
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

SIG Scheduling Updates

Kensei Nakada
Mercari

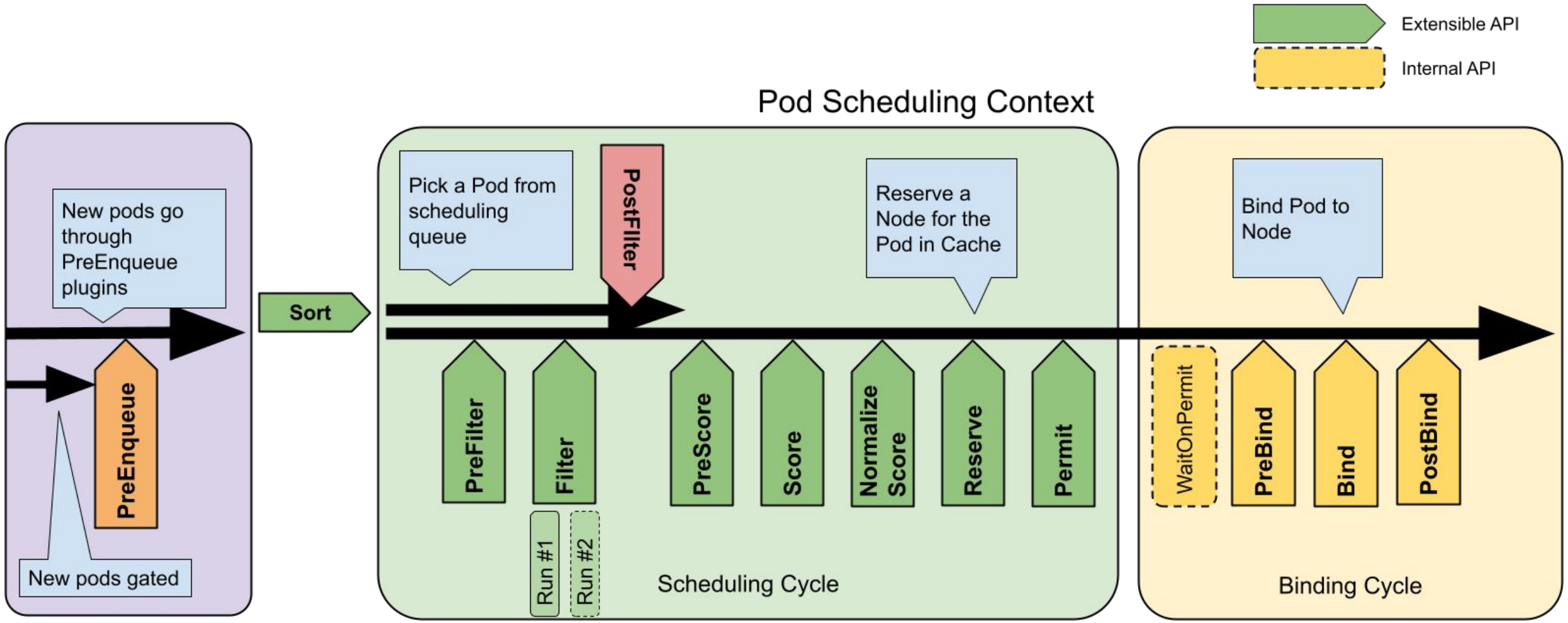
Aldo Culquicondor
Google

Agenda

- kube-scheduler overview
- Recent improvements in kube-scheduler
- Sub-projects updates
  - kube-scheduler-wasm-extension
  - kueue
  - descheduler
  - kwok

Scheduler Overview

Scheduling Framework



Recent Improvements

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

Alpha @ v1.29

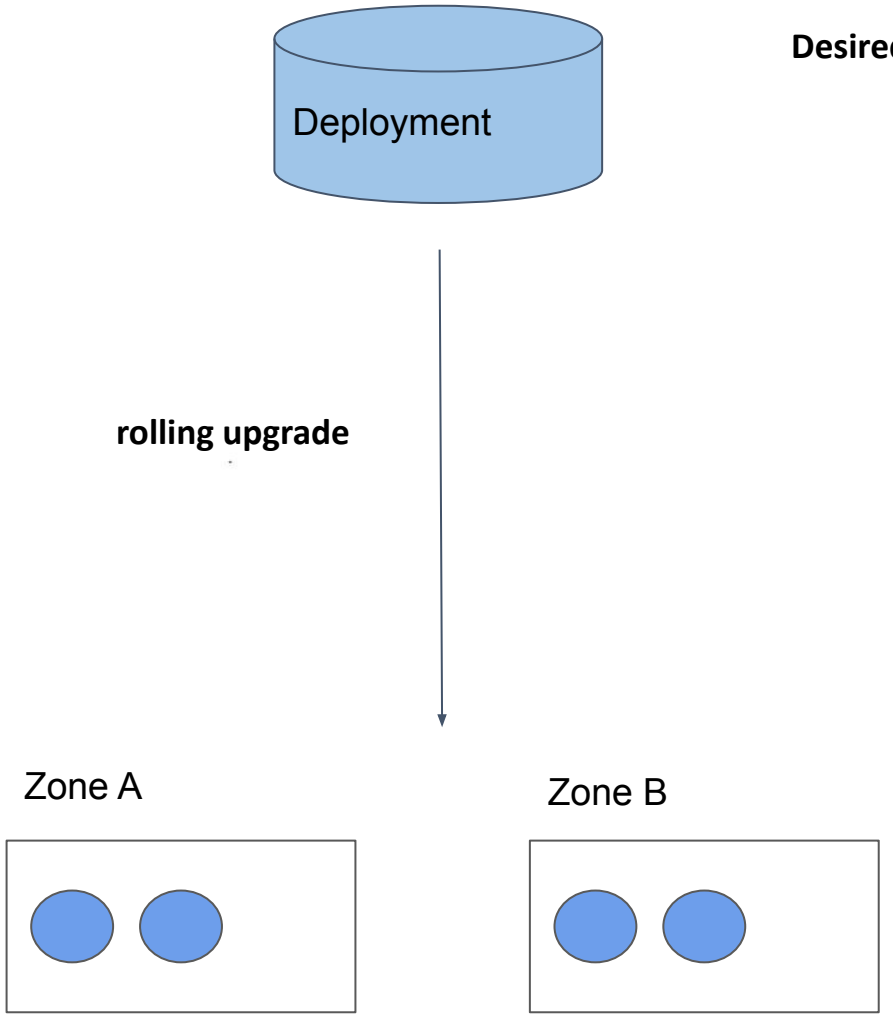
matchLabelKeys specifies the keys for the labels that should match with the incoming Pod's labels, when satisfying the Pod (anti)affinity.

A common use-case is to add **pod-template-hash** to a Deployment so that the PodAffinity is only calculated based on the new Pods in the new revision (ReplicaSet) of the Deployment.

Example: All pods of the Deployment need to be in the same zone, but the zone can change during upgrades.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

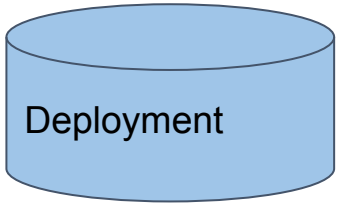


Desired outcome: consolidate all Pods of the new version in the same zone.

Without matchLabelKeys

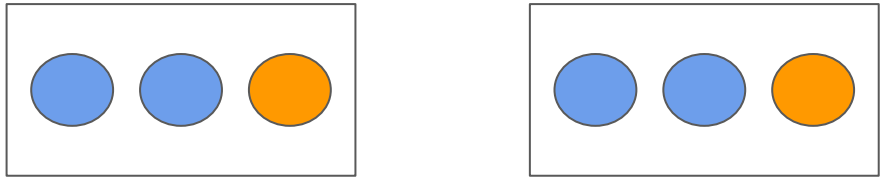
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity



rolling upgrade

The scheduler takes Pods in both old/new version into consideration

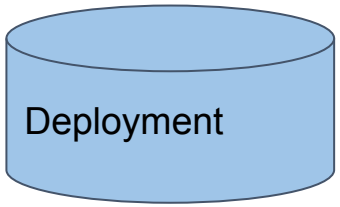


Desired outcome: consolidate all Pods of the new version in the same zone.

