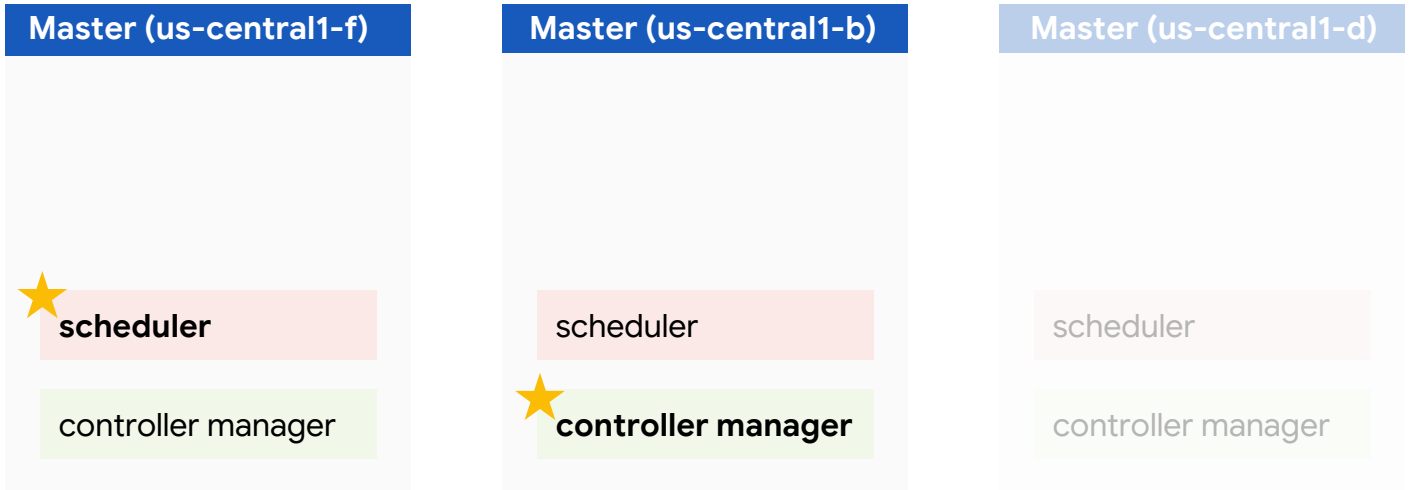
Google Cloud

Kubernetes Active-Active Components



Google Cloud

Kubernetes Active-Passive Components



Google Cloud

Configuring Leader Election

`--leader-elect`

`--leader-elect-lease-duration`

`--leader-elect-renew-deadline`

`--leader-elect-resource-lock`

`--leader-elect-retry-period`

