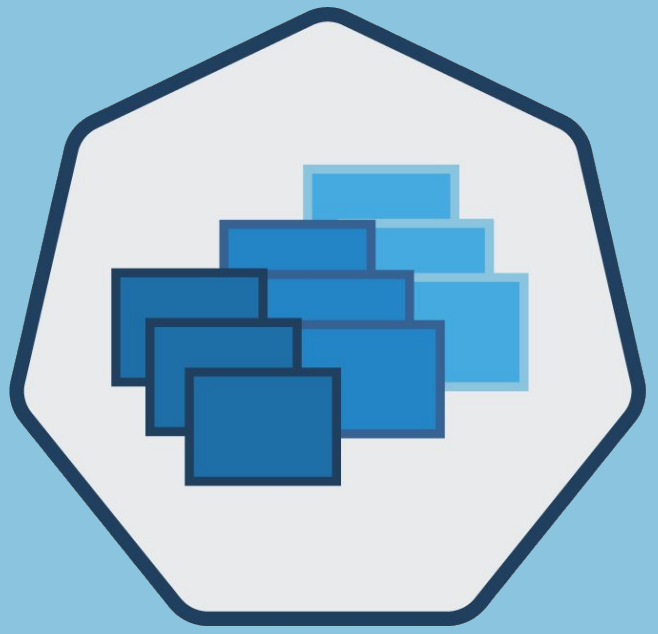


KUBERNETES

BATCH + HPC DAY

EUROPE

KUBERNETES
BATCH + HPC DAY
EUROPE

Building a Batch System for the Cloud with Kueue

Aldo Culquicondor
@alculquicondor
Google

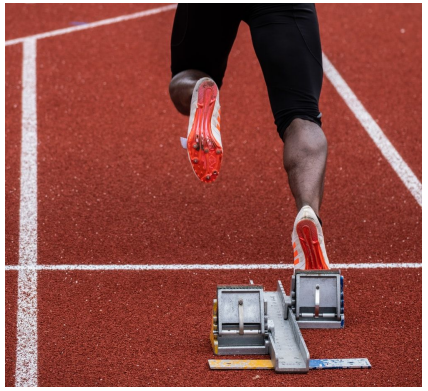
Kante Yin
@kerthcet
DaoCloud

Background

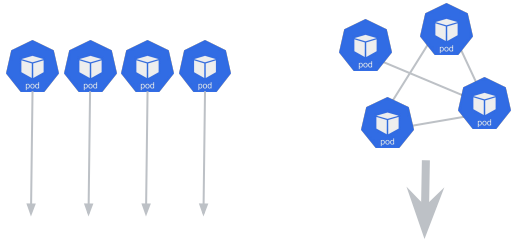
KUBERNETES
BATCH + HPC DAY
EUROPE

What is a Job?

Computations that **run to completion**



A **group of pods;** run independently or collaboratively to process a task



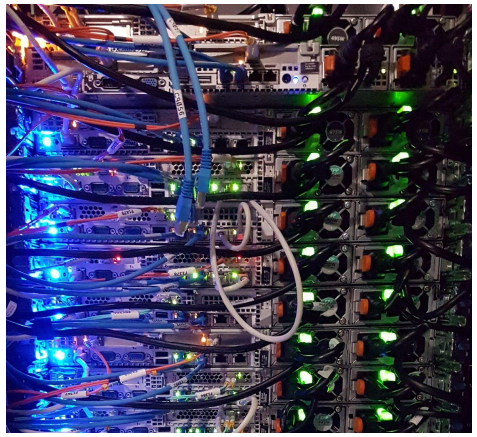
Often flexible on time, location and/or types of resources

Background

Why Job Queueing?