Without matchLabelKeys

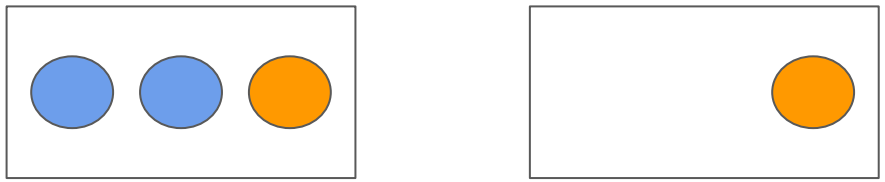
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity



rolling upgrade

The scheduler takes Pods in both old/new version into consideration

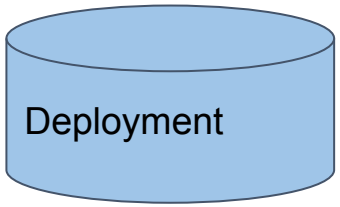


Desired outcome: consolidate all Pods of the new version in the same zone.

Without matchLabelKeys

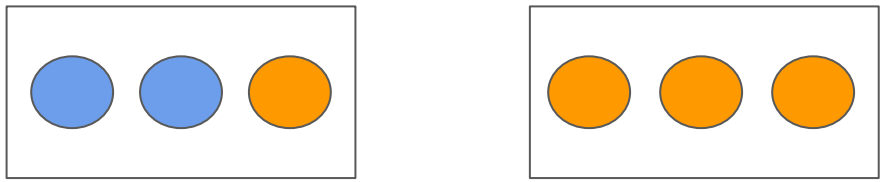
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity



rolling upgrade

The scheduler takes Pods in both old/new version into consideration

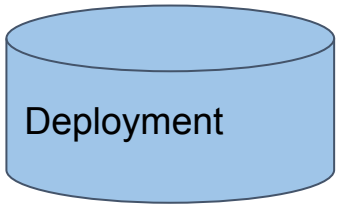


Desired outcome: consolidate all Pods of the new version in the same zone.

Without matchLabelKeys

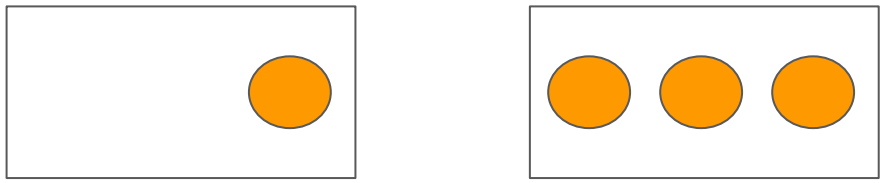
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity



rolling upgrade

The scheduler takes Pods in both old/new version into consideration



FAILED!

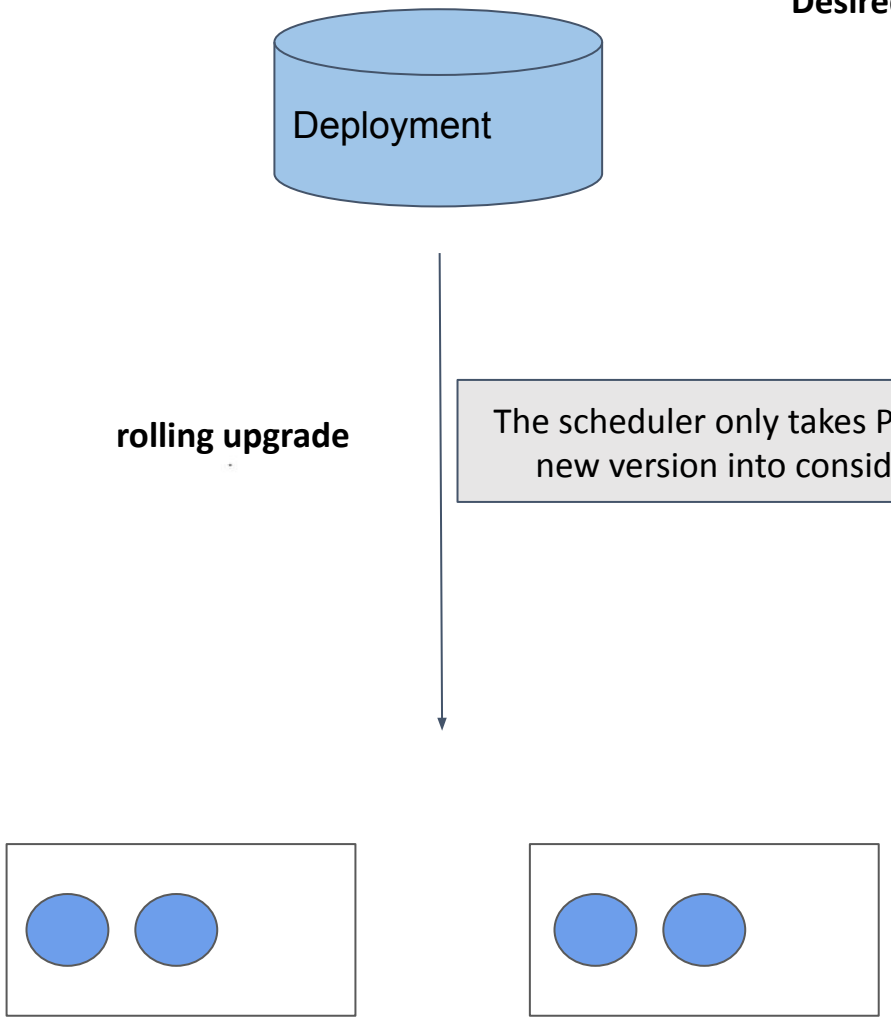
Desired outcome: consolidate all Pods of the new version in the same zone.

Without matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

Desired outcome: consolidate all Pods of the new version in the same zone.

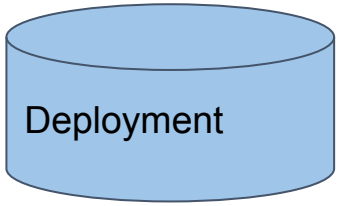


With matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

Desired outcome: consolidate all Pods of the new version in the same zone.



rolling upgrade

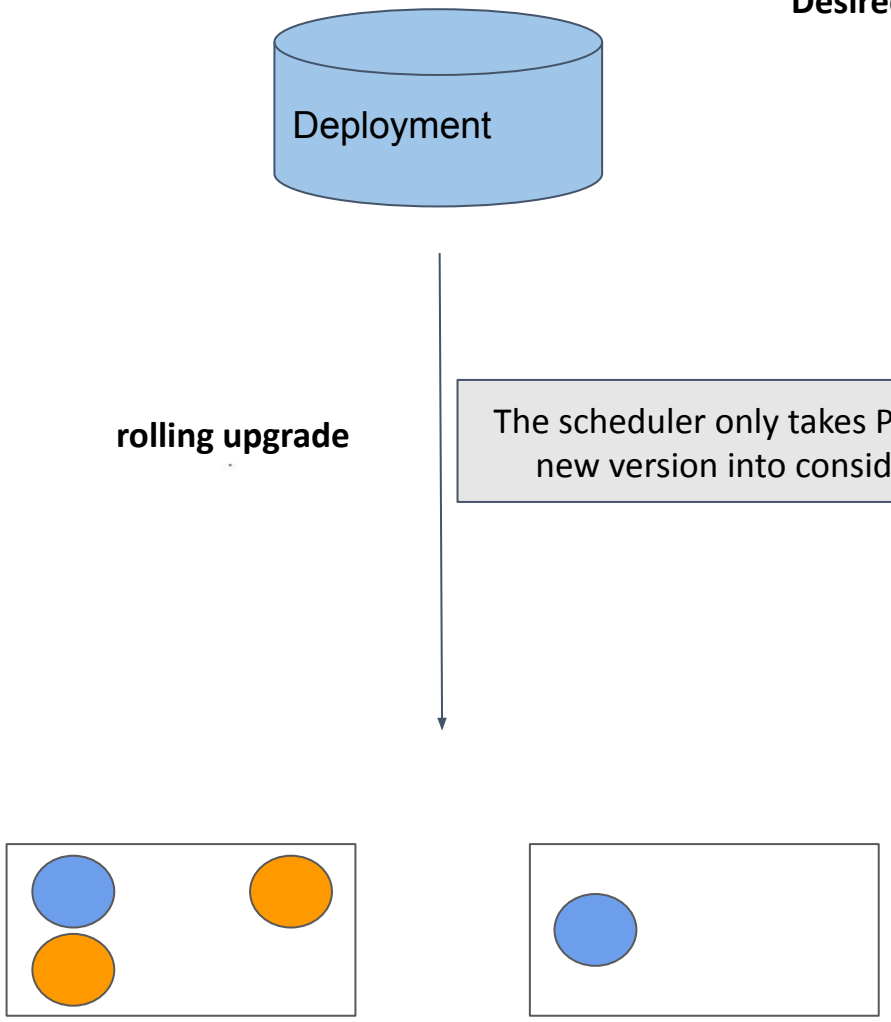
The scheduler only takes Pods in the new version into consideration

With matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

Desired outcome: consolidate all Pods of the new version in the same zone.

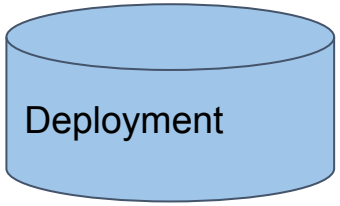


With matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

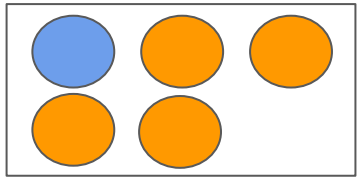
KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

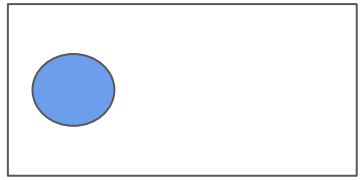
Desired outcome: consolidate all Pods of the new version in the same zone.



rolling upgrade

The scheduler only takes Pods in the new version into consideration



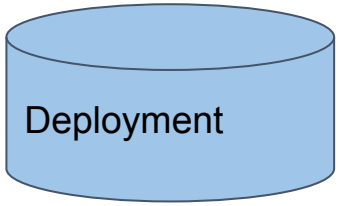


With matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

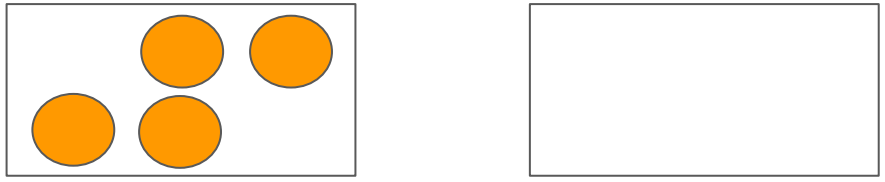
KEP-3633: MatchLabelKeys in Pod(Anti)Affinity

Desired outcome: consolidate all Pods of the new version in the same zone.



rolling upgrade

The scheduler only takes Pods in the new version into consideration



SUCCESS!

With matchLabelKeys

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: application-server
...
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - tenant-a
        topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
        - pod-template-hash
```

KEP-3633: MismatchLabelKeys in Pod(Anti)Affinity

Alpha @ v1.29

mismatchLabelKeys specifies the keys for the labels that should NOT match with the incoming Pod's labels, when satisfying the Pod (anti)affinity.

```yaml
affinity:
  podAffinity:
    # ensures the pods of this tenant land on the same node
    requiredDuringSchedulingIgnoredDuringExecution:
    - matchLabelKeys:
        - tenant
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    # ensures only Pods from this tenant lands on the same node
    requiredDuringSchedulingIgnoredDuringExecution:
    - mismatchLabelKeys:
        - tenant
      labelSelector:
        matchExpressions:
        - key: tenant
          operator: Exists
      topologyKey: kubernetes.io/hostname
```

KEP-3633: MismatchLabelKeys in Pod(Anti)Affinity

Example:

- **podAffinity** ensures this Pod goes to the Node which already has Pods from the same tenant.
- **podAntiAffinity** ensures this Pod goes to the Node which does not have Pods from different tenants.

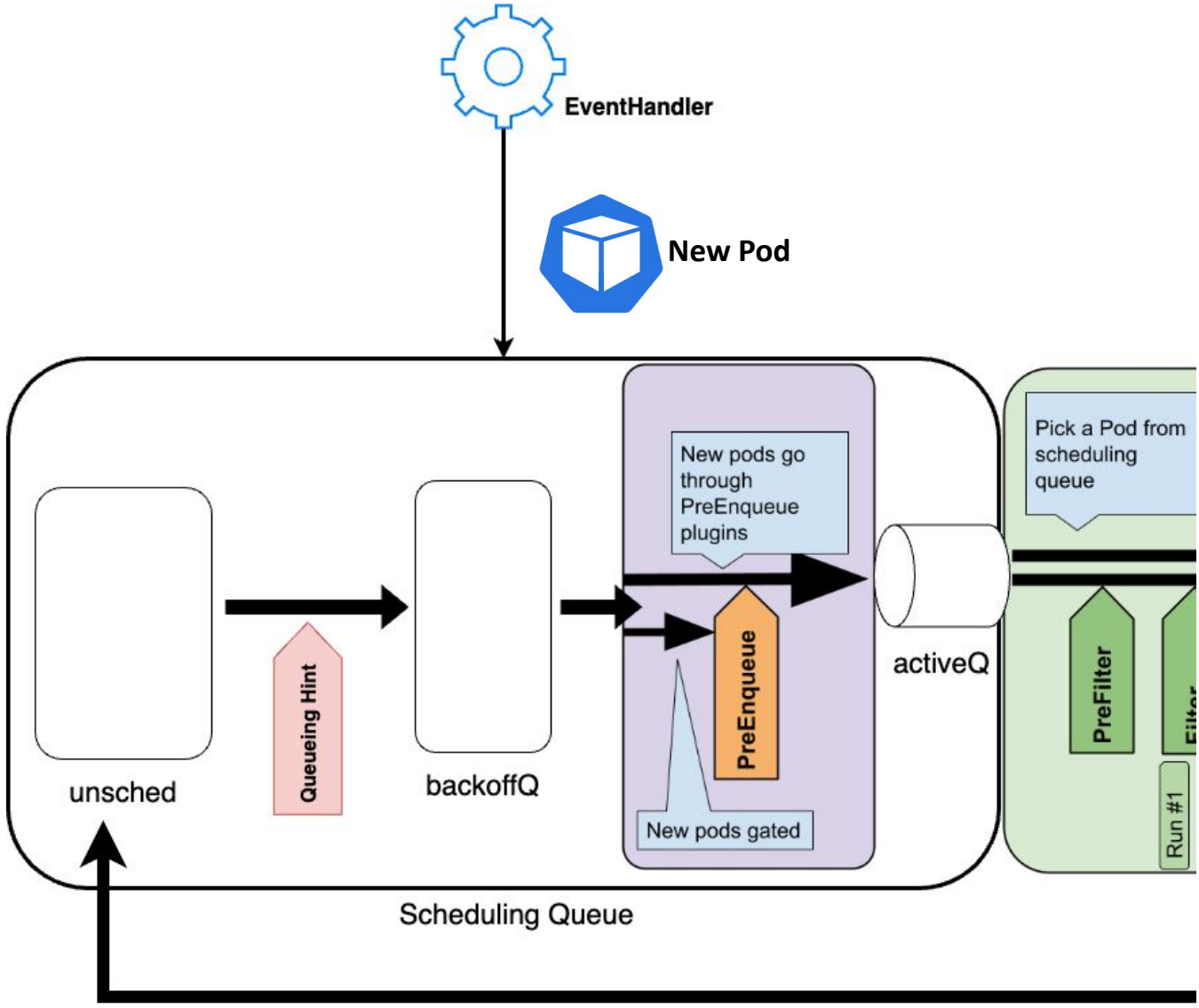
As a result, these rules ensure that this Pod goes to the Node which has only Pods from the same tenant.

