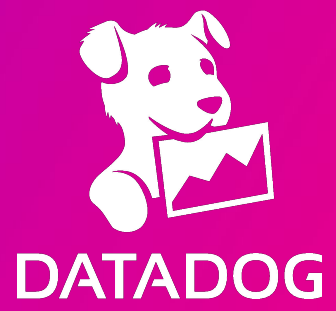
KubeCon + CloudNativeCon North America 2023

How to Carefully Replace Thousands of Nodes Every Day



DATADOG

Adrien Trouillaud

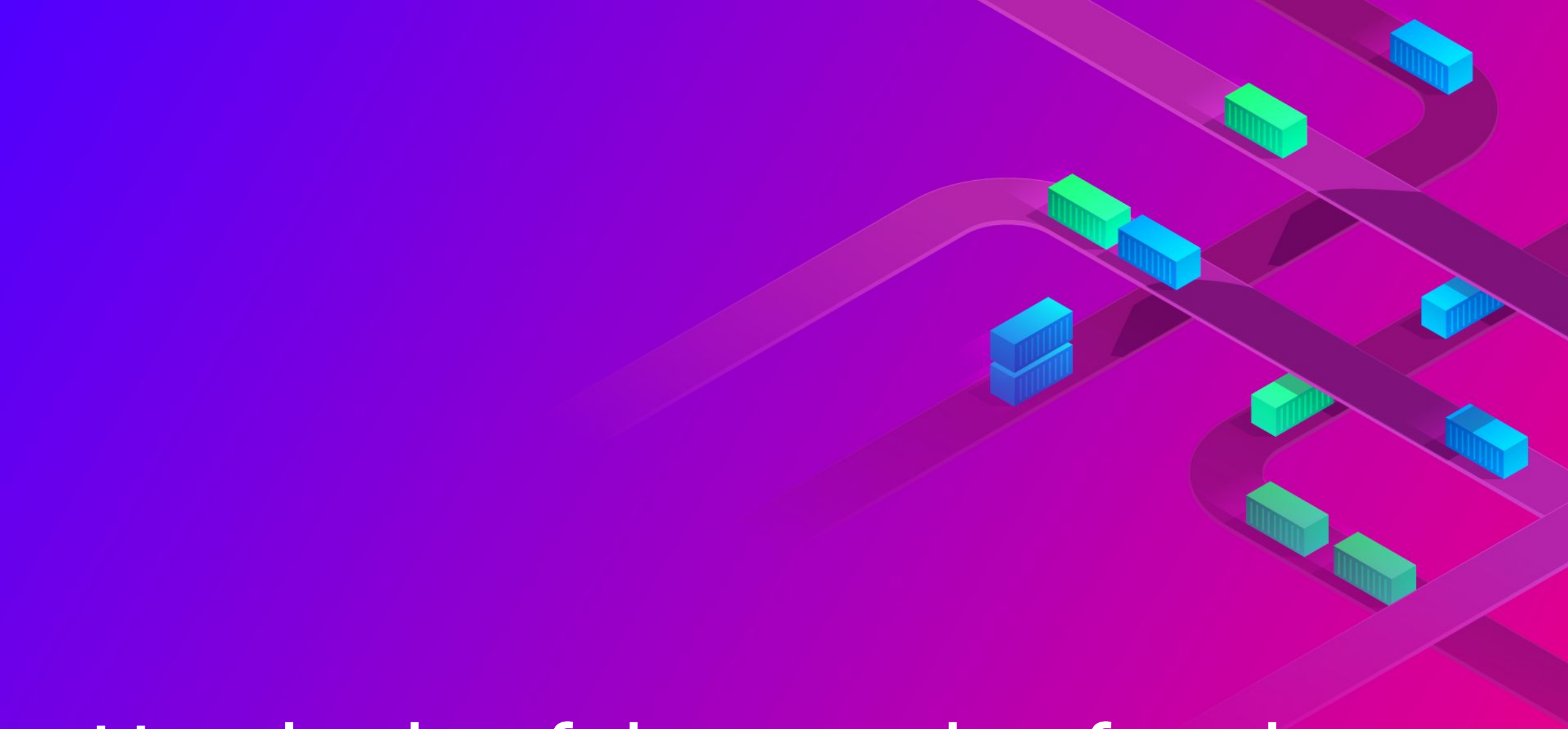
Engineering Manager

Ryan McNamara

Senior Software Engineer

DATADOG

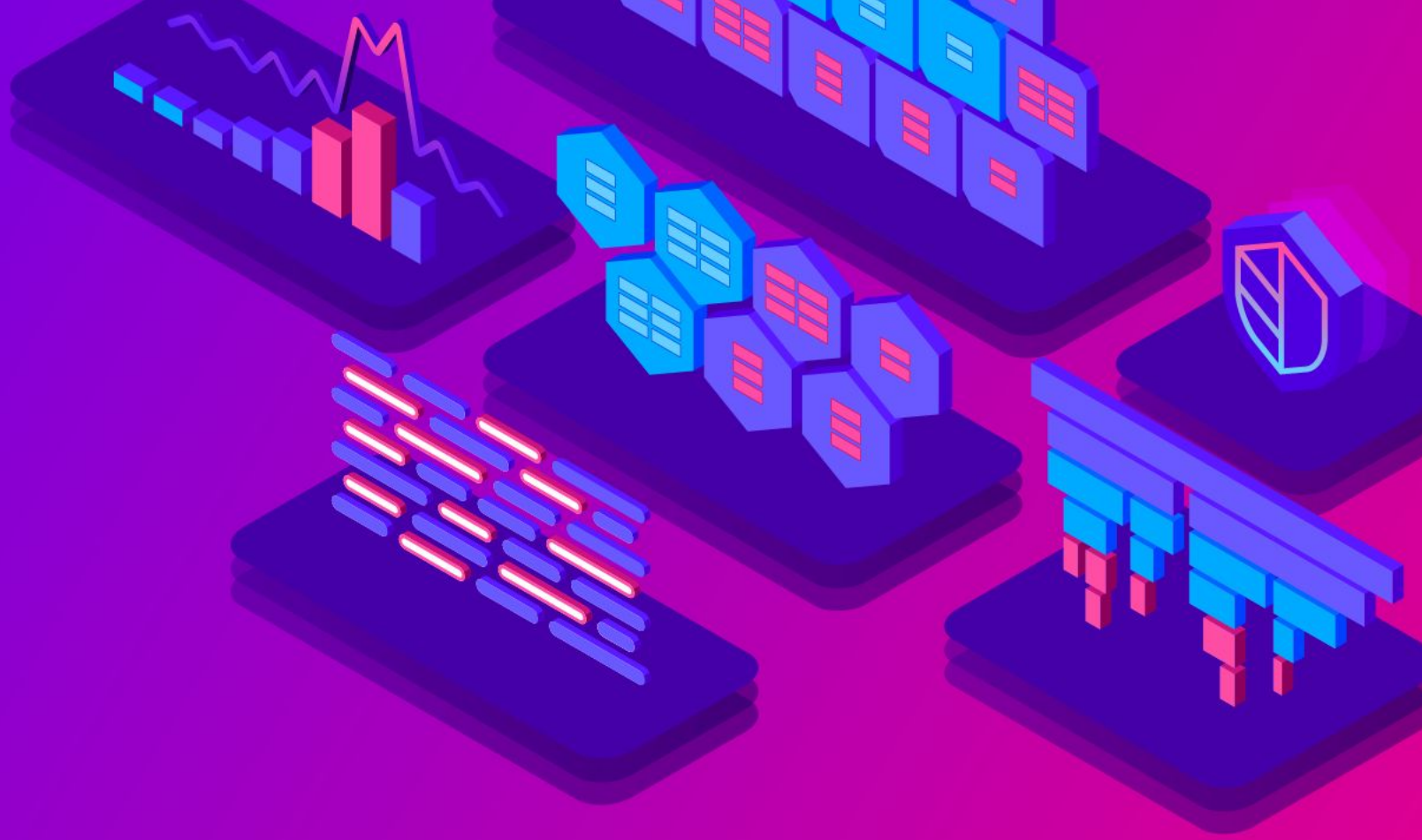
Trillions of data points per hour



Hundreds of thousands of pods

Tens of thousands of nodes

Multiple clouds



Kubernetes from scratch

Agenda

01 Why and how to replace nodes

02 The case for default Pod Disruption Budgets (PDBs)

03 Readiness probe overload

04 The missing node lifecycle hooks

05 Recent, ongoing, and possible changes to Evictions and PDBs

Reasons to replace nodes

Example solutions

EKS managed node group updates

AKS and GKE node pool upgrades, node auto-repair (Node Problem Detector)

Karpenter deprovisioning methods (Drift, Interruption)