Lots of jobs, limited resources

On-prem: clusters are static



Cloud: discounts , quotas, scalability

What is Kueue

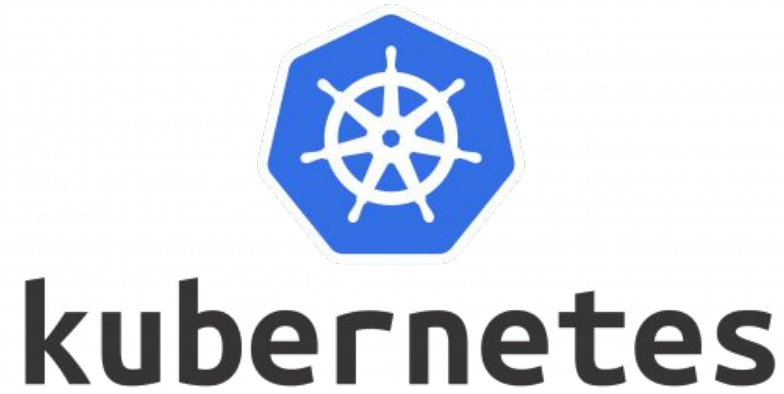
A Kubernetes-native job queueing system, offering:

- Resource quota management, with borrowing and preemption semantics.
- Resource fungibility in heterogeneous clusters.
- Support for k8s batch/v1.Job and kubeflow's MPIJob.
- Extension points and libraries for supporting custom job CRDs.
- More Job integrations coming soon

Kueue and Kubernetes

KUBERNETES
BATCH + HPC DAY
EUROPE

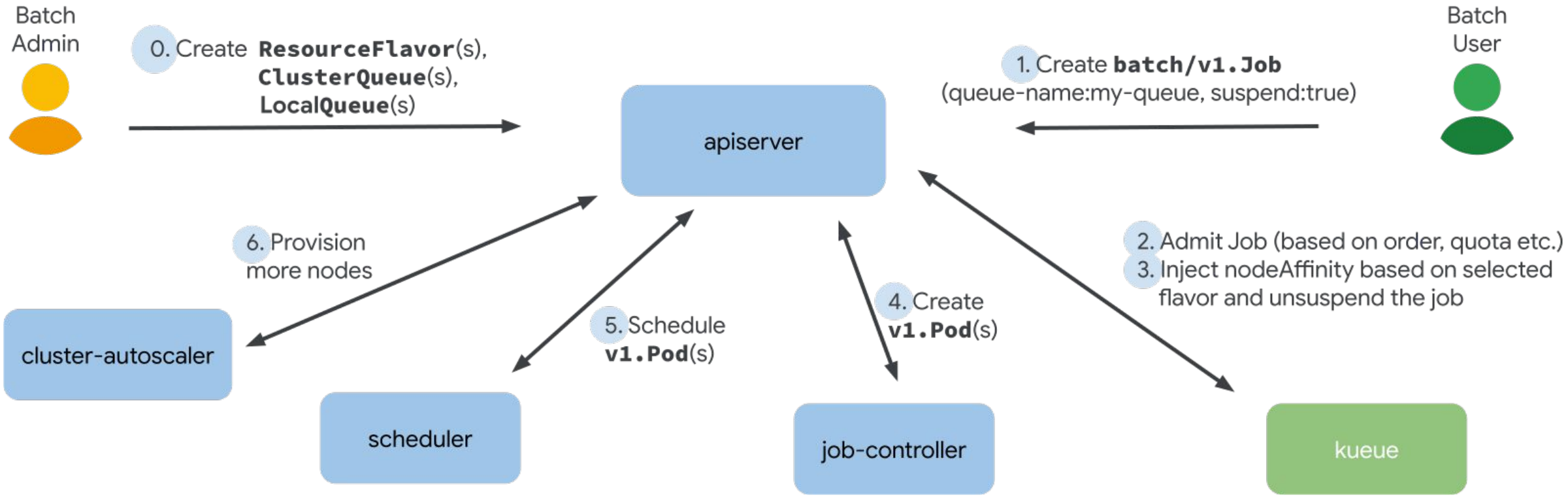
+



Design principle: compatibility and separation of concerns with standard k8s components: kube-scheduler, kube-controller-manager, cluster-autoscaler.