```yaml
affinity:
  podAffinity:
    # ensures the pods of this tenant land on the same node
    requiredDuringSchedulingIgnoredDuringExecution:
    - matchLabelKeys:
        - tenant
      topologyKey: kubernetes.io/hostname
  podAntiAffinity:
    # ensures only Pods from this tenant lands on the same node
    requiredDuringSchedulingIgnoredDuringExecution:
    - mismatchLabelKeys:
        - tenant
      labelSelector:
        matchExpressions:
        - key: tenant
          operator: Exists
      topologyKey: kubernetes.io/hostname
```

KEP-4247: QueueingHint

Example:

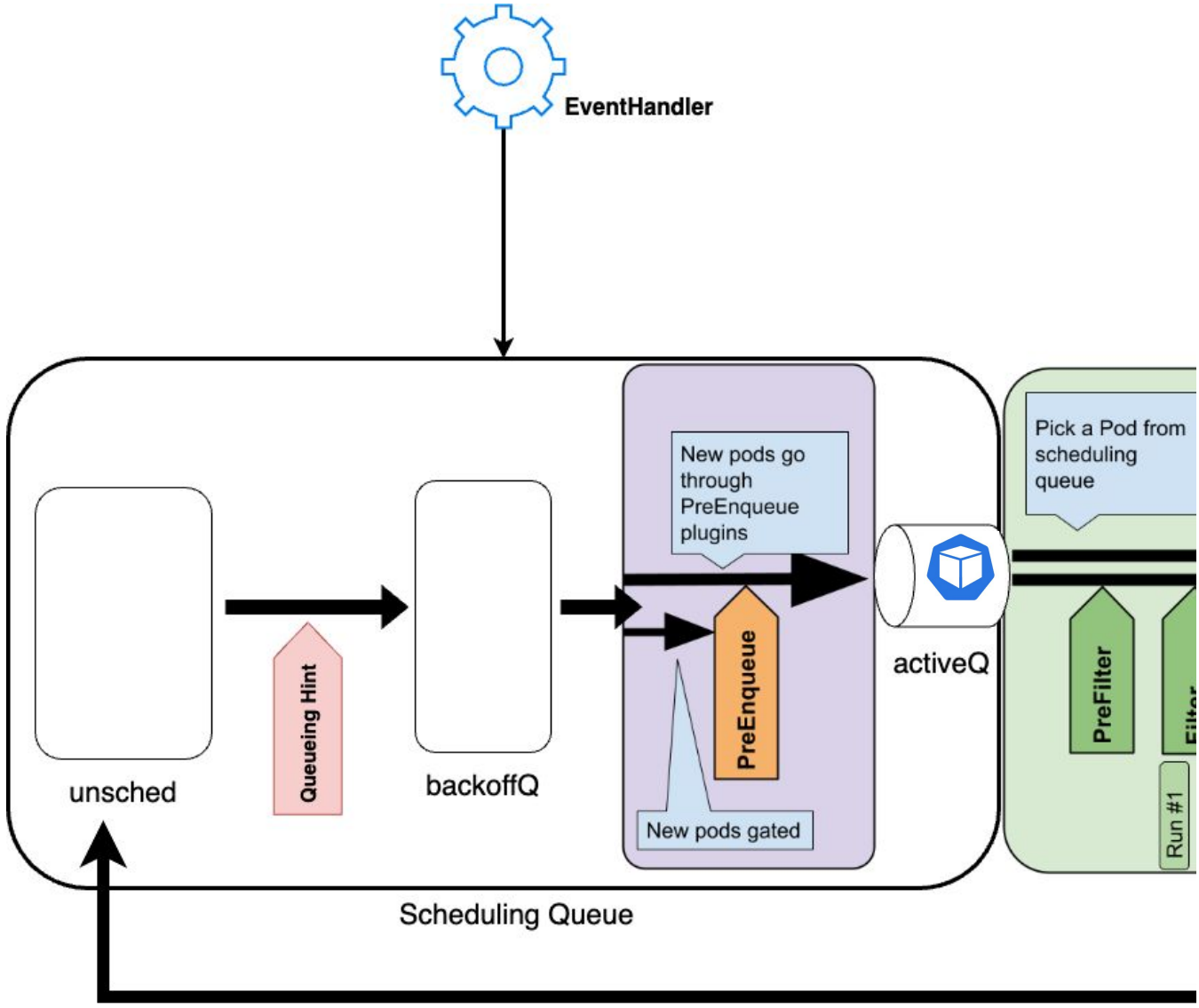
1. Create a Pod which has a required PodAffinity.
2. The scheduler notices the Pod via EventHandler,



KEP-4247: QueueingHint

Example:

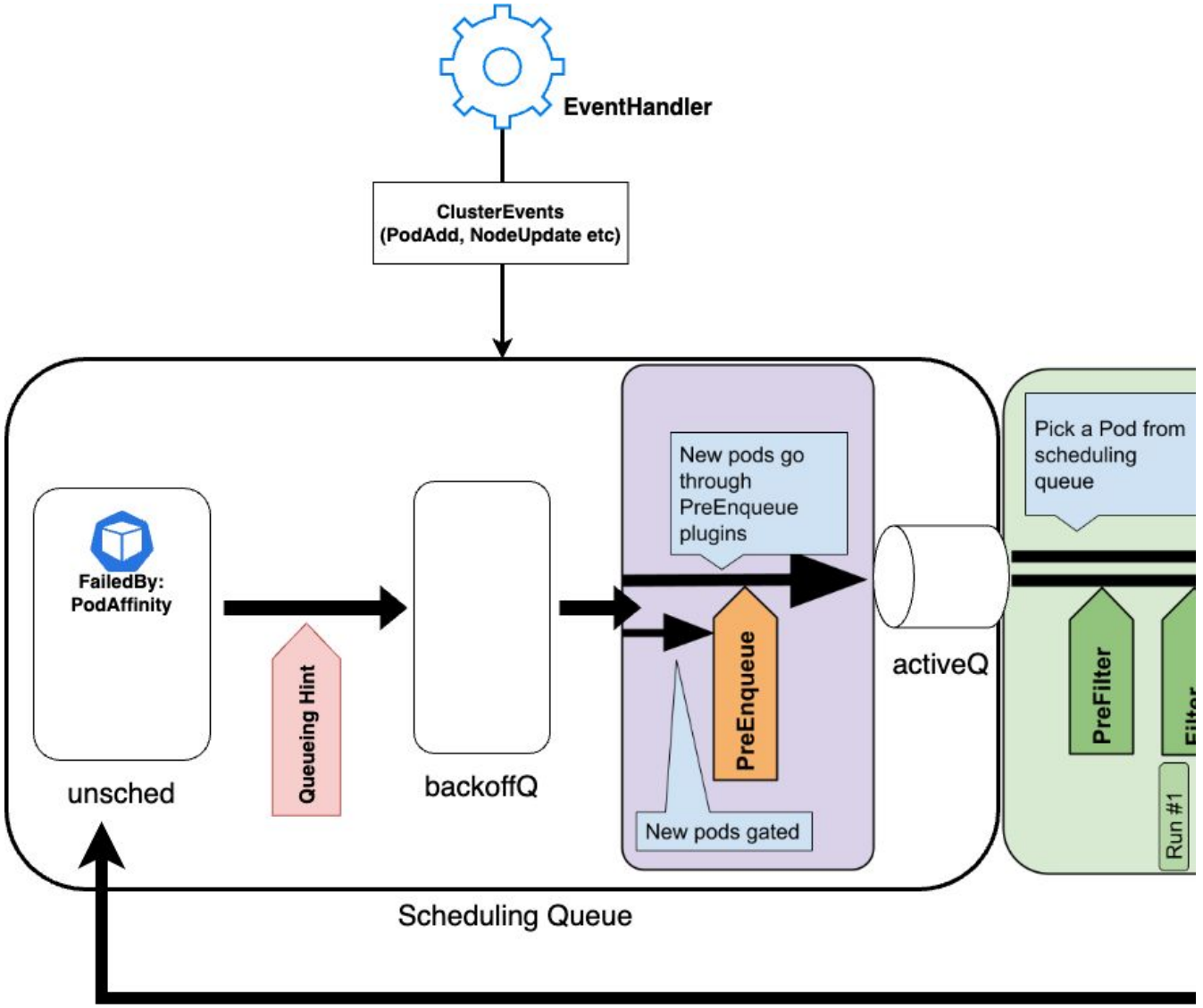
1. Create a Pod which has a required PodAffinity.