Managing the Control Plane

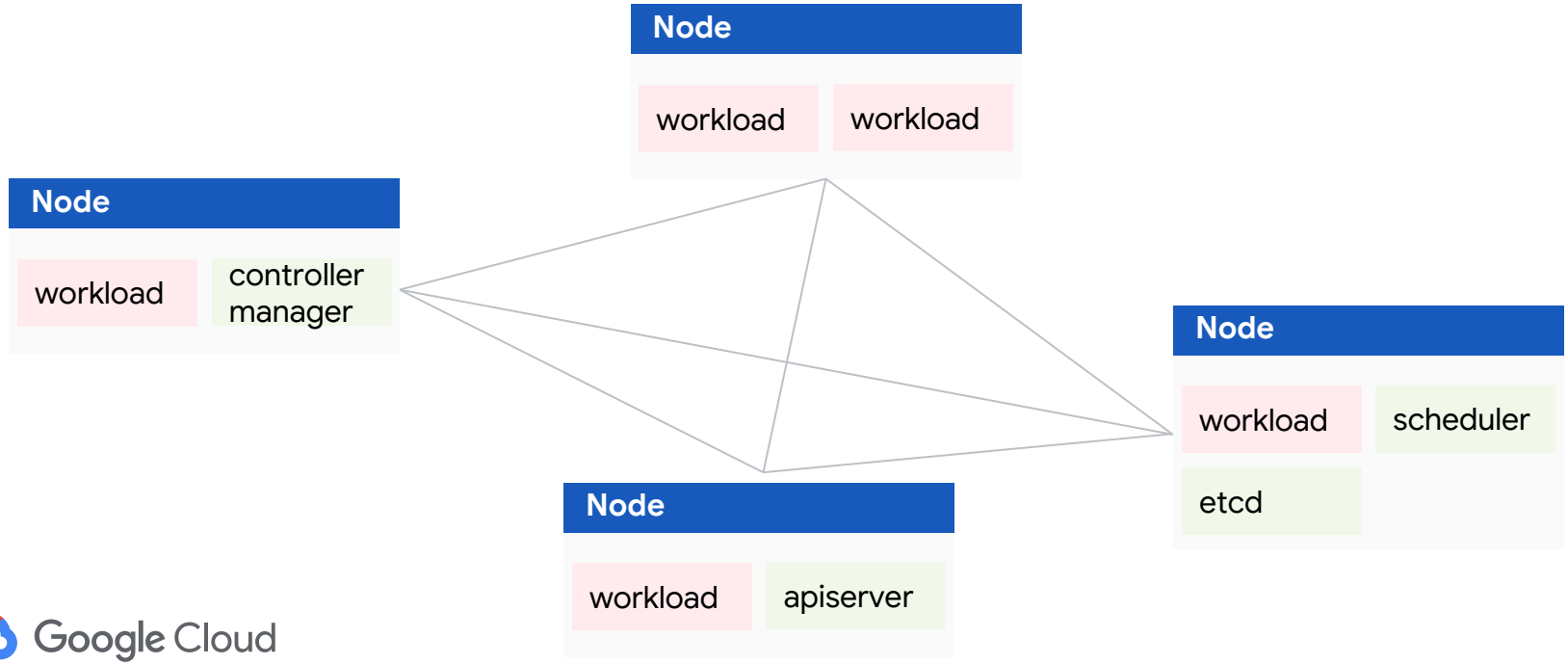
Unsolved problems:

- Health checking
- Failure recovery
- Upgrades without downtime

Options to explore:

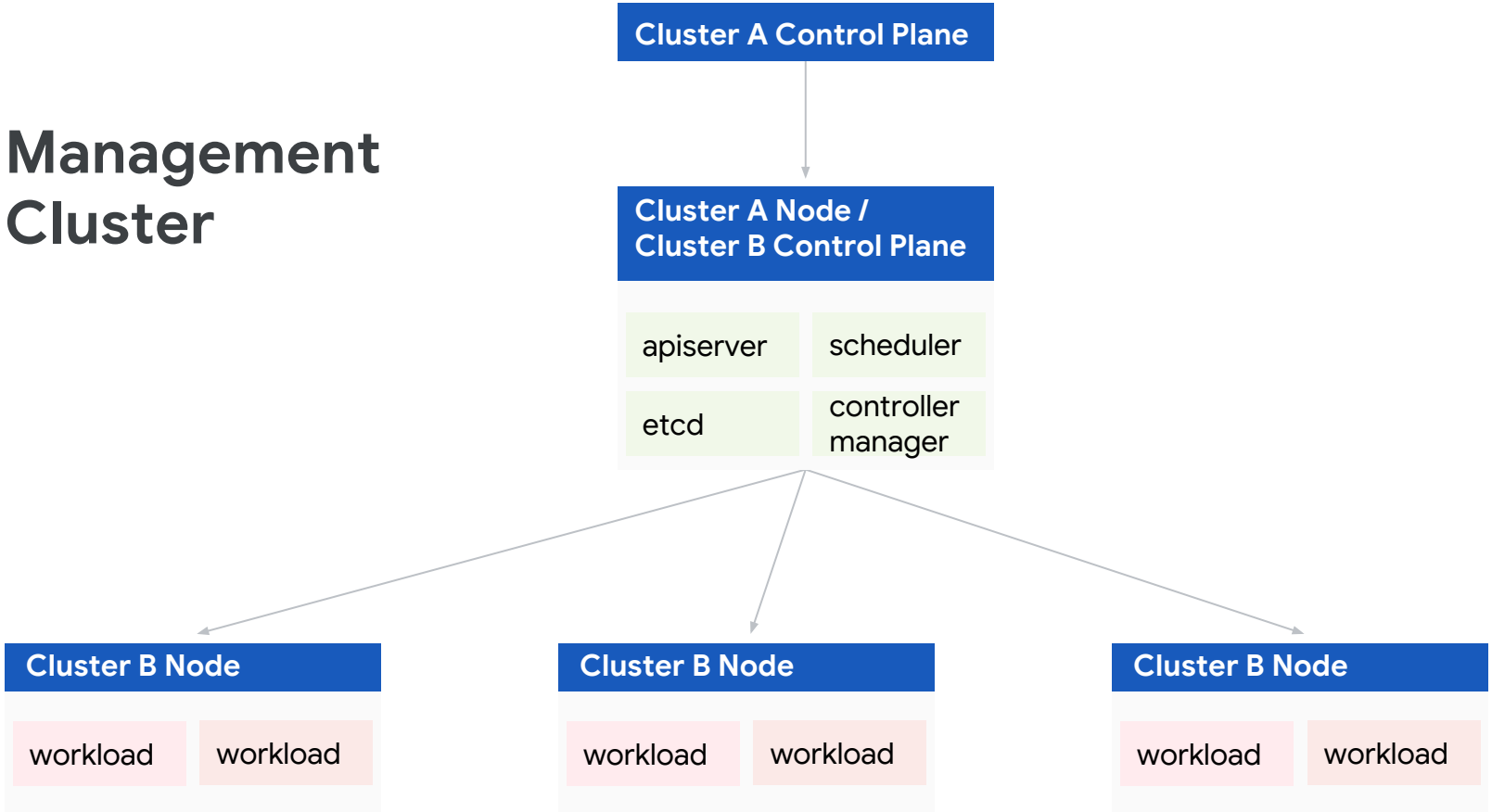
- Hosted solution
- Managed instance groups
- Build your own monitoring server (bash/golang server)
- Kubernetes itself!