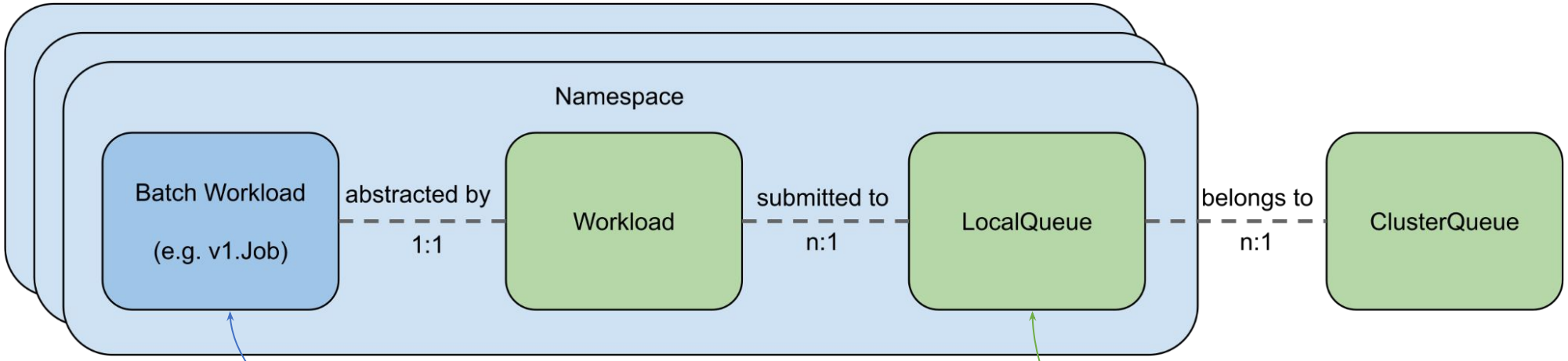
- ➔ Kueue is developed as subproject sponsored by **SIG Scheduling**.
- ➔ Close collaboration with **SIG Apps**.
- ➔ Roadmaps involving both projects are discussed in **WG Batch**.

Kueue+k8s operation overview



Kueue APIs (for end users)