2. The scheduler notices the Pod via EventHandler, puts it into activeQ, and try to schedule it.



KEP-4247: QueueingHint

Example:

1. Create a Pod which has a required PodAffinity.
2. The scheduler notices the Pod via EventHandler, puts it into activeQ, and try to schedule it.
3. Pod is rejected by PodAffinity plugin and put back into Scheduling Queue. Scheduling Queue remembers who rejected this Pod.

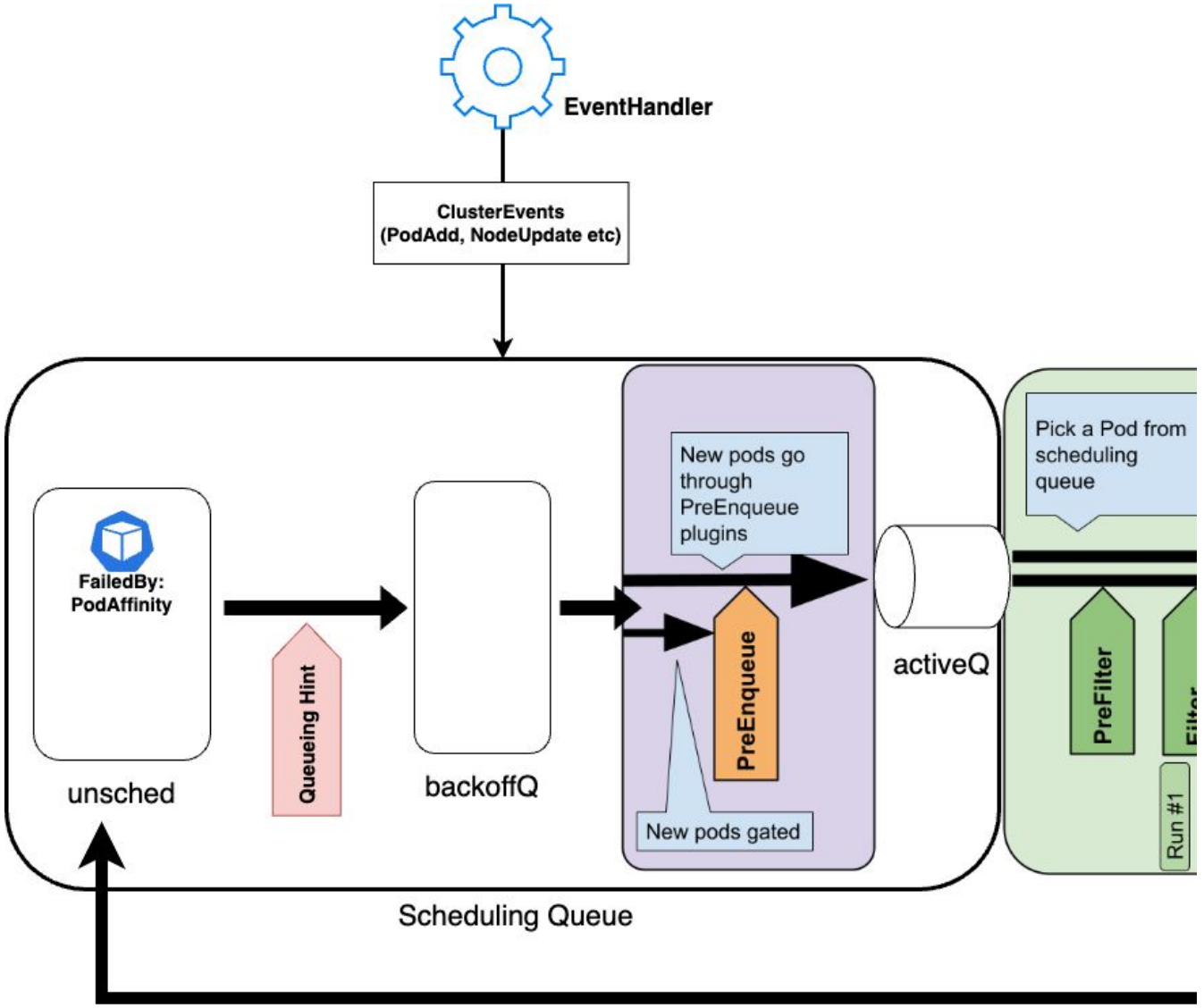


KEP-4247: QueueingHint

Beta @ v1.28

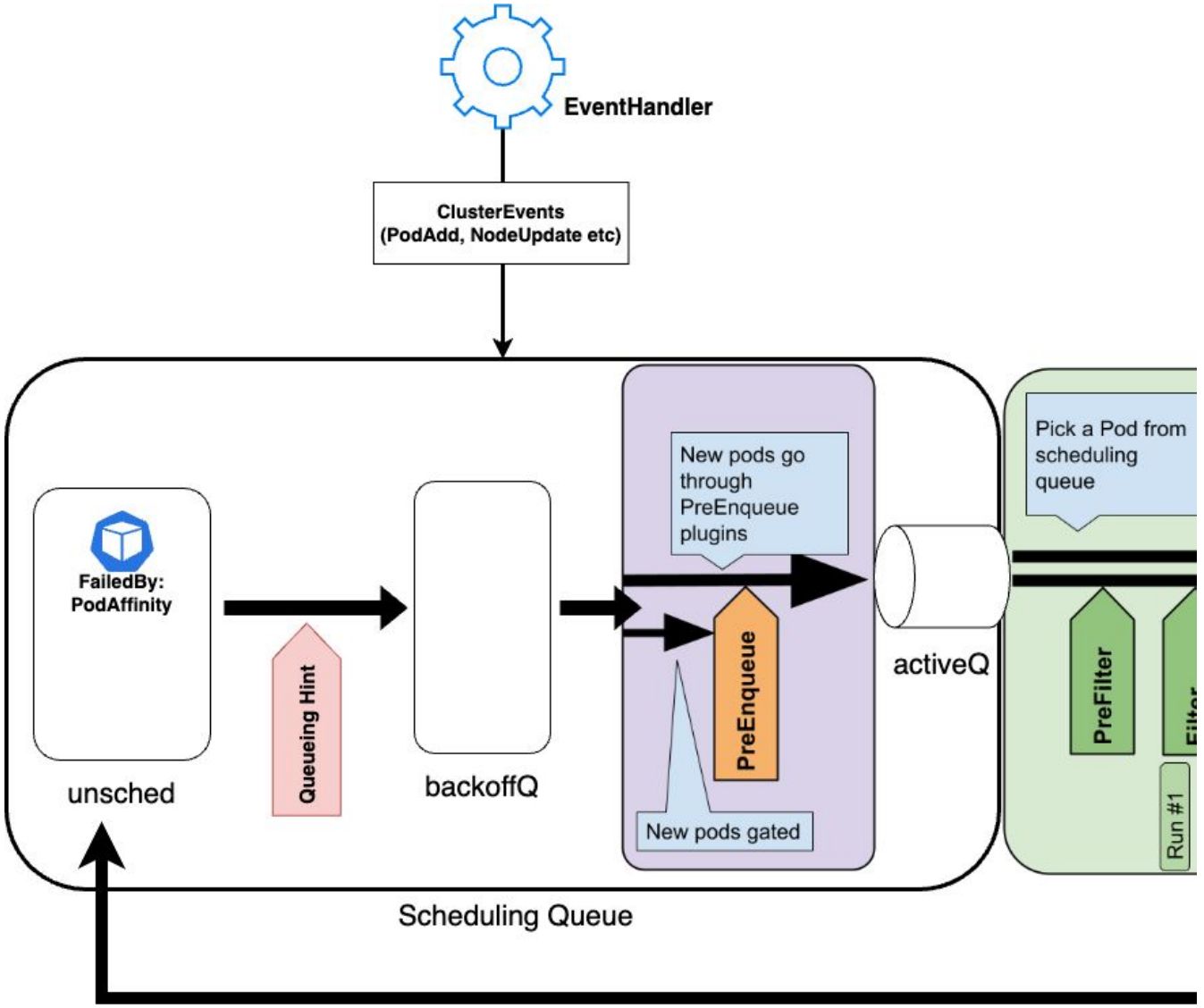
A **QueueingHint** is a callback function, per plugin, to decide if a Pod can be requeued to activeQ/backoffQ when an event happens.

Goal: Avoid wasteful scheduling cycles by filtering the cluster events that could make a Pod schedulable.



KEP-4247: QueueingHint

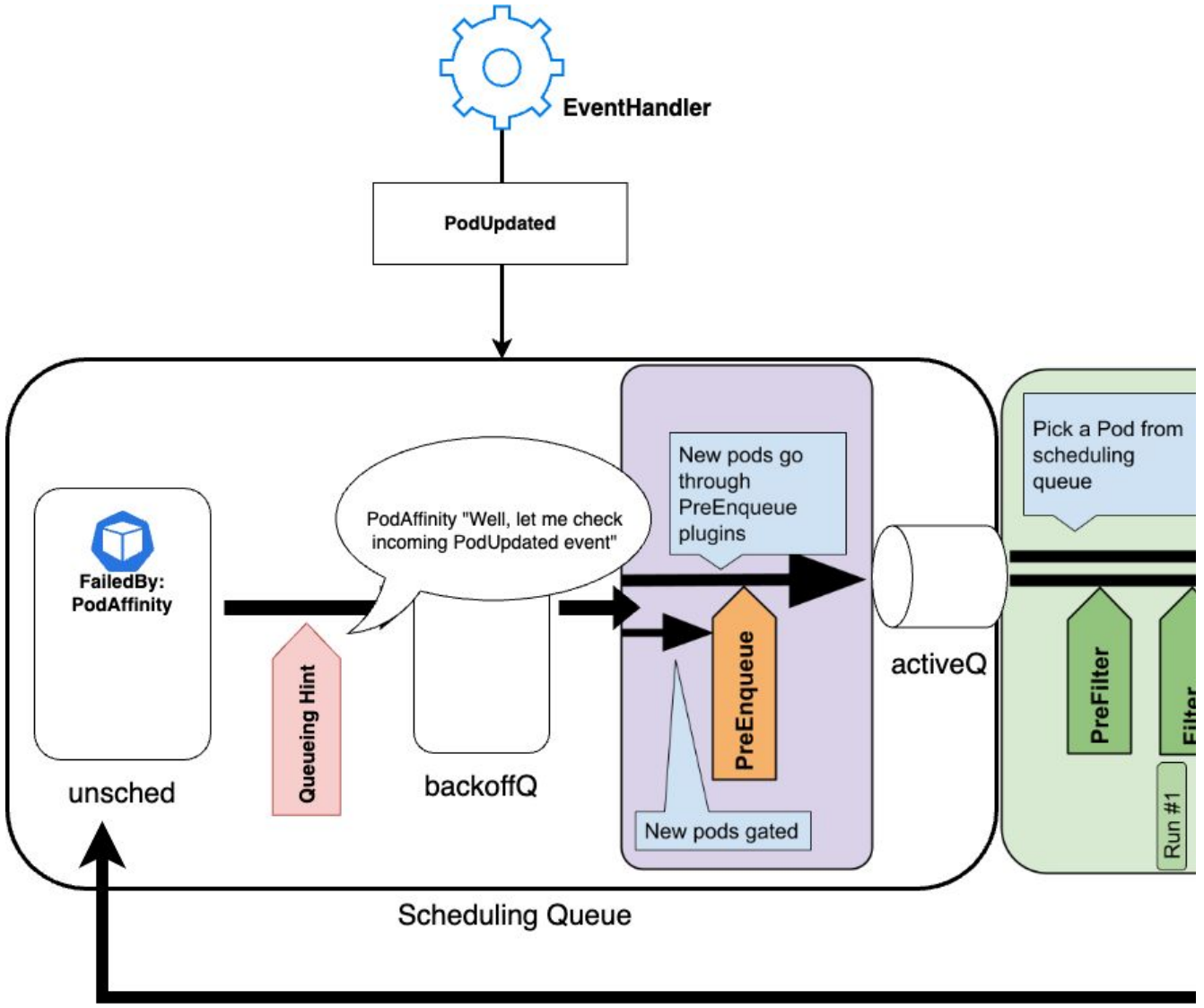