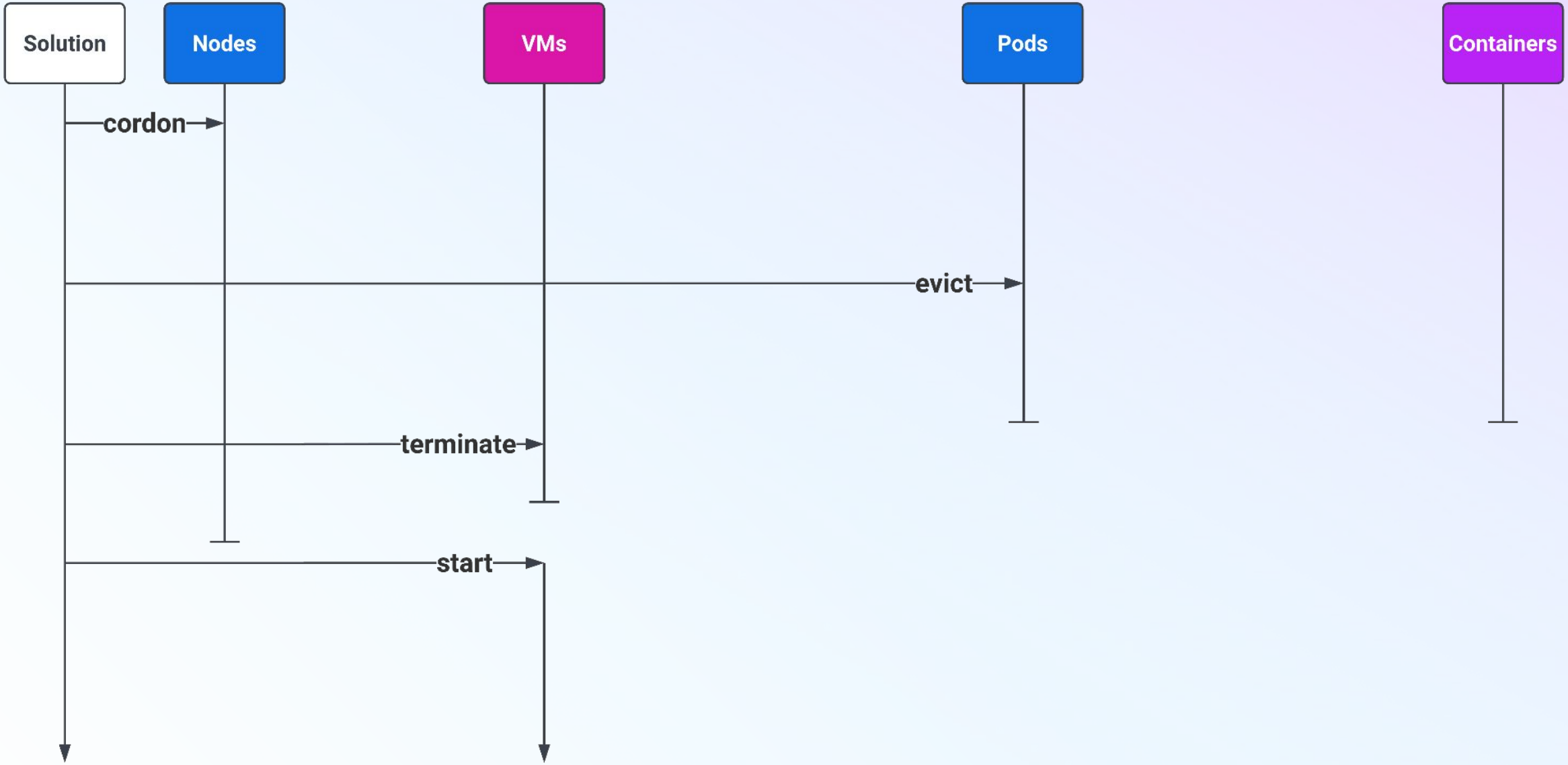
All rely on the Eviction API.

Pod Disruption Budget (PDB) and Readiness Probe

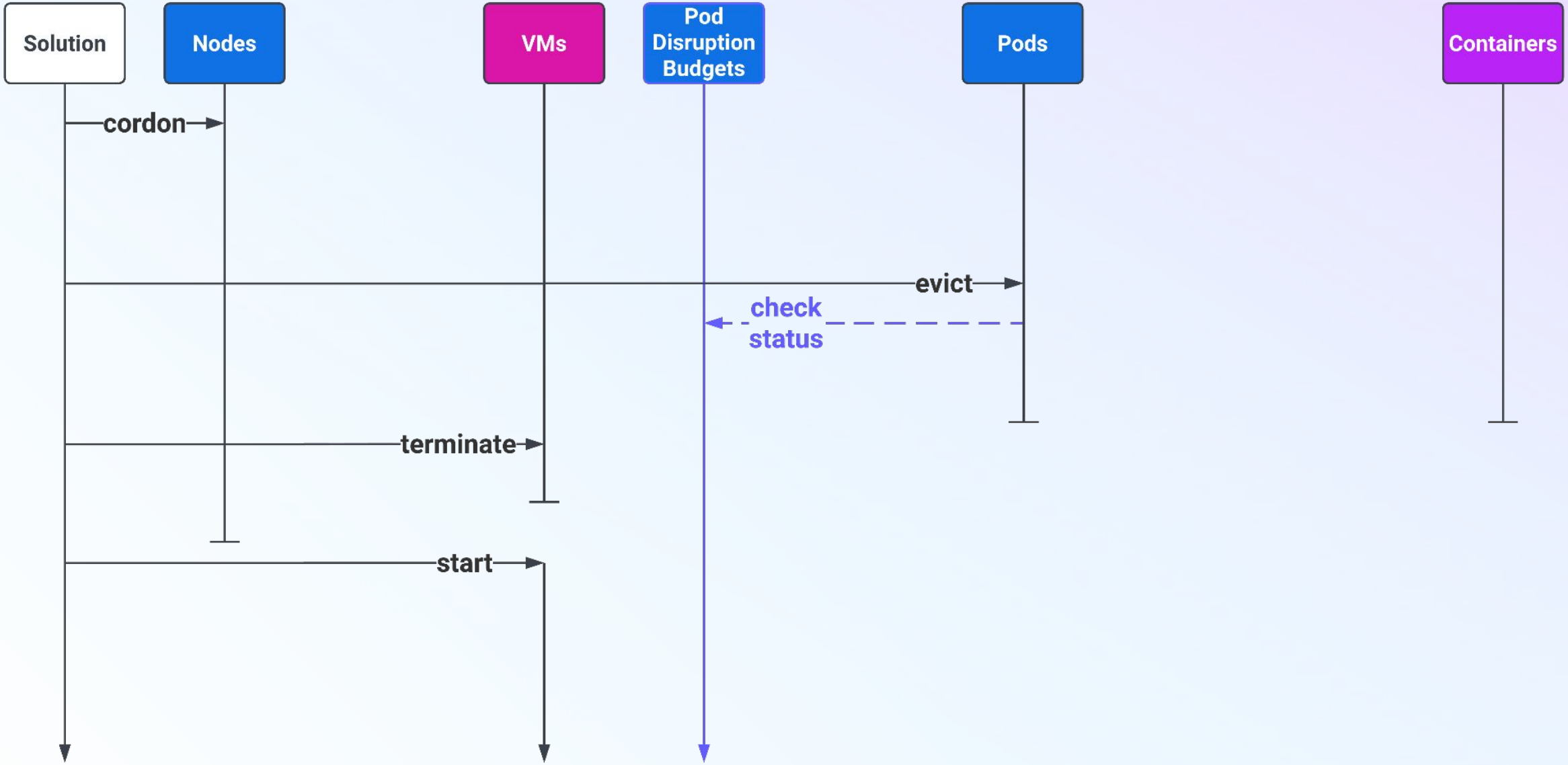
```yaml
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  # ...
spec:
  selector:
    matchLabels:
      app: my-app
  maxUnavailable: 1
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  # ...
spec:
  containers:
  - ports:
    - containerPort: 8080
    readinessProbe:
      httpGet:
        path: /healthz
        port: 8080
```