Self-Hosted Kubernetes



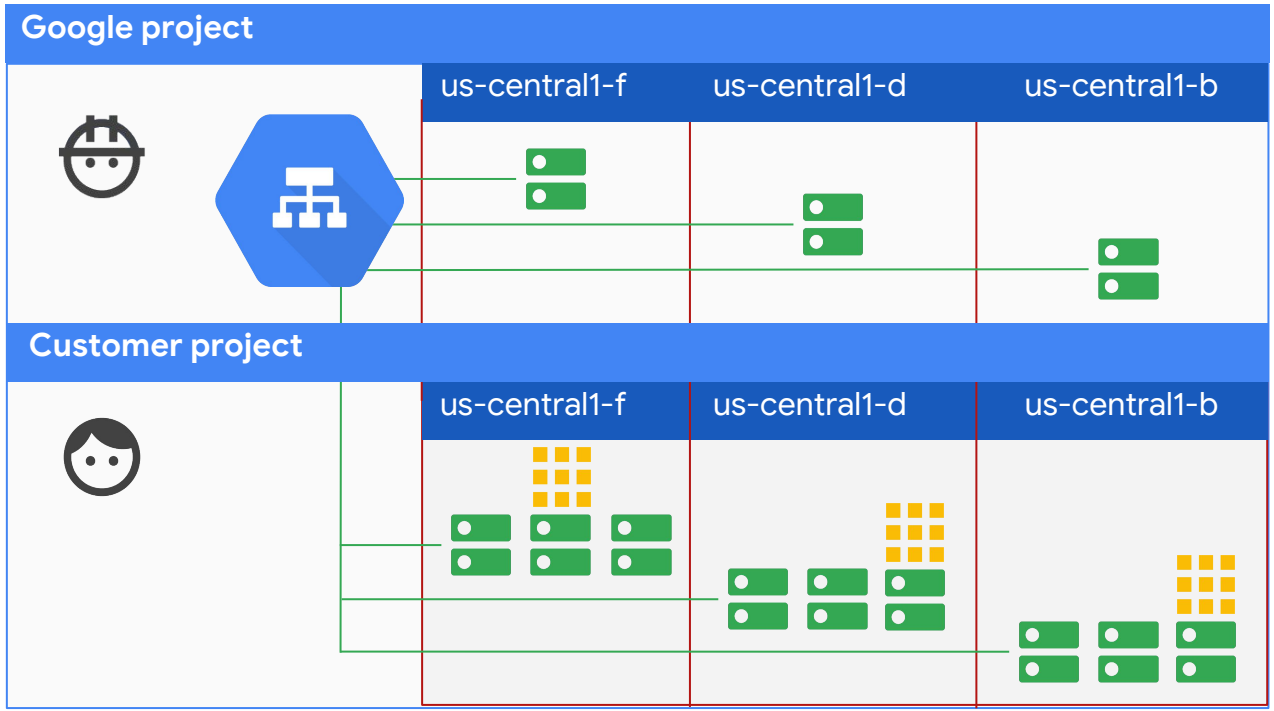
Google Cloud

Management Cluster



Google Cloud

GKE's Solution



Google Cloud

Legend | Virtual Machine | Container | Load Balancer

cluster-api

- Kubernetes-style APIs for Machines and Cluster
- Sig-cluster-lifecycle
  - Cluster API working group