The scheduling queue subscribes cluster events (PodAdd, NodeUpdate… etc).



KEP-4247: QueueingHint

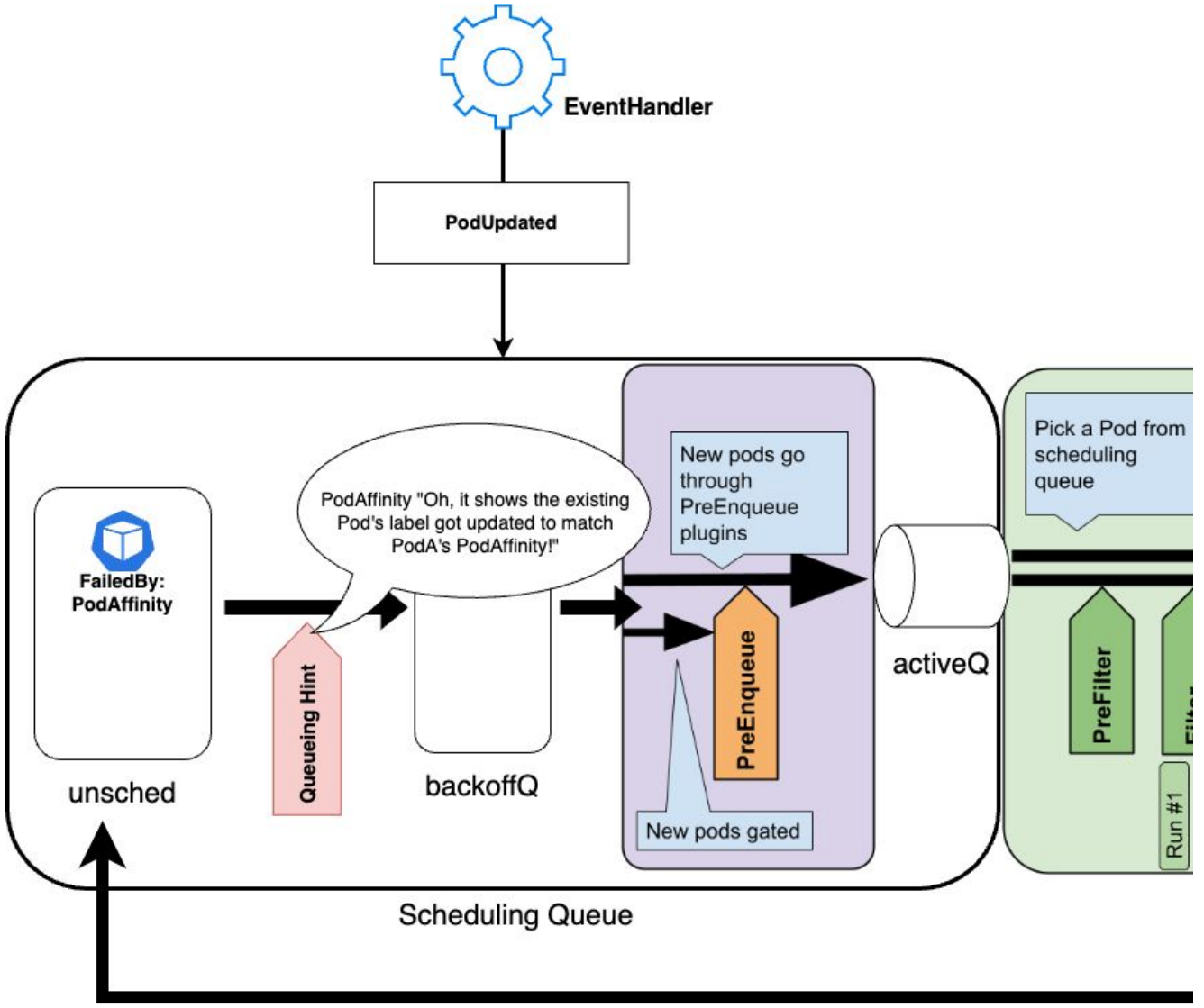
Each QueueingHint is defined to be executed with certain kind of events.

In this example, **PodAffinity** has a QueueingHint for **PodUpdated** event.



KEP-4247: QueueingHint

When the QueueingHint finds that the event could make the Pod schedulable, the scheduling queue requeues the Pod into activeQ/backoffQ so that the scheduler will retry the scheduling of the Pod.



Other updates

- Introducing **Skip** status in **PreFilter** and **PreScore**, avoiding execution of the corresponding **Filter** and **Score**.
- Removing KubeSchedulerConfiguration v1beta3 in v1.29. Use v1 instead.

Sub-project Updates

Kube-scheduler wasm-extension

An experimental extension to build scheduling plugins on WebAssembly.