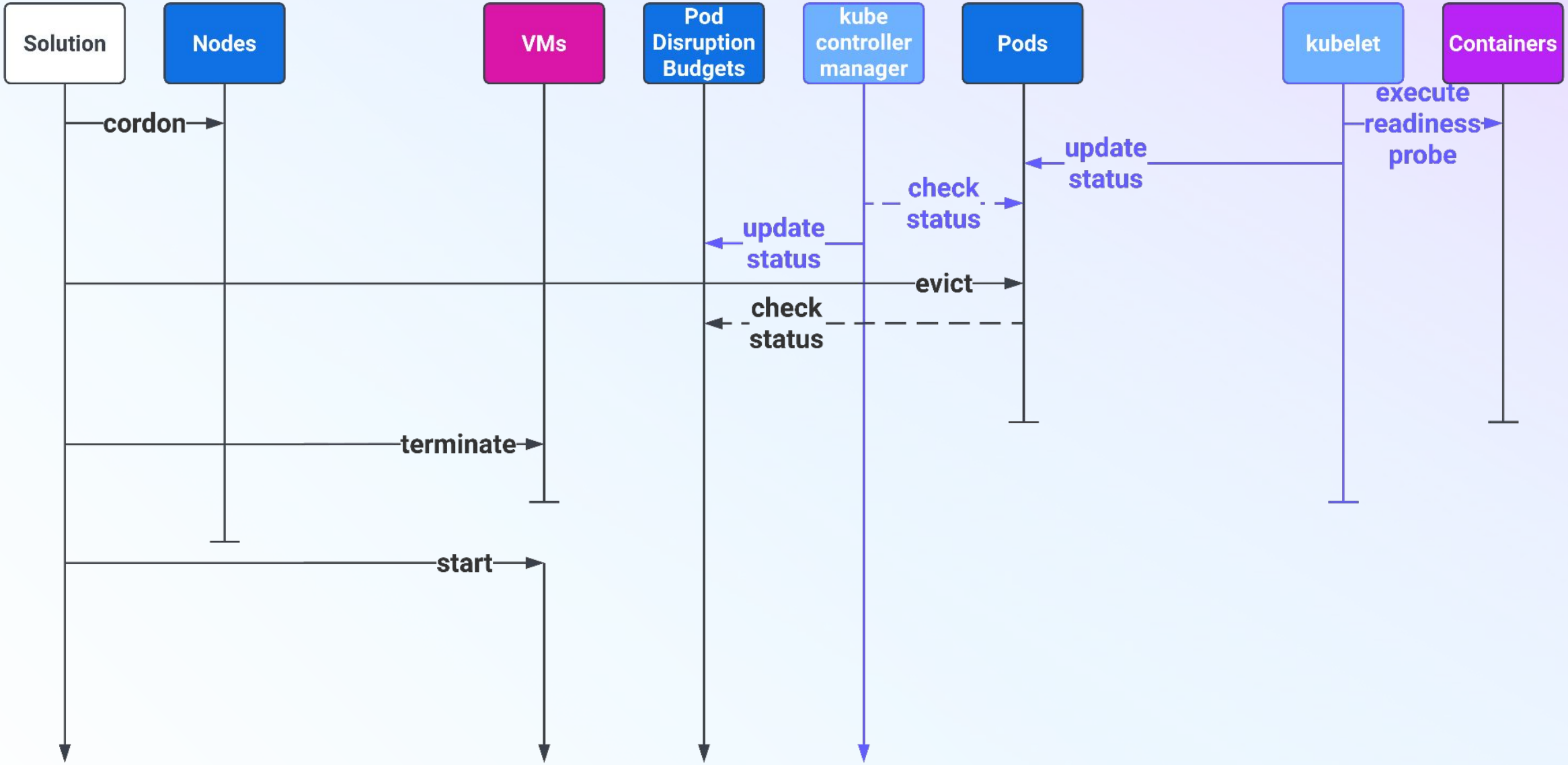
How to replace a node



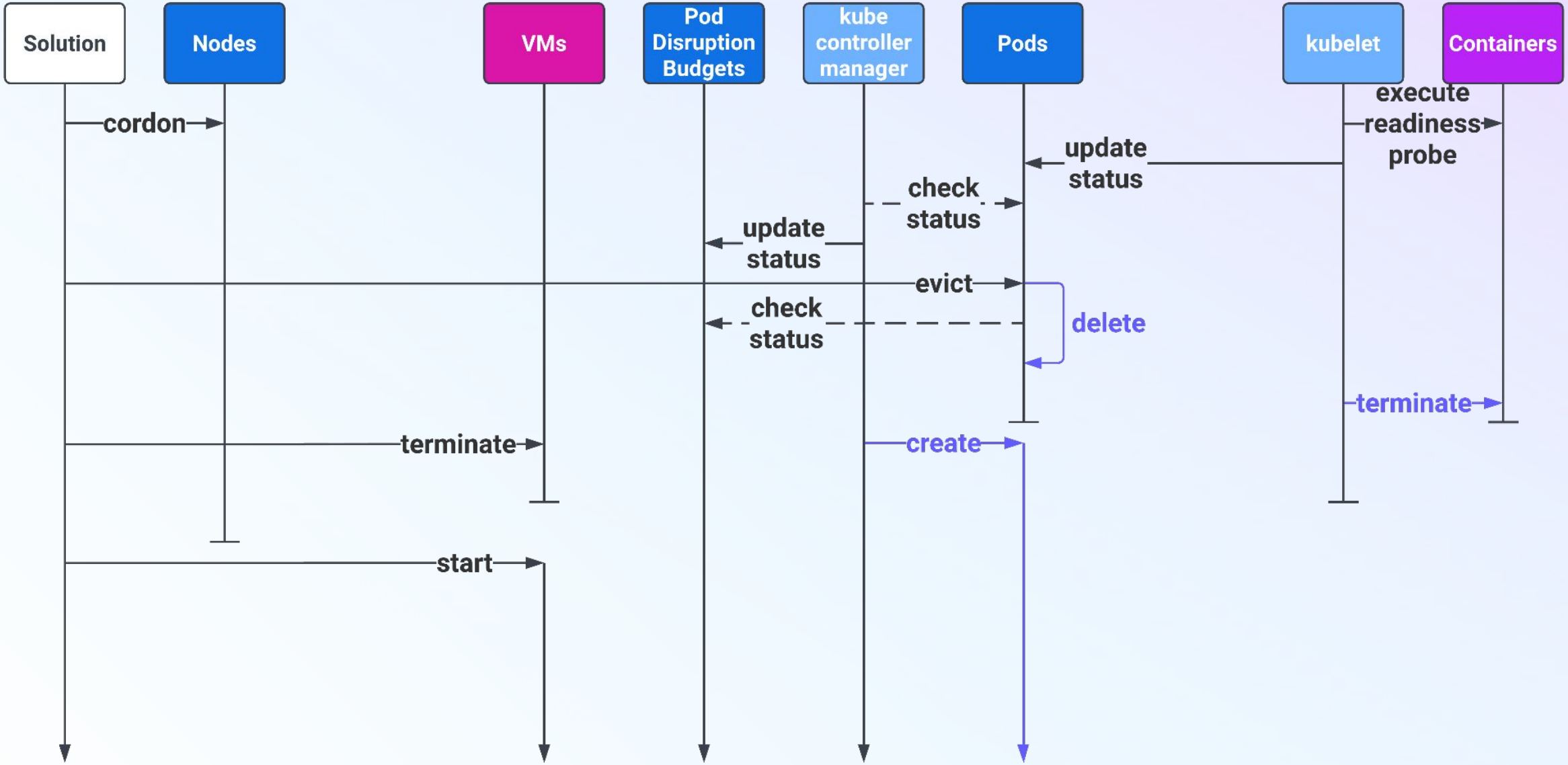
How to replace a node



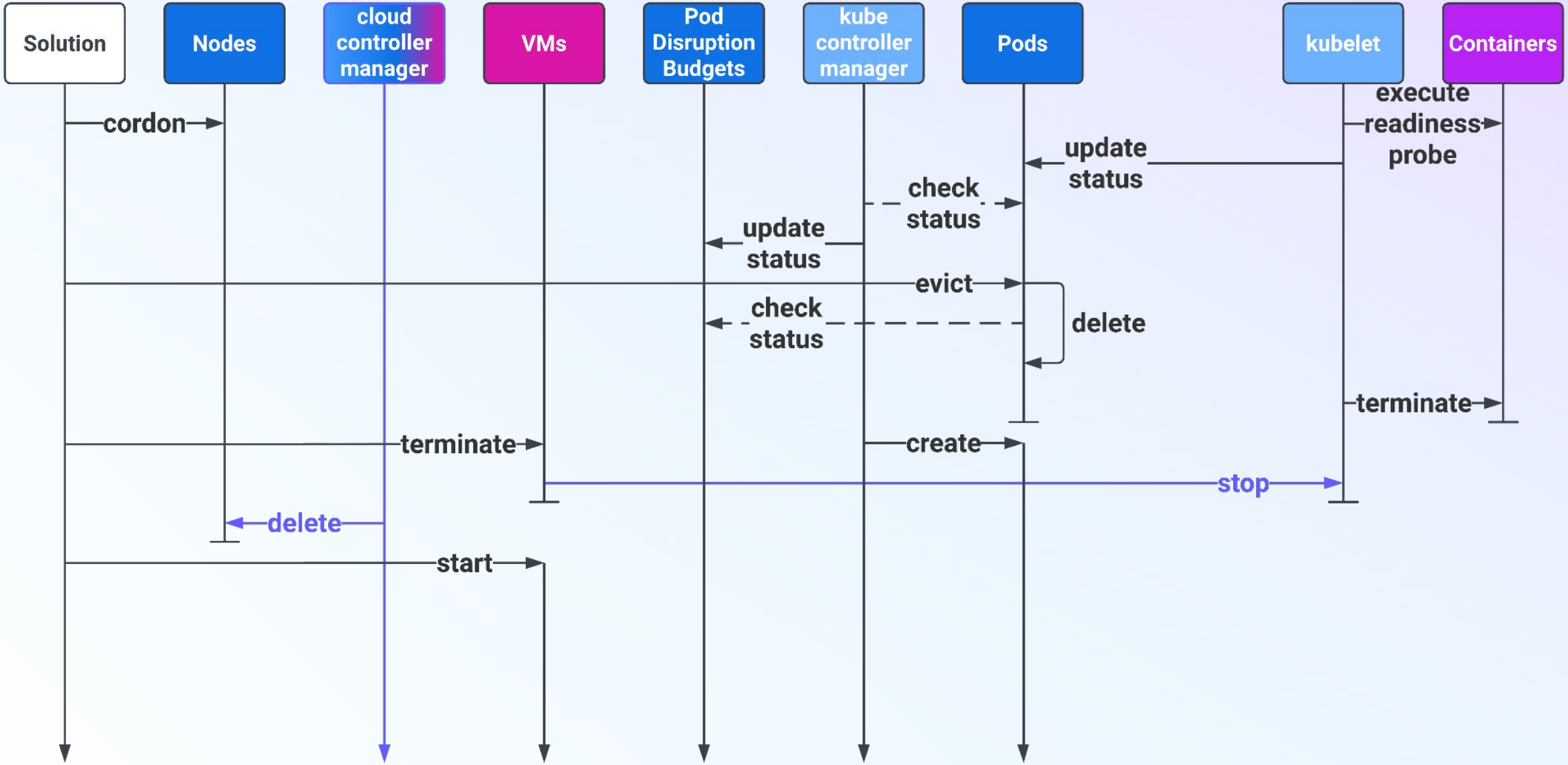
How to replace a node



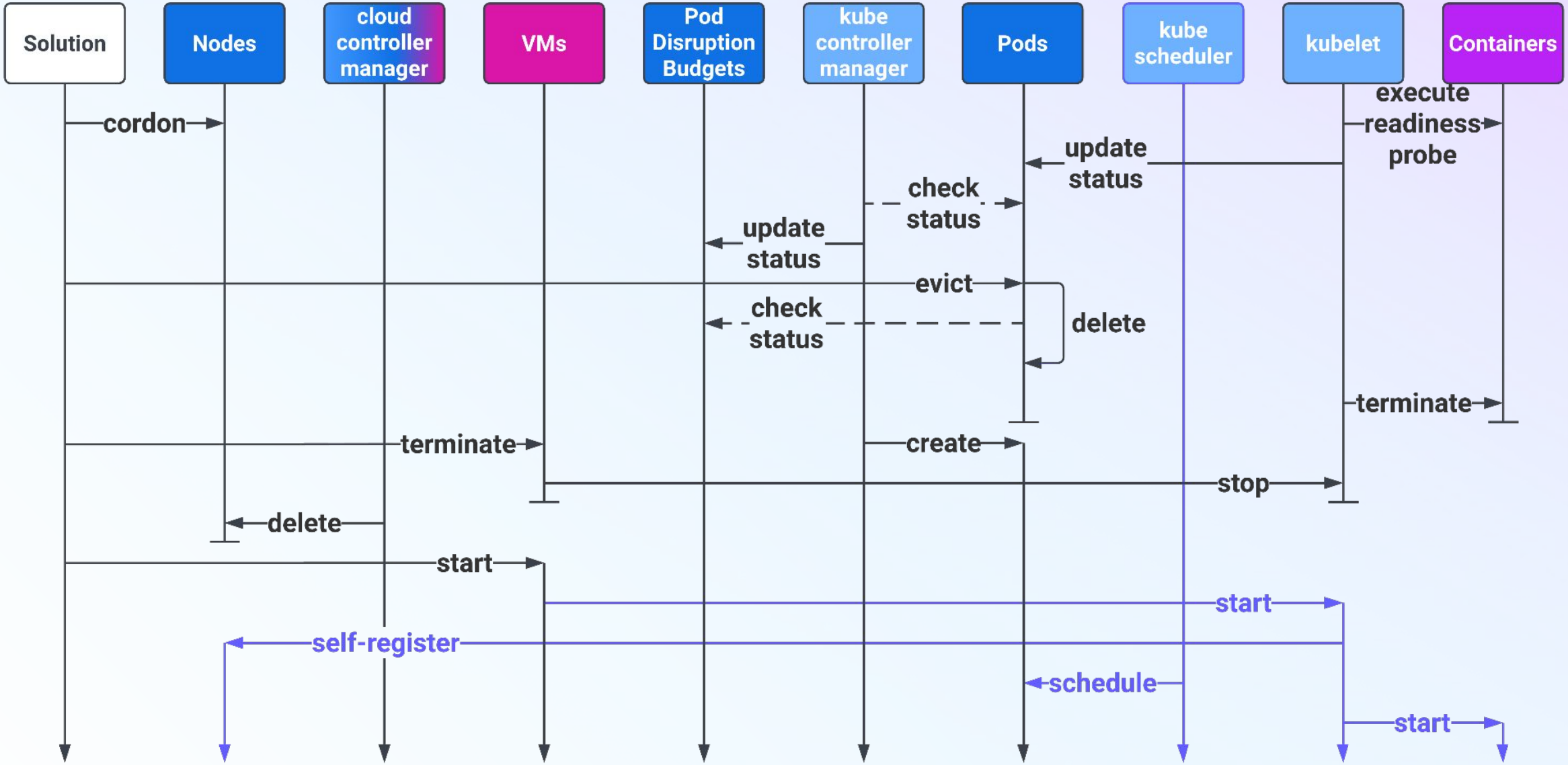
How to replace a node



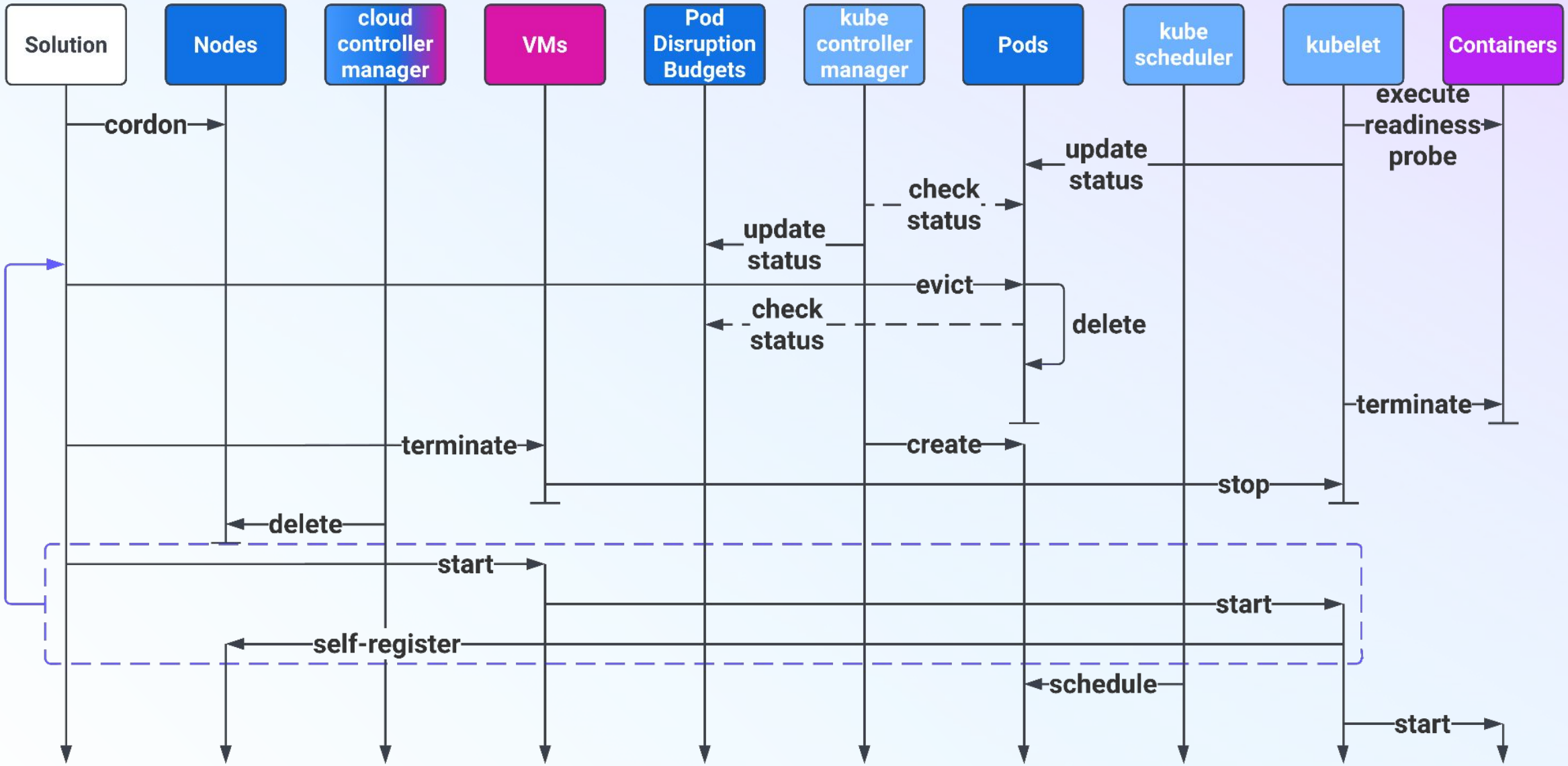
How to replace a node



How to replace a node



How to replace a node



How we do it

See “Datadog on Kubernetes Node Management” episode for details.