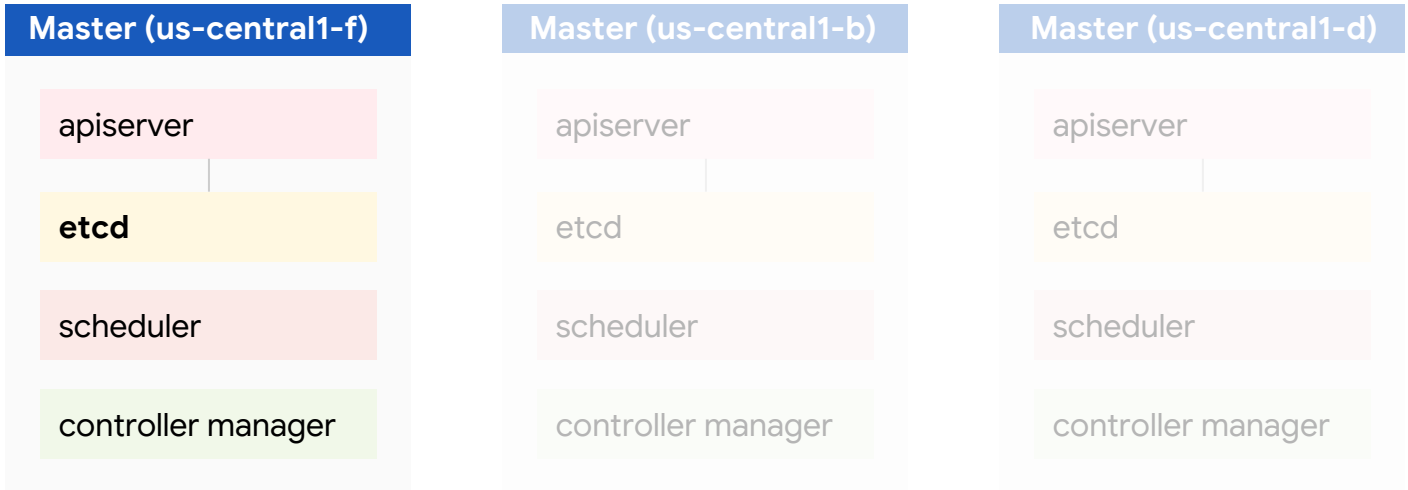
```
apiVersion: cluster.k8s.io/v1alpha1
kind: Cluster
metadata:
   name: cluster-example
spec:
...
---
apiVersion: cluster.k8s.io/v1alpha1
Kind: Machine
metadata:
   name: machine-example
spec:
...
```