```go
// Score implements api.ScorePlugin
func (pl *NodeNumber) Score(state api.CycleState, pod proto.Pod, nodeName string) (int32, *api.Status) {
        klog.InfoS("execute Score on NodeNumber plugin", "pod", klog.KObj(pod))

        var match bool
        if data, ok := state.Read(preScoreStateKey); ok {
                // Match is when there is a last digit, and it is the pod suffix.
                nodenum, ok := lastNumber(nodeName)
                match = ok && data.(*preScoreState).podSuffixNumber == nodenum
        } else {
                // Match is also when there is no pod spec node name.
                match = true
        }

        if pl.reverse {
                match = !match // invert the condition.
        }
```

implement the interfaces
→ compile this to wasm

Website: sigs.k8s.io/kube-scheduler-wasm-extension

Kube-scheduler wasm-extension


```
kind: KubeSchedulerConfiguration
apiVersion: kubescheduler.config.k8s.io/v1
profiles:
  - plugins:
      multiPoint:
        enabled:
          - name: wasm
    pluginConfig:
      - name: wasm
        args:
          guestPath: "../../../examples/nodenumber/main.wasm"
```

enable it via the scheduler configuration.

Kube-scheduler wasm-extension

Advantages 🙂

- No need to rebuild the scheduler to put your own logic into the scheduler.
- As extendable as the native Go plugins (Scheduling Framework).
- The same developer experience as Scheduling Framework via Go SDK.

Disadvantages 😕

- Latency impact: Much faster than extenders (webhook based extension), but slower than the native Go plugins.
- Wasm peculiar limitations.

Kube-scheduler wasm-extension

Project status: It's an early stage, there are many TODOs for basic features, like support all extension points.

We have an example implementation of wasm based plugins with current supported extension points. (PreFilter, Filter, PreScore and Score)
See /examples in the repo.

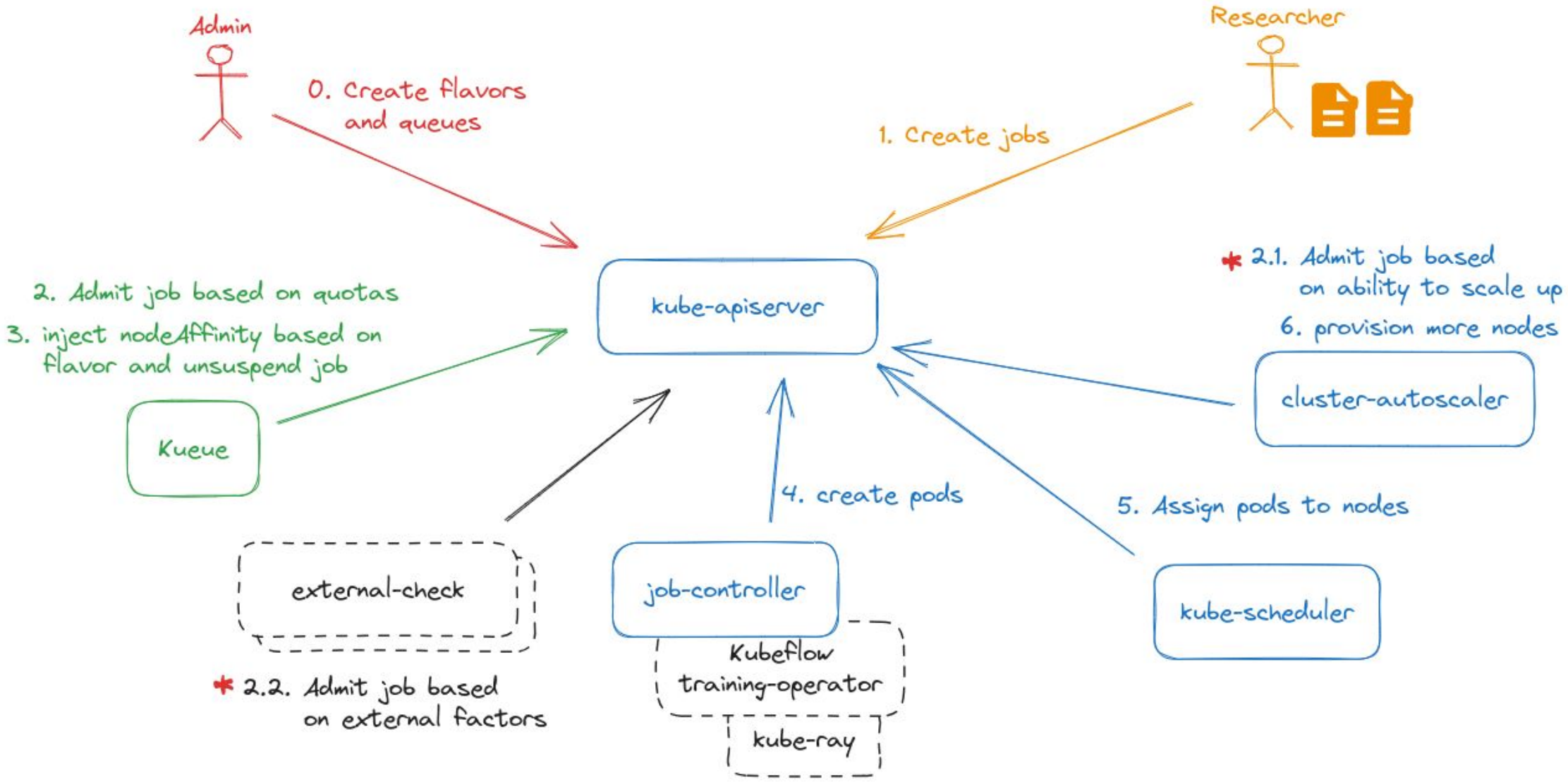
Kueue

A Kubernetes-native Job queueing system, offering:

- Resource quota management
  - Two-level hierarchy for fair sharing and borrowing
  - Priority-based ordering and preemption.