Datadog Node Problem Detector

NPD-like, running as a controller, watching Datadog monitors

Drain Controller

initially planetlabs/draino

Cluster Autoscaler (open source)

terminates empty nodes and scales up replacement nodes if needed

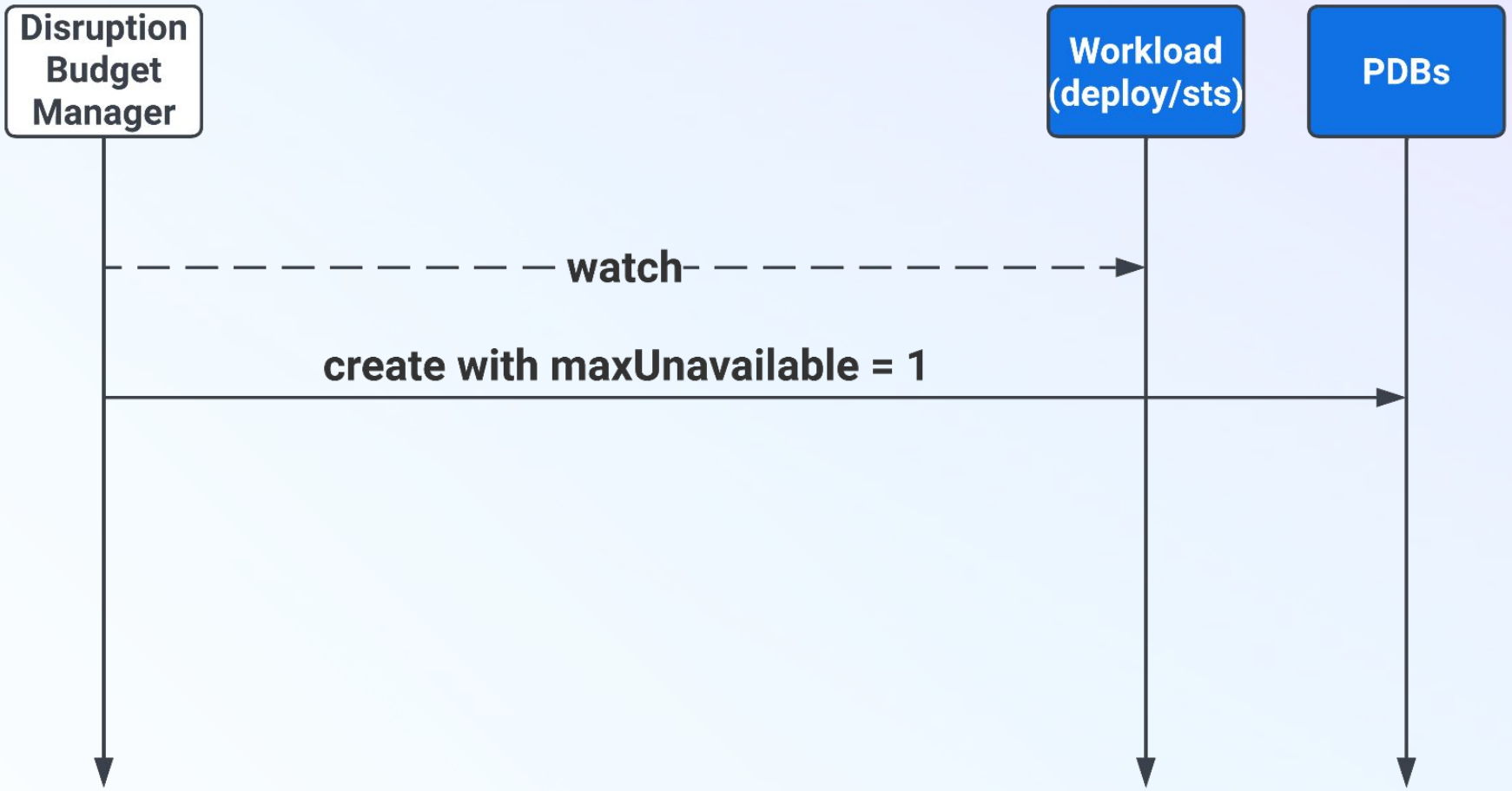
Disruption Budget Manager

default PDBs, dynamically updated from Datadog monitors and locks, evictions validated at admission