Google Cloud

Data Plane - etcd

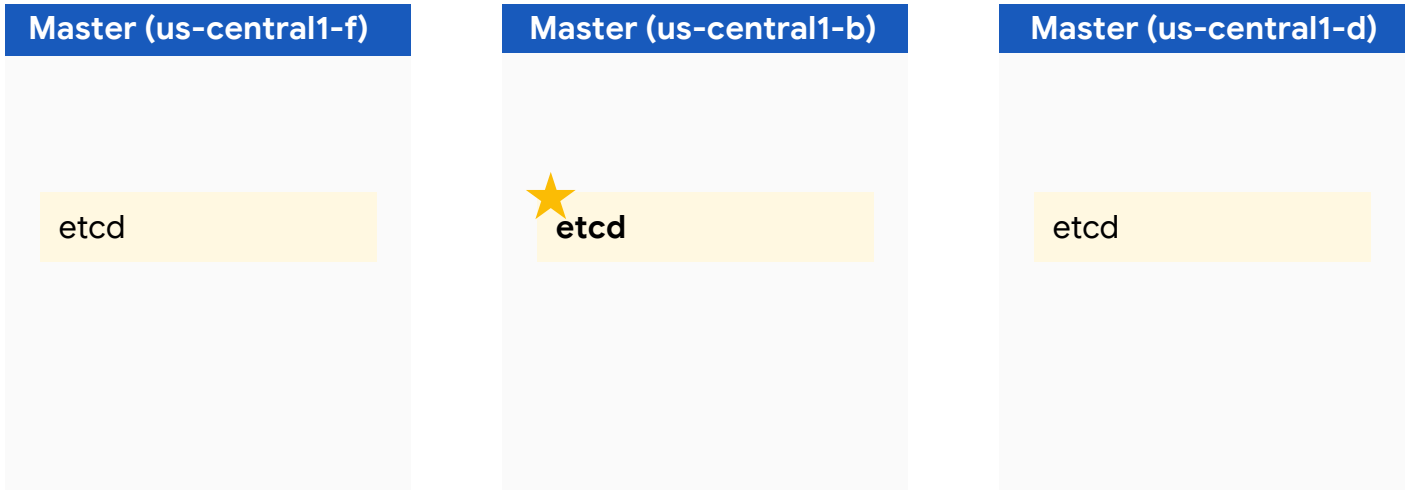
Google Cloud

Control Plane High Availability



Google Cloud

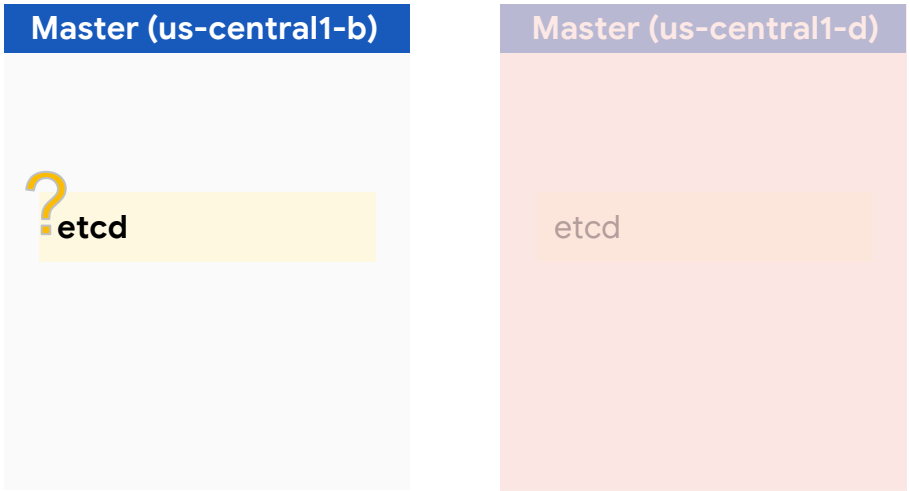
Quorum with etcd



Google Cloud

N / 2 + 1

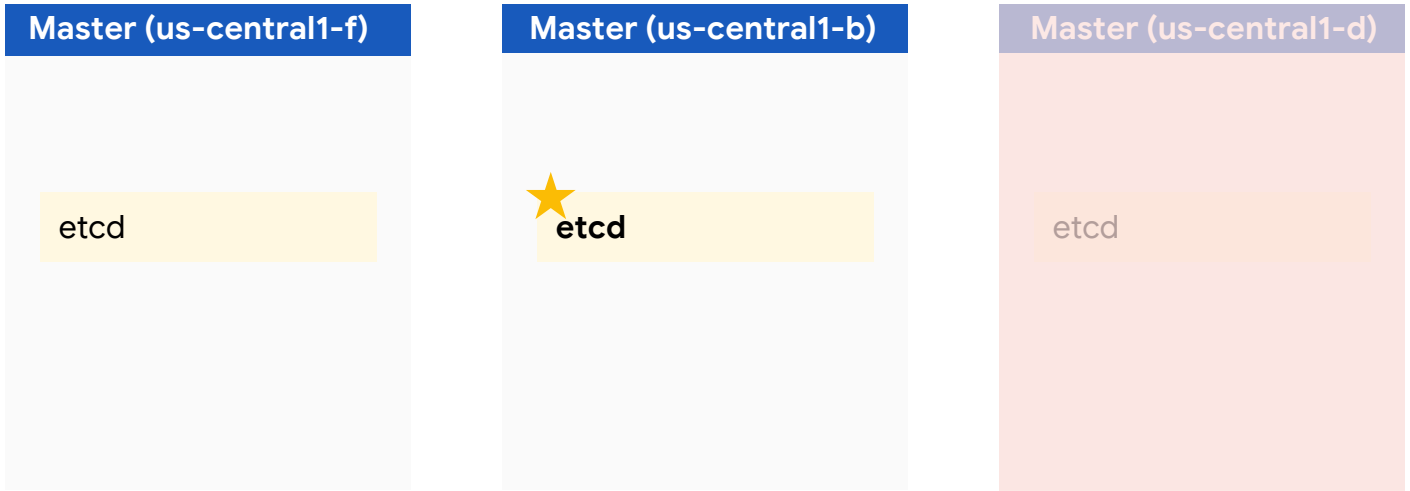
Quorum with etcd



N / 2 + 1

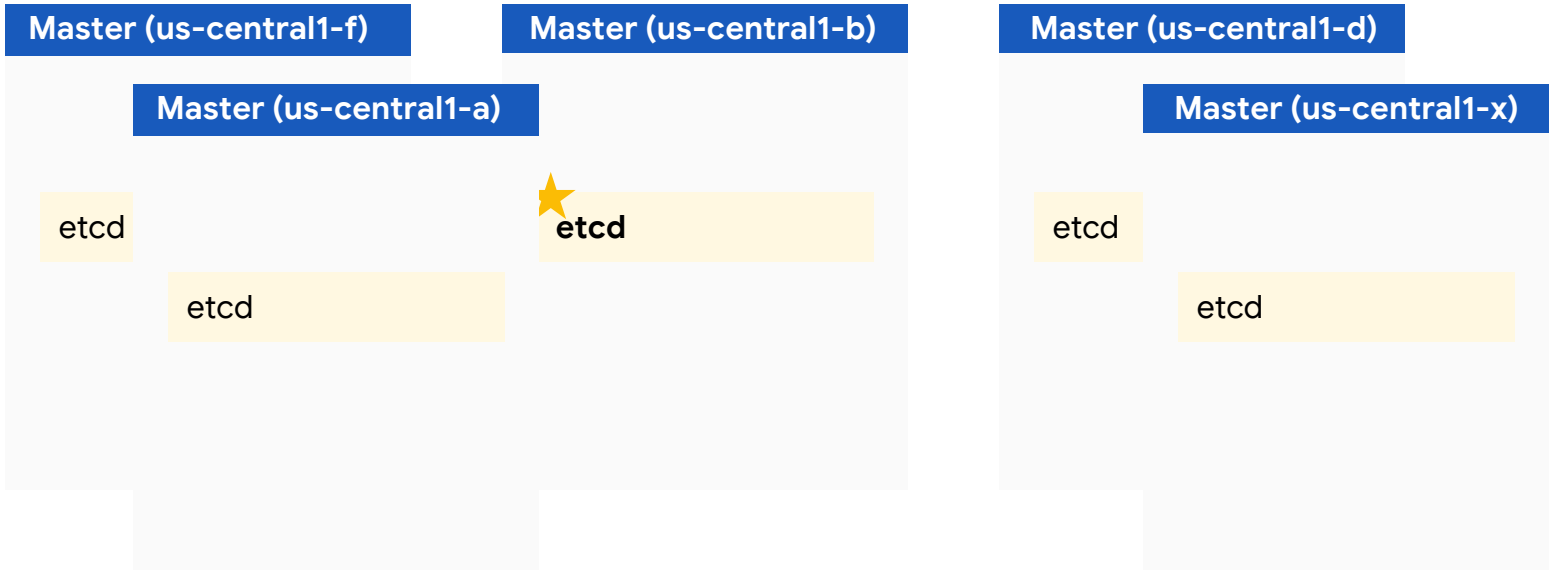
CLUSTER SIZE	MAJORITY	FAILURE TOLERANCE
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

Quorum with etcd



Google Cloud

HA during upgrade



Google Cloud

Configure

- StatefulSet
  - Example

```
# 1. Connect clients
./etcd
--listen-client-urls=http://$IP1:2379,
http://$IP2:2379, http://$IP3:2379,
http://$IP4:2379, http://$IP5:2379
--advertise-client-urls=http://$IP1:2379,
http://$IP2:2379, http://$IP3:2379,
http://$IP4:2379, http://$IP5:2379

# 2. Start API server
./kube-apiserver --etcd-servers=$IP1:2379,
$IP2:2379, $IP3:2379, $IP4:2379, $IP5:2379
```