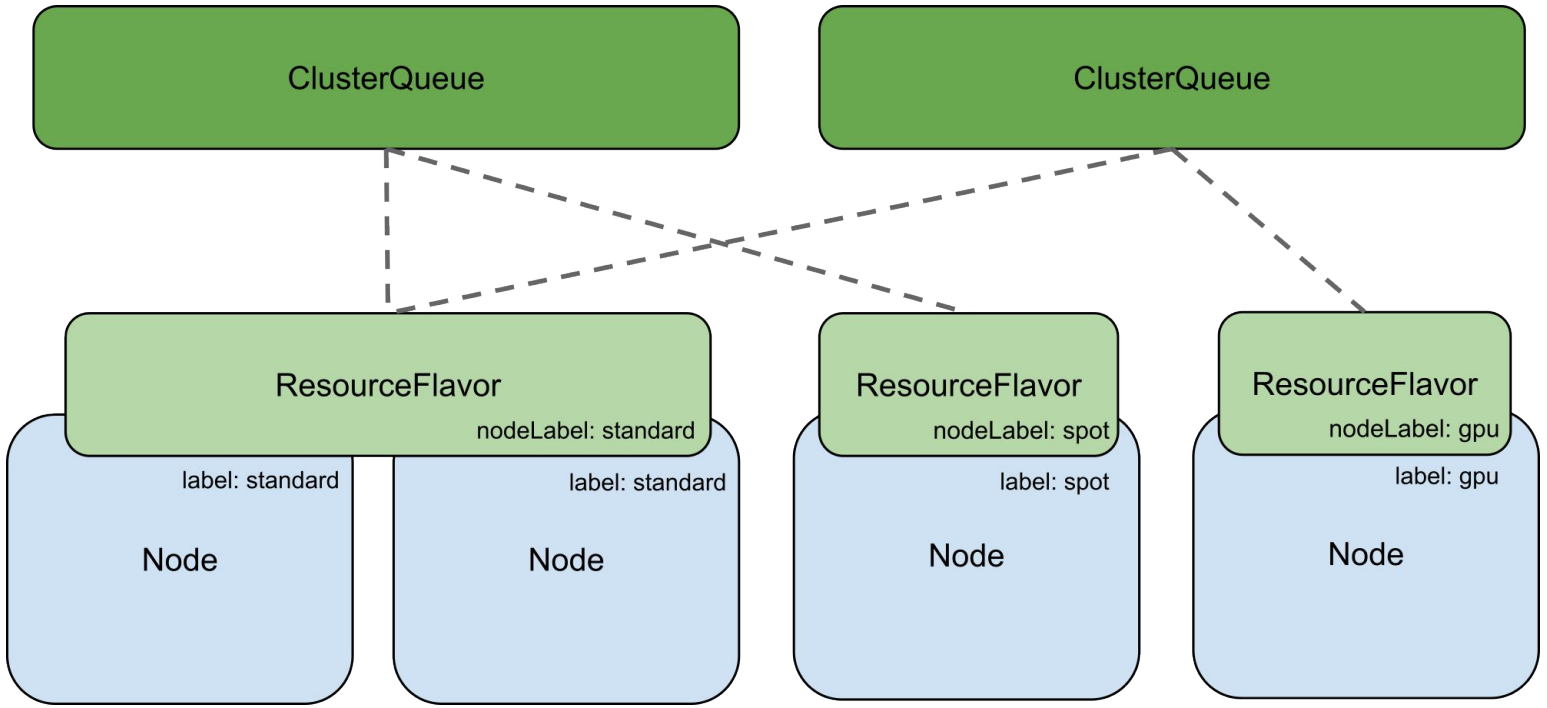
KUBERNETES
BATCH + HPC DAY
EUROPE




```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: sample-job
  labels:
    kueue.x-k8s.io/queue-name: user-queue
spec:
  parallelism: 3
  template:
    spec:
      containers:
      - name: job
        image: foo
        resources:
          requests:
            cpu: 1
```

Kueue APIs (for admins)


```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: a-cluster-queue
spec:
  resourceGroups:
  - coveredResources: ["cpu", "memory"]
    flavors:
    - name: standard
      resources:
      - name: cpu
        nominalQuota: 40
        borrowingLimit: 20
      - name: memory
        nominalQuota: 128Gi
        borrowingLimit: 64Gi
    - name: spot
      resources:
      - name: cpu
        nominalQuota: 160
      - name: memory
        nominalQuota: 512Gi
```



Use case 1: VM fungibility

KUBERNETES BATCH + HPC DAY EUROPE

Scenario: A single tenant bought a reservation of on-demand VMs, but wants to spill over to spot if there is a backlog of jobs

Batch Admin

Batch User

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: cluster-queue
spec:
  namespaceSelector: {}
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: reservation
      resources:
      - name: cpu
        nominalQuota: 40
    - name: spot
      resources:
      - name: cpu
        nominalQuota: 20
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: LocalQueue
metadata:
  name: queue
  namespace: default
spec:
  clusterQueue: cluster-queue
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ResourceFlavor
metadata:
  name: reservation
spec:
  nodeLabels:
    cloud.google.com/gke-provisioning: standard
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ResourceFlavor
metadata:
  name: spot
spec:
  nodeLabels:
    cloud.google.com/gke-provisioning: spot
```

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: sample-job
  labels:
    kueue.x-k8s.io/queue-name: queue
spec:
  parallelism: 3
  completions: 3
  template:
    spec:
      containers:
      - name: job
        image: foo
        resources:
          requests:
            cpu: 1
      restartPolicy: Never
```