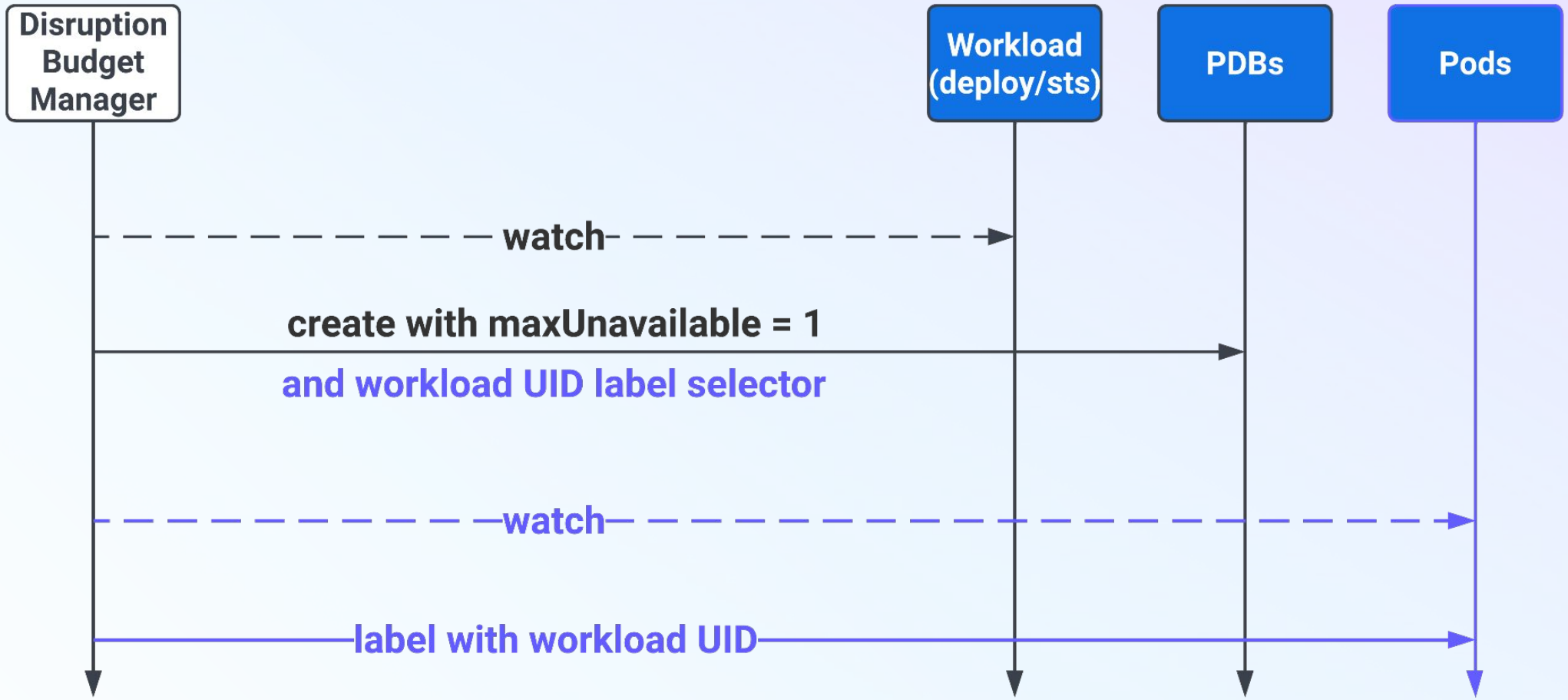
Carefully

The case for default PDBs

Conservative defaults



Conservative defaults, preventing PDB overlaps



Readiness probe overload

Readiness probes control both service availability and disruption budget.