Backup and Restore

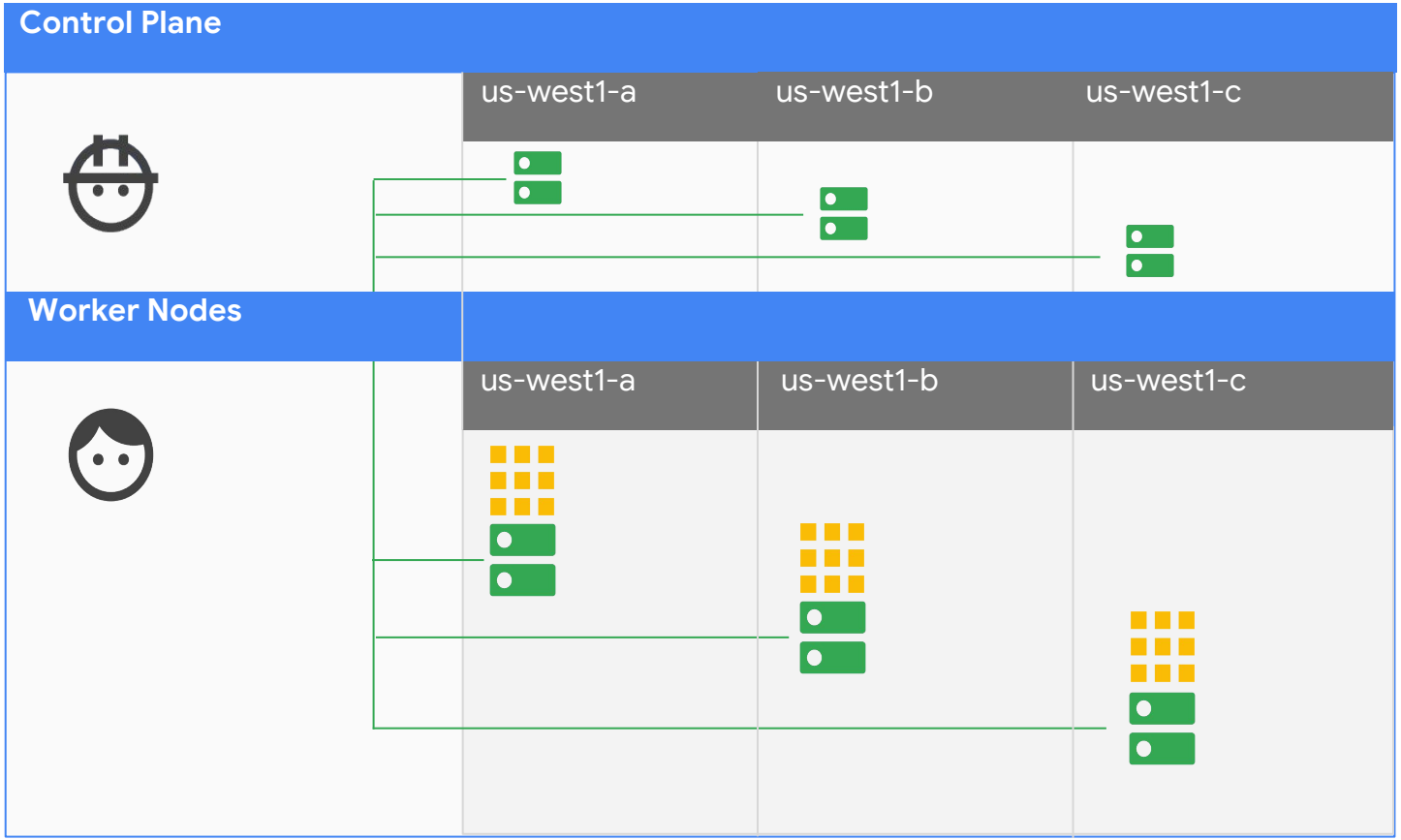
- Periodic backups
- Monitoring that backups happened
- Alerting when backups are too stale
- Automatically test restore function
- https://coreos.com/etcd/docs/latest/op-guide/recovery.html