Use case 1: VM fungibility

KUBERNETES BATCH + HPC DAY EUROPE

Scenario: A single tenant bought a reservation of on-demand VMs, but wants to spill over to spot if there is a backlog of jobs


```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: cluster-queue
spec:
  namespaceSelector: {}
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: reservation
      resources:
      - name: cpu
        nominalQuota: 40
    - name: spot
      resources:
      - name: cpu
        nominalQuota: 20
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: LocalQueue
metadata:
  name: queue
  namespace: default
spec:
  clusterQueue: cluster-queue
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ResourceFlavor
metadata:
  name: reservation
spec:
  nodeLabels:
    cloud.google.com/gke-provisioning: standard
```

The job is assigned the reservation flavor

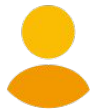
```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: sample-job
  labels:
    kueue.x-k8s.io/queue-name: queue
spec:
  parallelism: 3
  completions: 3
  template:
    spec:
      containers:
      - name: job
        image: foo
        resources:
          requests:
            cpu: 1
      restartPolicy: Never
      nodeSelector:
        cloud.google.com/gke-provisioning: standard
```

Use case 1: VM fungibility

KUBERNETES BATCH + HPC DAY EUROPE

Scenario: A single tenant bought a reservation of on-demand VMs, but wants to spill over to spot if there is a backlog of jobs

Batch Admin



```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: cluster-queue
spec:
  namespaceSelector: {}
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: reservation
      resources:
      - name: cpu
        nominalQuota: 40
    - name: spot
      resources:
      - name: cpu
        nominalQuota: 20
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: LocalQueue
metadata:
  name: queue
  namespace: default
spec:
  clusterQueue: cluster-queue
```

The job is assigned the spot flavor

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ResourceFlavor
metadata:
  name: spot
spec:
  nodeLabels:
    cloud.google.com/gke-provisioning: spot
```

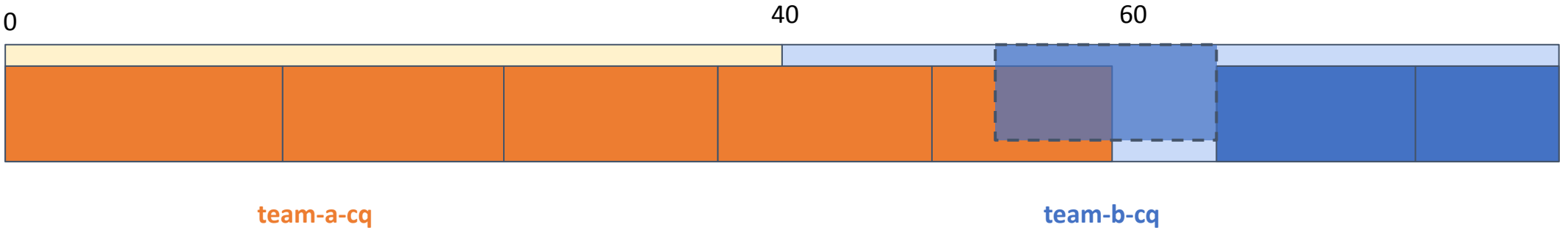
Batch User

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: sample-job
  labels:
    kueue.x-k8s.io/queue-name: queue
spec:
  parallelism: 3
  completions: 3
  template:
    spec:
      containers:
      - name: job
        image: foo
        resources:
          requests:
            cpu: 1
      restartPolicy: Never
      nodeSelector:
        cloud.google.com/gke-provisioning: spot
```