Allow traffic but deny evictions

while under pressure

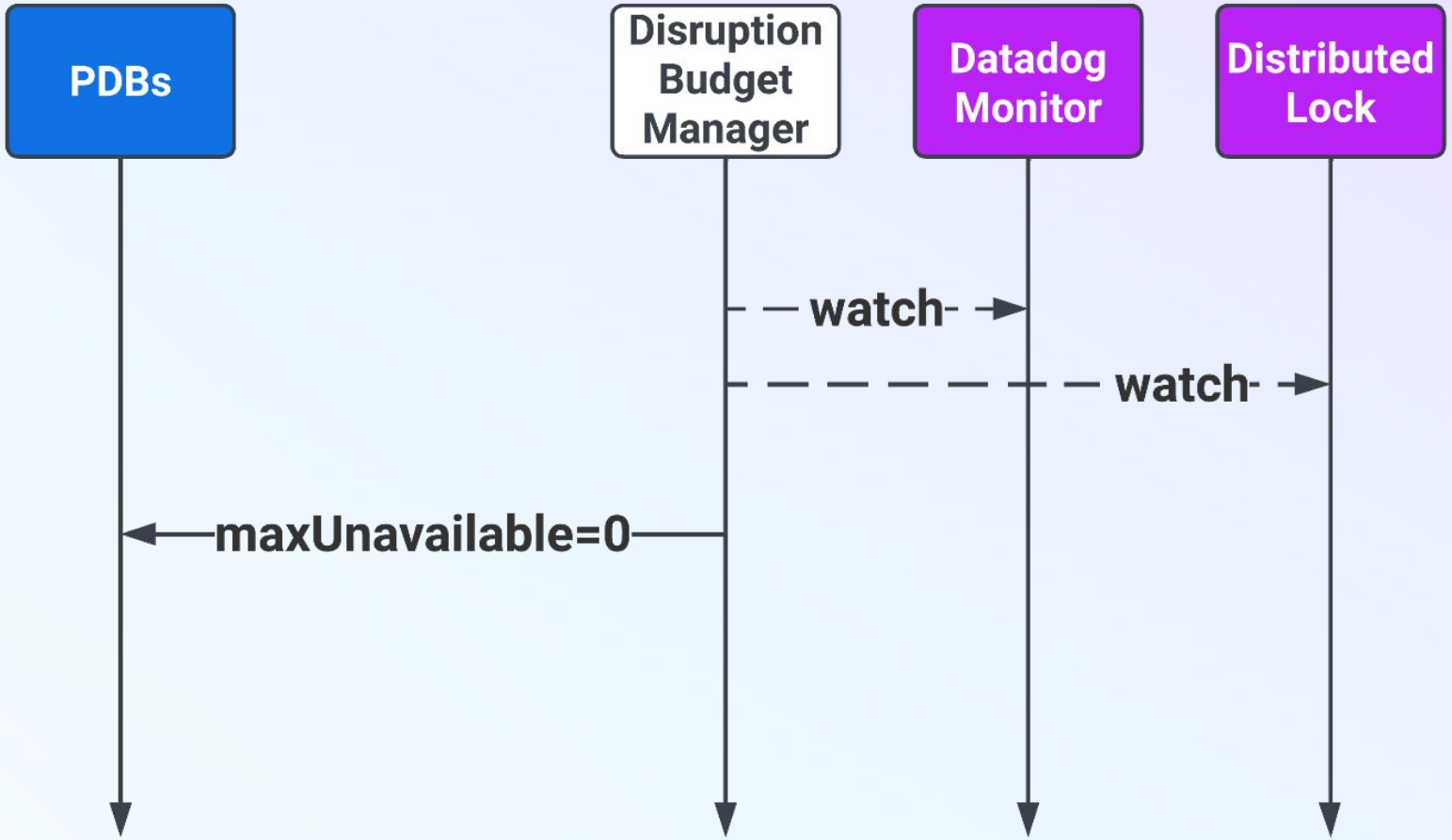
---

during operations

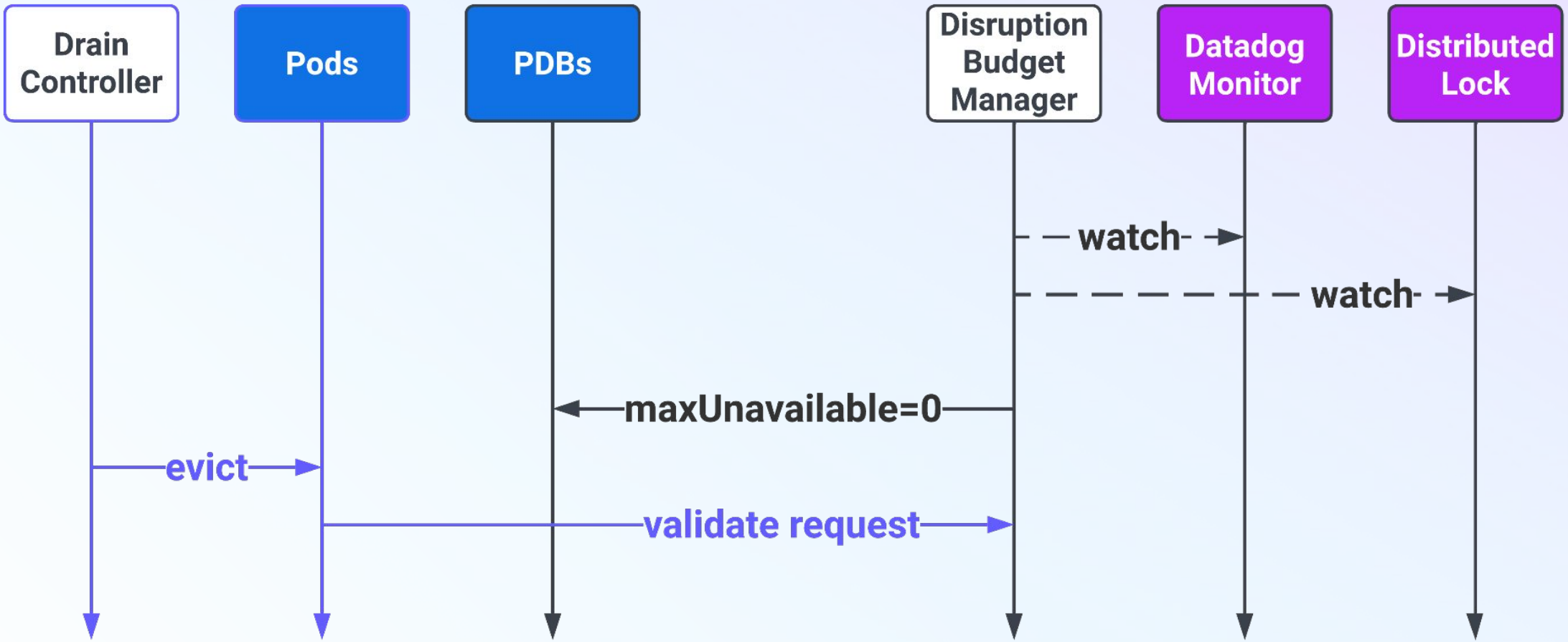
---

during incidents

Dynamic budget

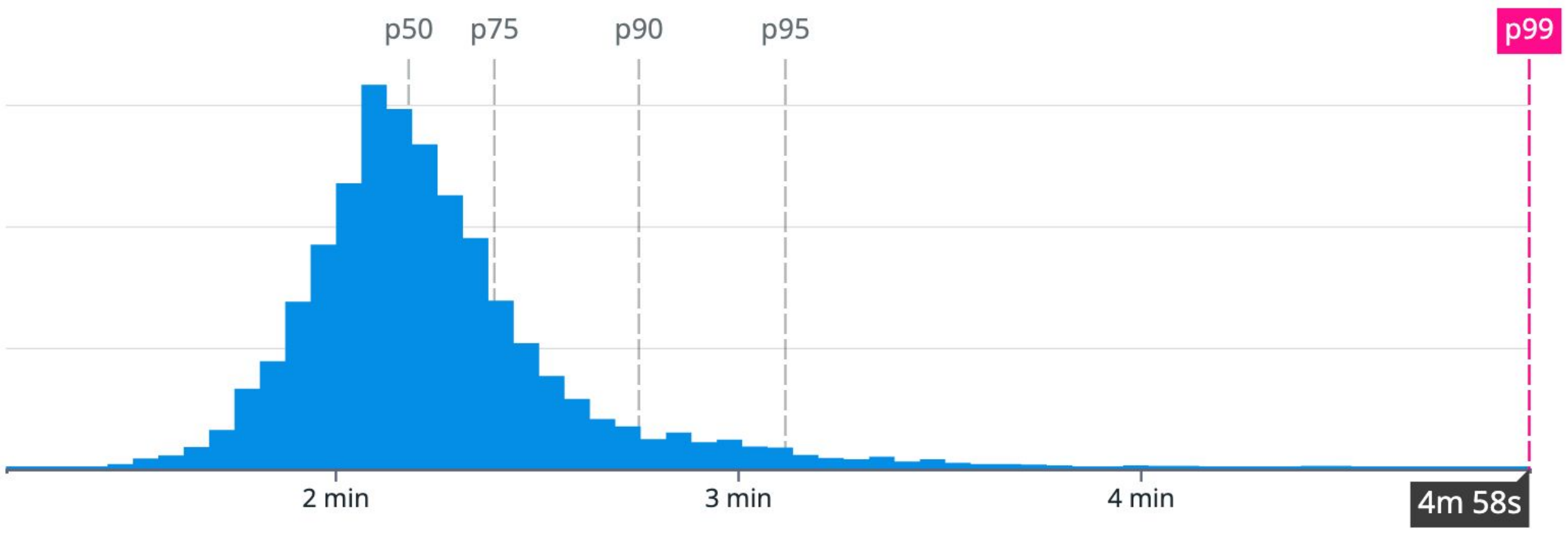


Dynamic budget and eviction validation

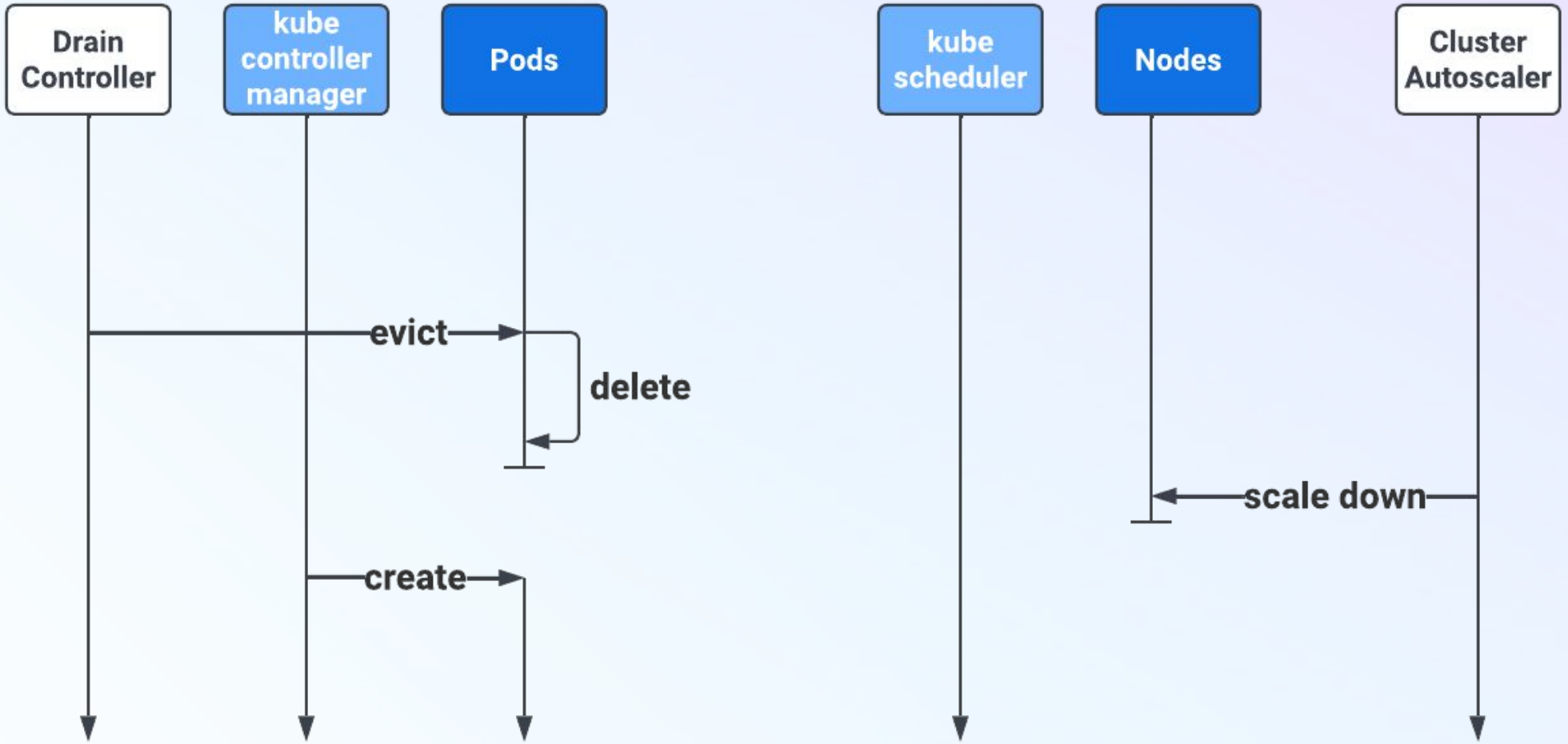


The missing node lifecycle hooks

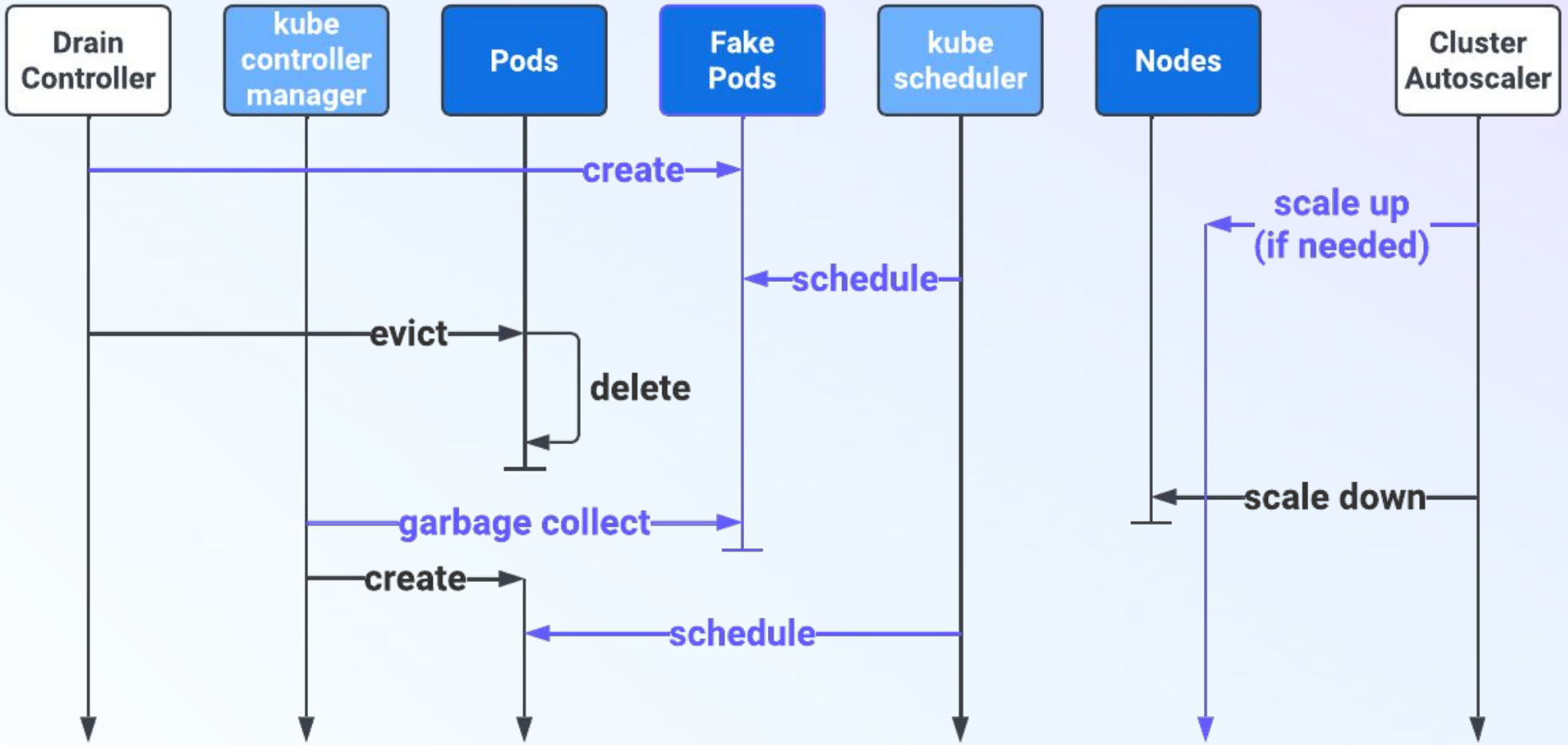
Pod scheduling latency when node scale-up needed