  - Fungibility: burst to on-demand, spot, other models.
- Support for k8s batch/v1.Job, JobSet, kubeflow training NEW, RayJob and plain Pods NEW.
- Extensions for:
  - Custom job CRDs.
  - Additional admission checks NEW.

Website: kueue.sigs.k8s.io

Kueue+k8s operation overview



Kueue roadmap

Top of mind:

- On-demand visibility of pending jobs.
- Support for groups of plain Pods
- Policies for requeuing Jobs from low availability (on-demand) to higher availability flavors (reservations).
- Hierarchical cohorts

Collecting feedback and prototyping:

- Multi cluster support (via AdmissionChecks)
- Workflows (DAGs)

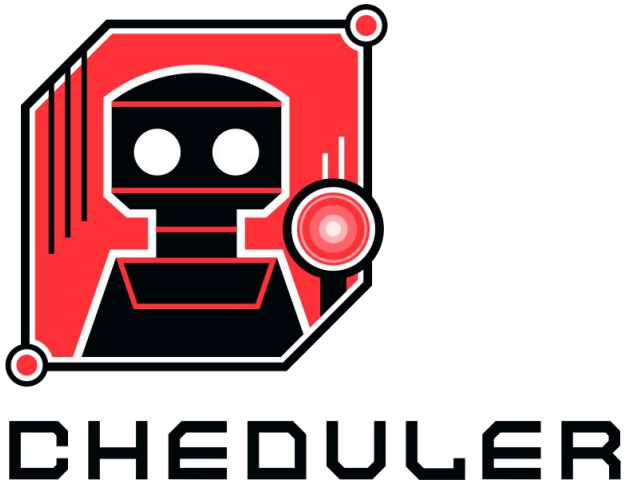
Leave your feedback in https://github.com/kubernetes-sigs/kueue/issues/1269

Descheduler

A post-scheduling Pod eviction component

New in **v0.28**:

- Consolidation of the Descheduler framework.
- Guarantees that all Balance plugins run after all Deschedule plugins #979.
- Bug fixes on the new v1alpha2 API and new framework structure.

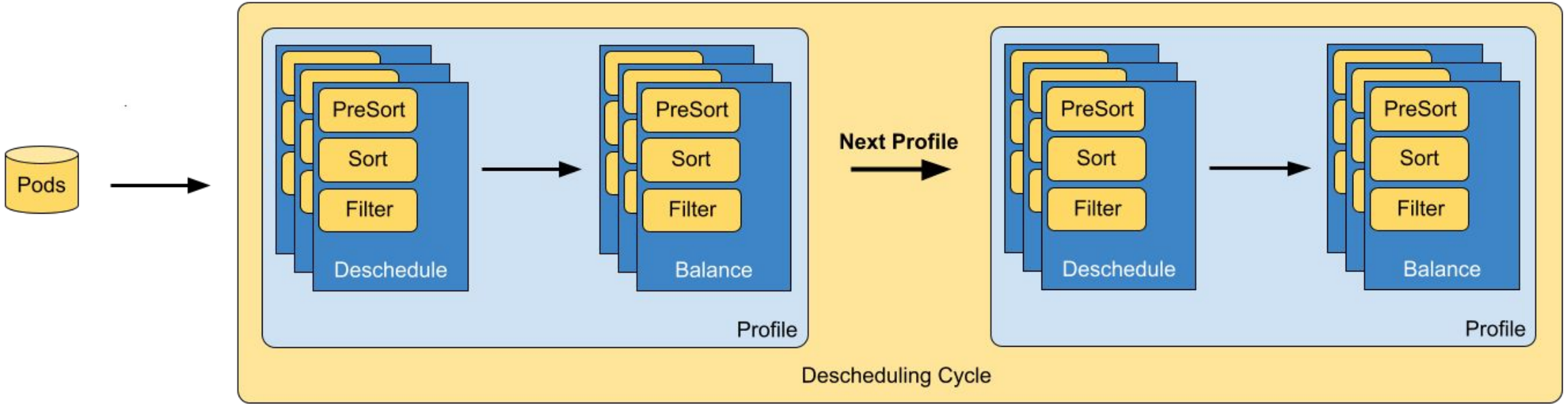


A total of 29 contributors involved in the last two releases!

Website: sigs.k8s.io/descheduler

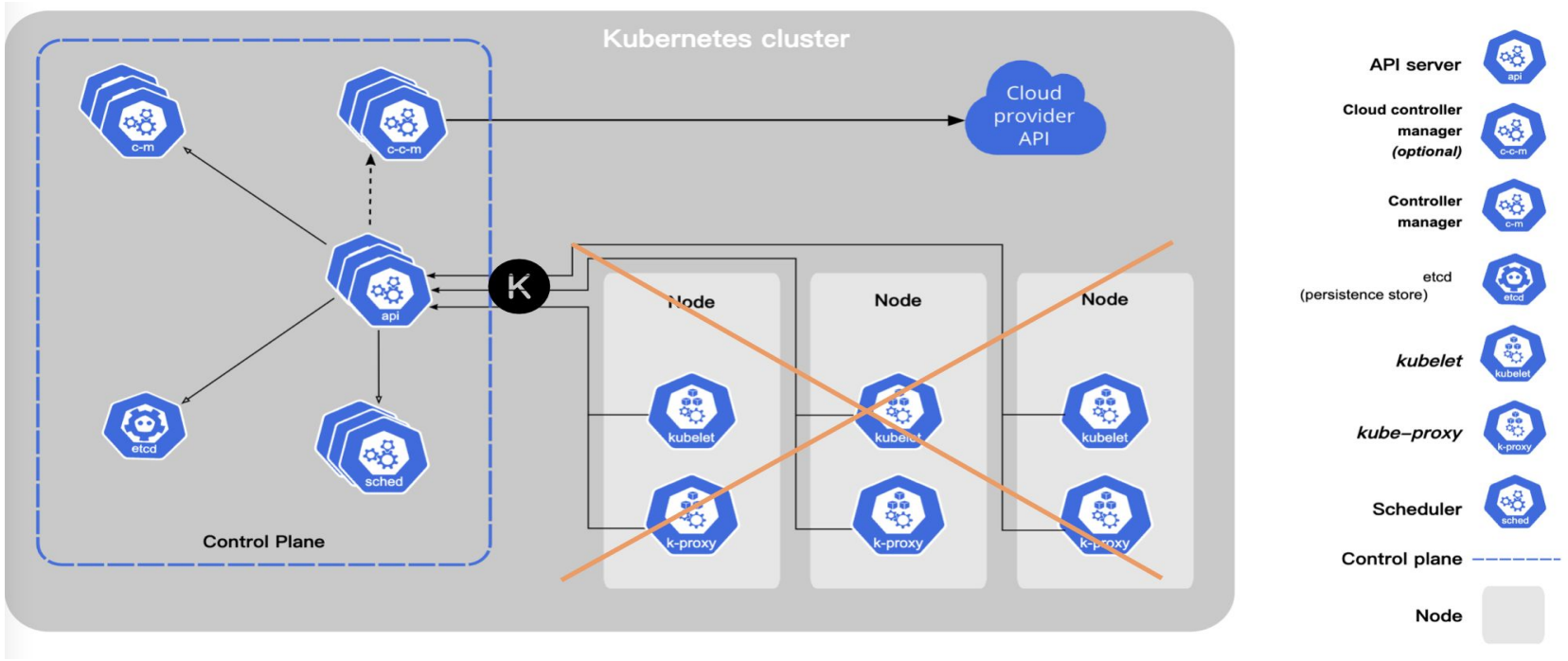
Descheduler

Descheduler Framework



KWOK

Kubernetes WithOut Kubelet is a toolkit that enables setting up a cluster of thousands of Nodes for control-plane scalability simulations.



- ➔ Github stars approaches 2k
- ➔ We have received a lot of feedback on kwok, and many users are already using it.

Website: kwok.sigs.k8s.io

KWOK updates

Highlights:

- Support for CRD/config for customization
  - Not only node and pod behavior, but also exec, logs, attach, port-forward can be emulated
- Support all-in-one image to create a kwok cluster
- Support for a kwok cluster for snapshot and restore
- kwokctl
  - can run in: docker, podman NEW, nerdctl, or even a non-container.
  - Support snapshot and restore
- Support exporting an external cluster to restore it in a kwok cluster

What's next?

- Usage simulation for CPU, memory, etc.
- [kwokctl] Record and replay support
  - Record an external cluster to replay it in a kwok cluster

Q & A

Please scan the QR Code above
to leave feedback on this session