Borrowing and cohort Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

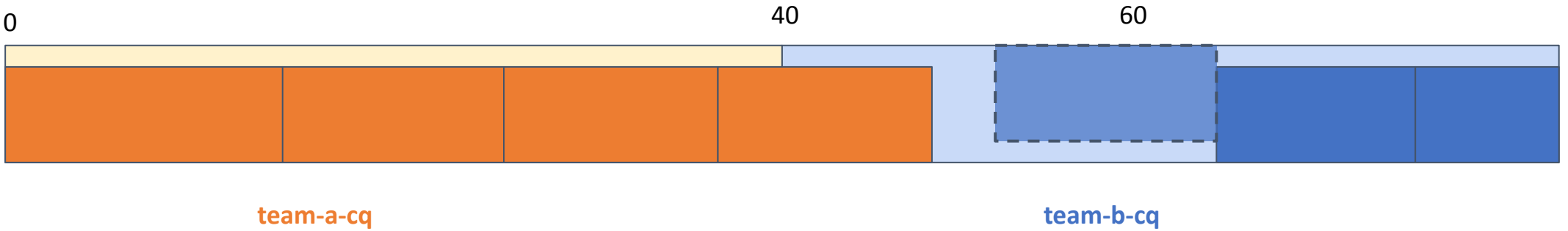
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
        borrowingLimit: 20
  preemption:
   reclaimWithinCohort: Any
```



Borrowing and cohort Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

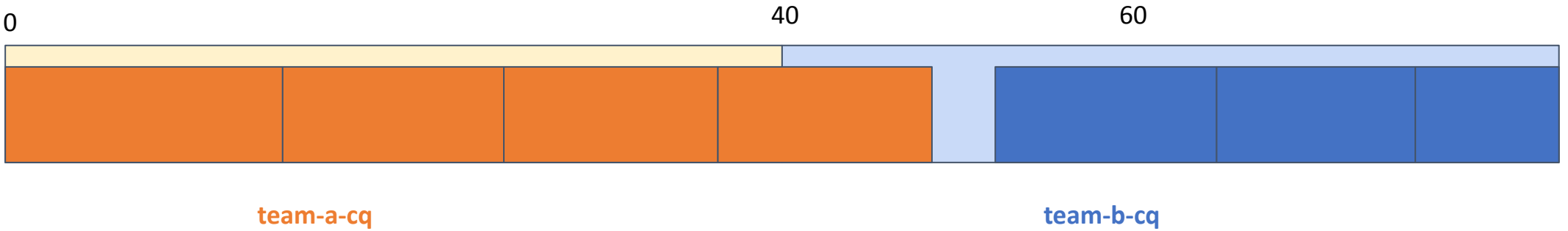
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
        borrowingLimit: 20
  preemption:
   reclaimWithinCohort: Any
```



Borrowing and cohort Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

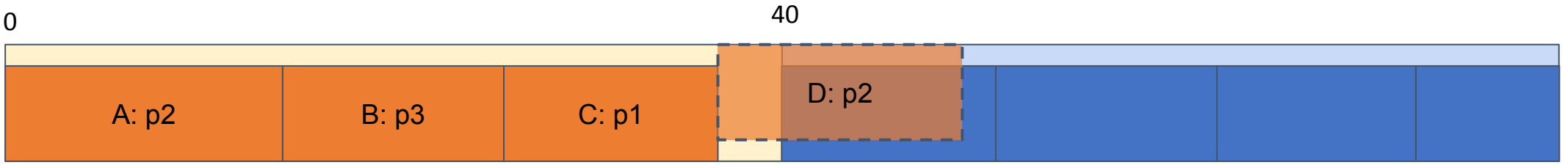
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
        borrowingLimit: 20
  preemption:
   reclaimWithinCohort: Any
```



Priority Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

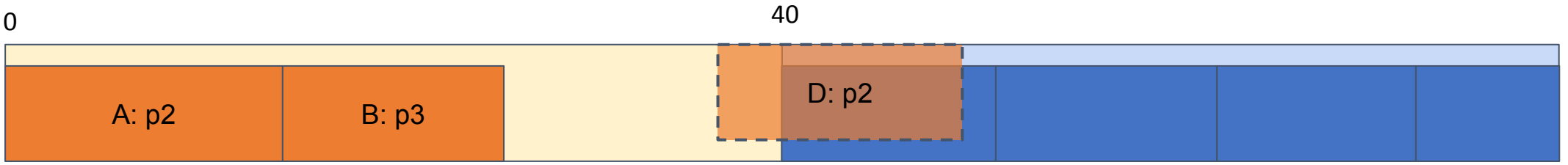
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
  preemption:
   reclaimWithinCohort: Any
   withinClusterQueue: LowerPriority
```