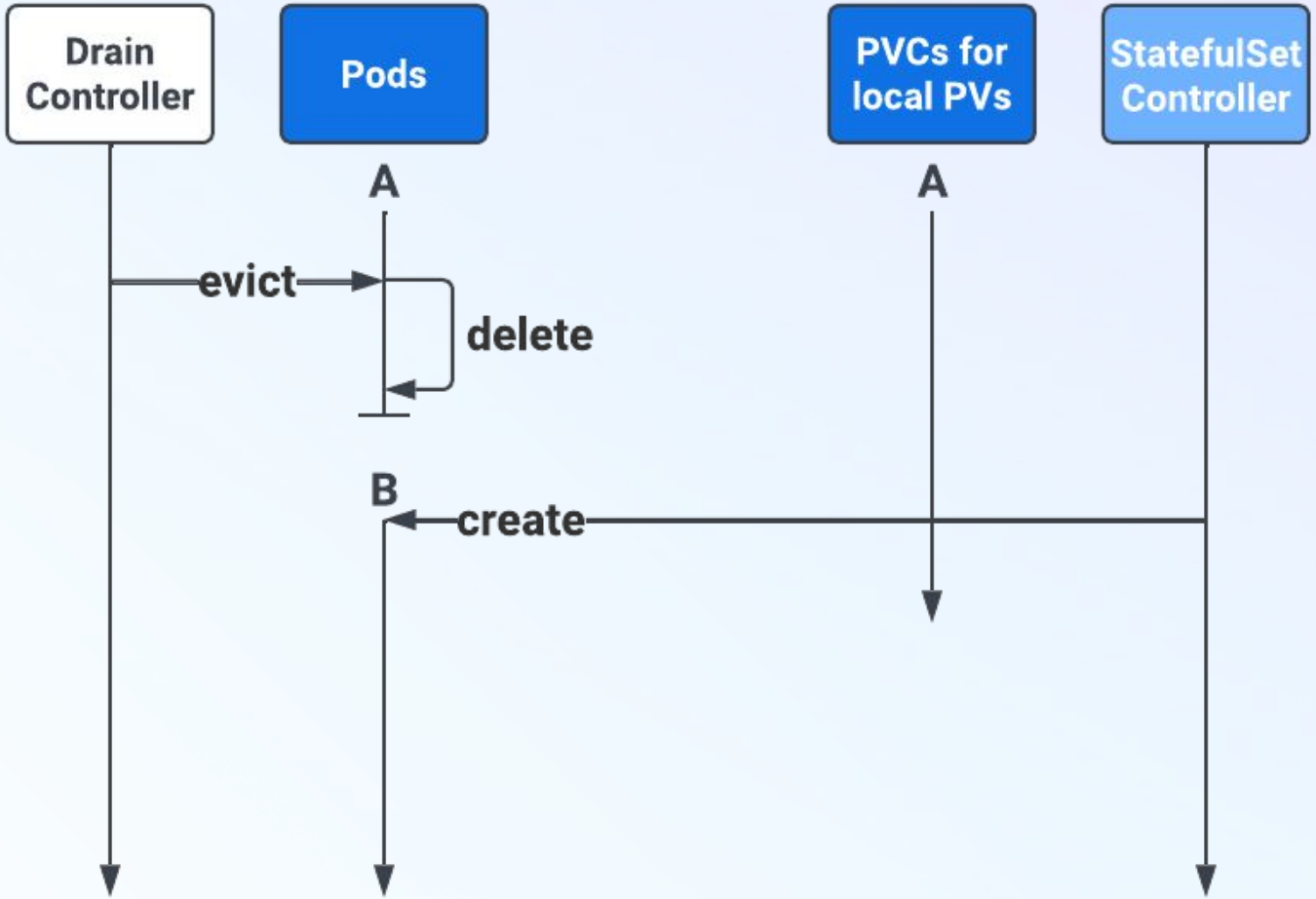
Node pre-provisioning



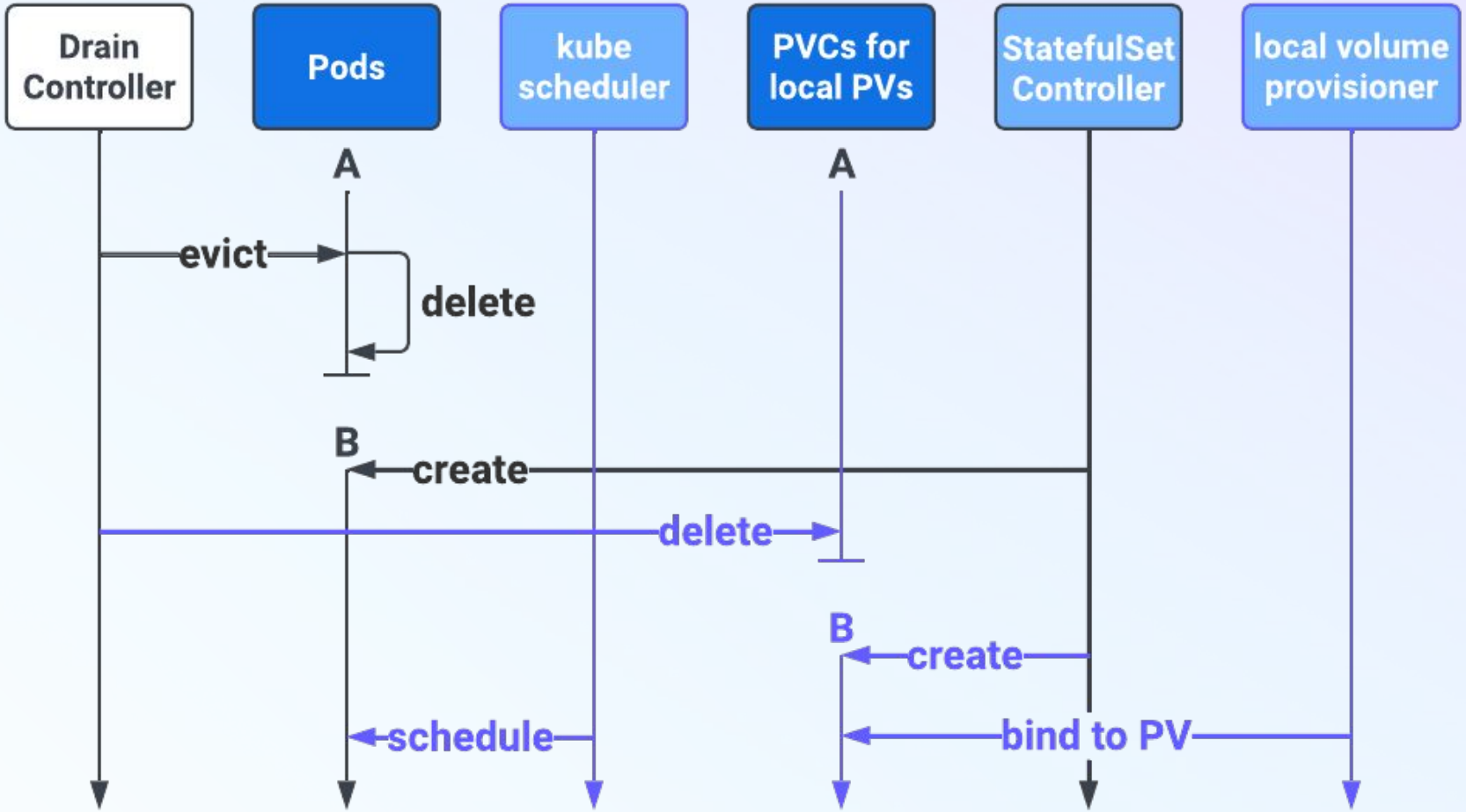
Node pre-provisioning



Persistent Volume Claim deletion



Persistent Volume Claim deletion



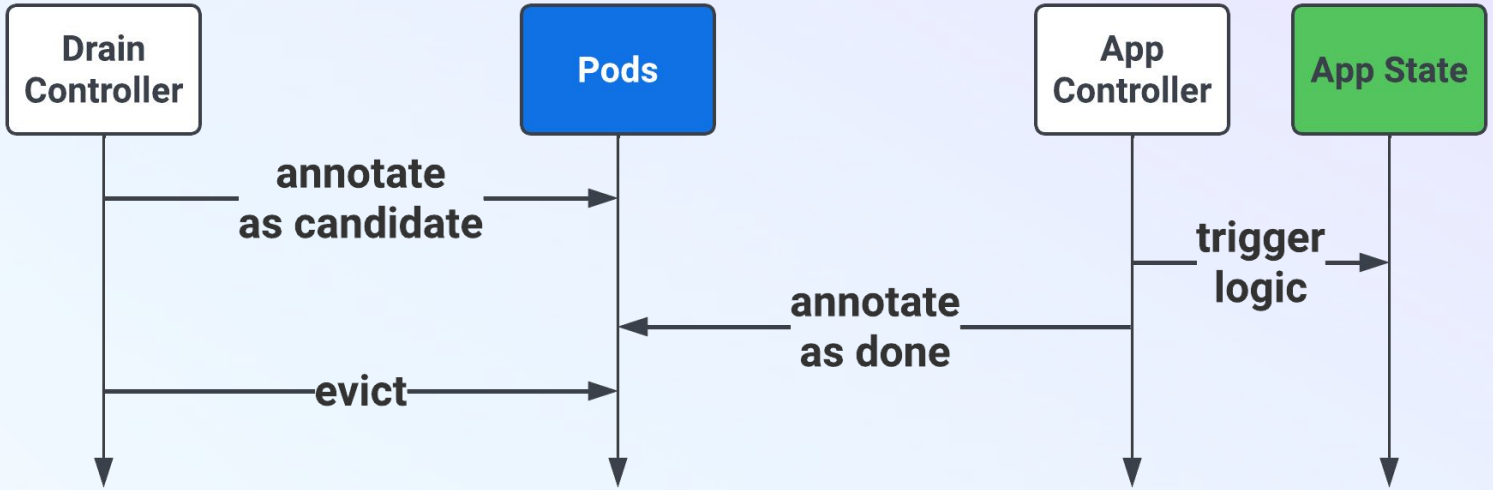
Pod eviction gate

App specific handling

Graceful / cancelable replica deletion

Examples

Leader election change, shard rebalancing, etc.