Backup and Restore

- Ark
  - What if apiserver is down?
- etcd Operator
- etcdctl snapshot + cron

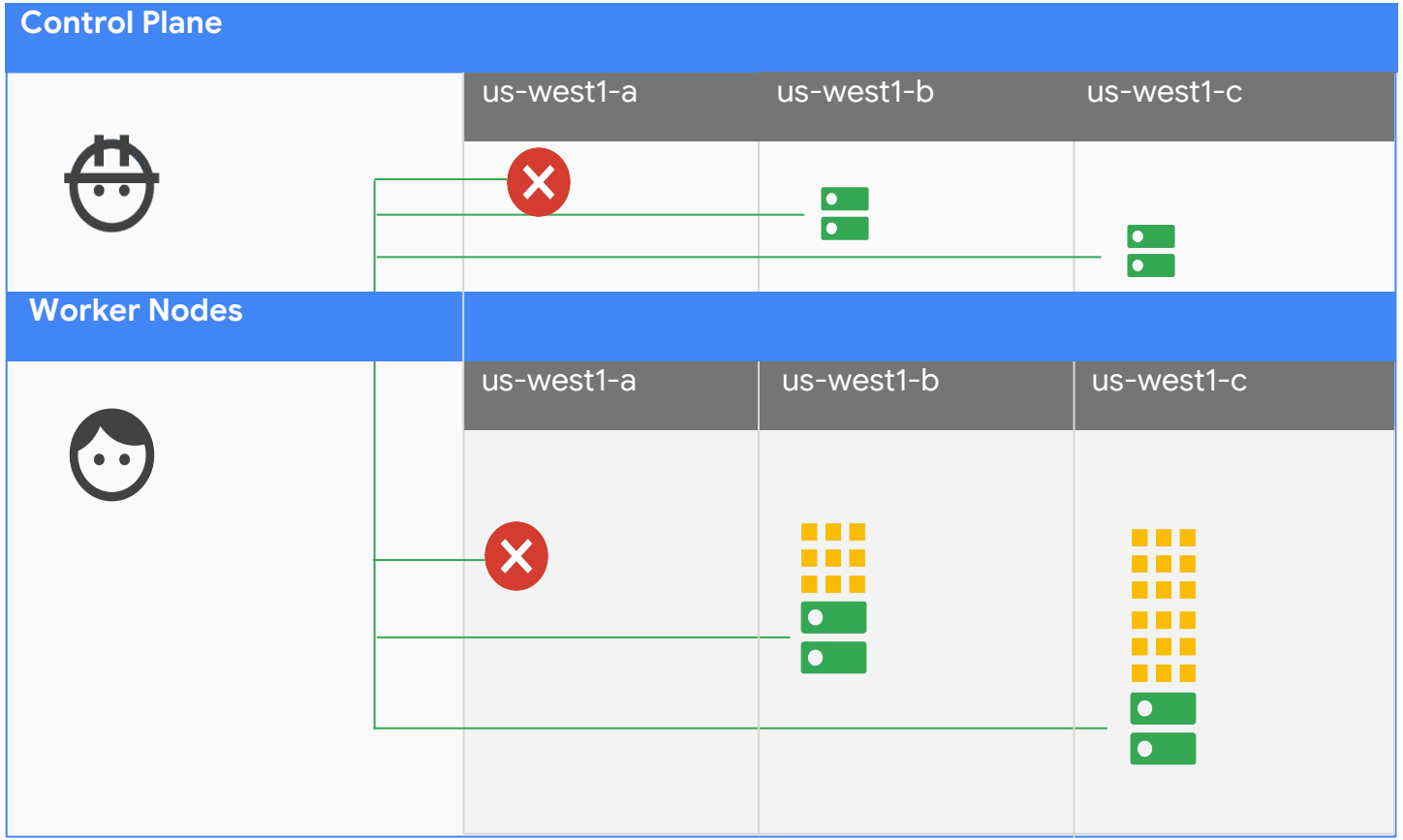
DEMO TIME!

1. Deploy an application on cross-zone, multi-master cluster
2. Balance application across GCE zones
3. Simulate zonal failure
4. See what happens!



Google Cloud

Legend | Virtual Machine | Container



Google Cloud

Legend | Virtual Machine | Container

Thank You!

@karangoel
@meaghnk

Additional Resources

https://cloud.google.com/kubernetes-engine/docs/concepts/regional-clusters

https://kubernetes.io/docs/setup/independent/high-availability/

https://github.com/kelseyhightower/kubernetes-the-hard-way

https://github.com/kubernetes-sigs/cluster-api

Google Cloud