Recent, ongoing, and possible changes to Eviction and PDBs

RECENT CHANGE

Unhealthy pod eviction behavior

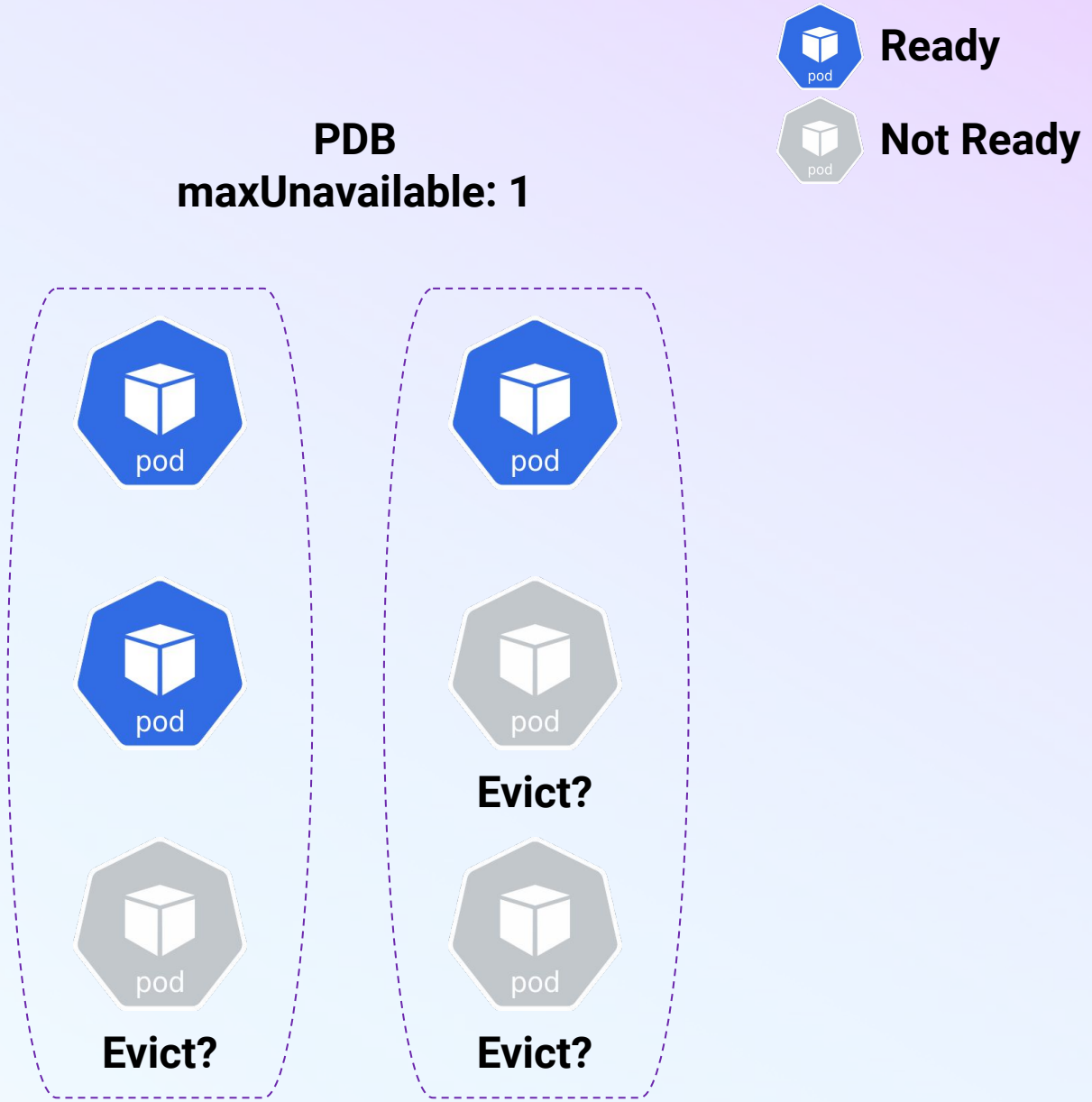
IfHealthyBudget

Since k8s 1.20

AlwaysAllow

Alternative policy since k8s 1.26 (alpha) / 1.27 (beta)

Risk of data loss



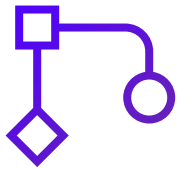
Possible changes

Disruption probe

Decouple service availability and pod disruption budget

Default PDB

Feature gate to protect deployments and stateful sets by default



Node lifecycle hooks

Optimize node deprovisioning at different node lifecycle stages

Should voluntary disruptions respect PDBs?

ONGOING CHANGE

Preemption respecting PDBs

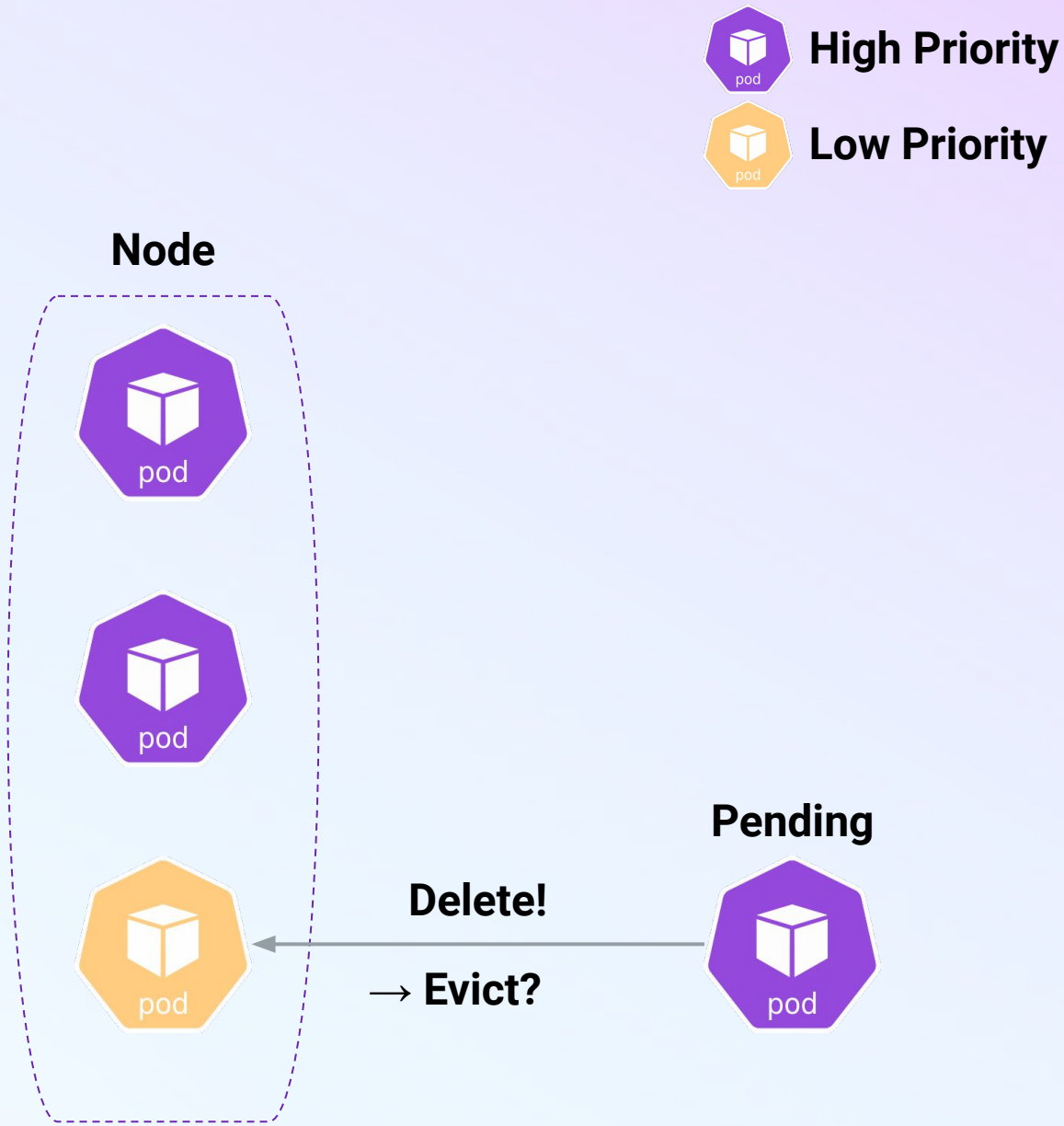
Currently “best effort”

DaemonSet rollout increasing resource requests caused internal incident

---

KEP 3280 approved to optionally respect PDBs

Implementation in progress



NOW

How taint-based eviction (TBE) works

Some node conditions propagated as NoExecute taints

Ready=False/Unknown, Disk/Memory/PIDPressure, NetworkUnavailable

Pods **deleted** if taint not tolerated or after tolerationSeconds

POSSIBLE CHANGE

TBE for drains

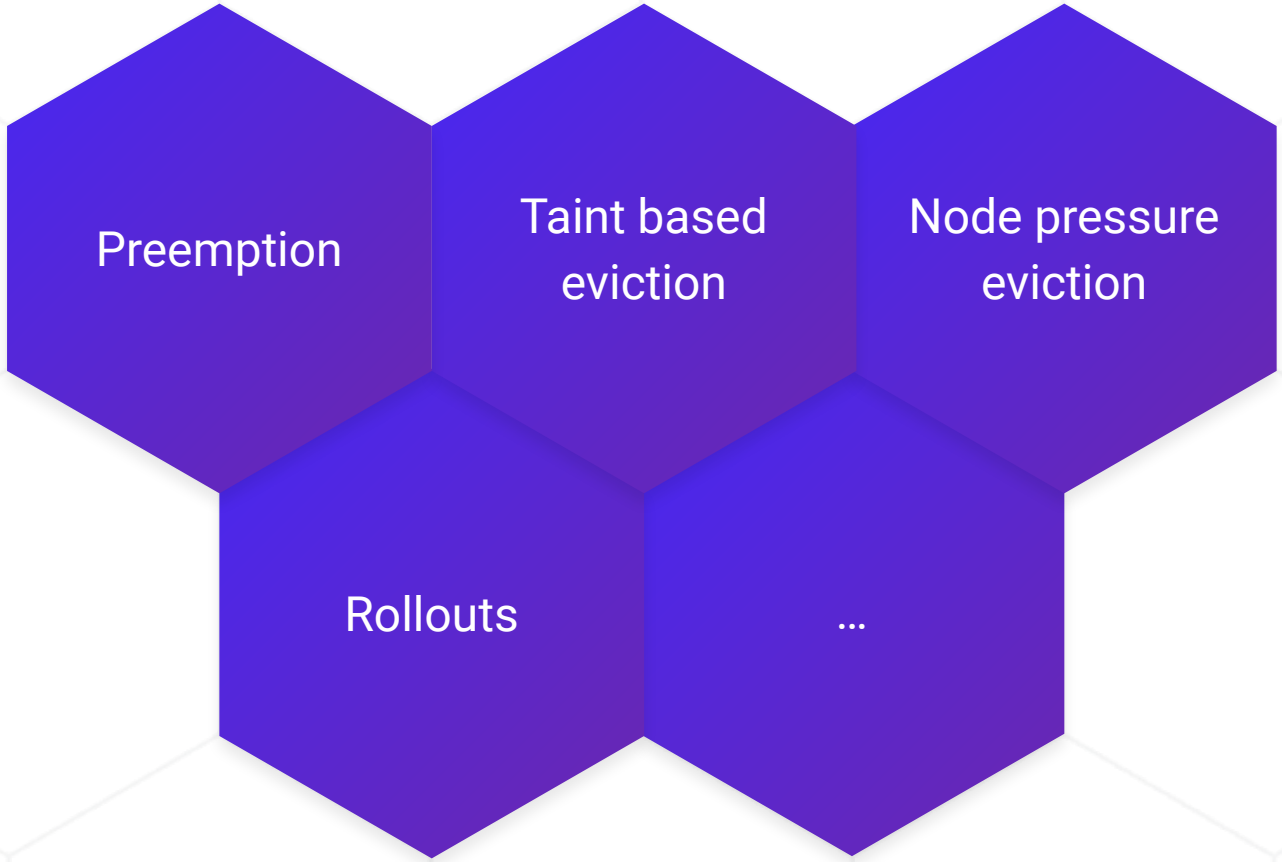
Configurable list of node conditions propagated as taints

Option to evict vs. delete, possibly falling back to delete

Fewer controllers

Evict always* because it’s guaranteed** safe

* except emergency
** PDB status is a lagging indicator



Thank you

Questions?

DATADOG