Priority Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

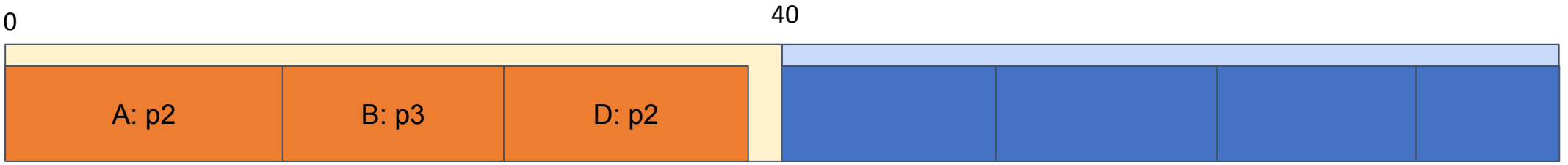
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
  preemption:
   reclaimWithinCohort: Any
   withinClusterQueue: LowerPriority
```



Priority Preemption

KUBERNETES
BATCH + HPC DAY
EUROPE

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
  preemption:
   reclaimWithinCohort: Any
   withinClusterQueue: LowerPriority
```



Use case 2: Borrowing among tenants

KUBERNETES
BATCH + HPC DAY
EUROPE

Scenario: Two teams with dedicated quota can borrow from each other when resources are underutilized

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: researcher-1
  labels:
    team: team-a
```

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: researcher-2
  labels:
    team: team-a
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  namespaceSelector:
    matchLabels:
      team: team-a
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 80
```

Use case 2: Borrowing among tenants

KUBERNETES
BATCH + HPC DAY
EUROPE

Scenario: Two teams with dedicated quota can borrow from each other when resources are underutilized

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: researcher-1
  labels:
    team: team-a
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: LocalQueue
metadata:
  name: queue
  namespace: researcher-1
spec:
  clusterQueue: team-a-cq
```

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: researcher-2
  labels:
    team: team-a
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: LocalQueue
metadata:
  name: queue
  namespace: researcher-2
spec:
  clusterQueue: team-a-cq
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  namespaceSelector:
    matchLabels:
      team: team-a
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 80
```

Use case 2: Borrowing among tenants

KUBERNETES
BATCH + HPC DAY
EUROPE

Scenario: Two teams with dedicated quota can borrow from each other when resources are underutilized

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-a-cq
spec:
  cohort: all-teams
  namespaceSelector:
    matchLabels:
      team: team-a
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 80
```

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: team-b-cq
spec:
  cohort: all-teams
  namespaceSelector:
    matchLabels:
      team: team-b
  resourceGroups:
  - coveredResources: ["cpu"]
    flavors:
    - name: default
      resources:
      - name: cpu
        nominalQuota: 40
        borrowingLimit: 40
```

Kueue release and roadmap

KUBERNETES BATCH + HPC DAY EUROPE

Kueue 0.3.0
Released: April 6th, 2023 🎉

✨ Highlights:

- API is now beta, respecting k8s deprecation policy.
- Increased validation via webhooks.
- Preemption support
- Support for kubeflow MPIJob (v1beta2)
- [Optional] WaitForPodsReady: Sequential admission for quasi all-or-nothing
- Support for LimitRanges and Runtime Classes (pod overhead)
- Library for integrating custom job-like CRDs

Kueue 0.4.0
Estimated released: June 2023

Top priorities:

- WaitForPodsReady
  - Requeue at back of queue
  - Optimistic admission and backoff
- Preemption:
  - Prevent starvation of large jobs
  - Account for terminating pods

Nice to have:

- Dynamic quota reclaiming
- Support for Ray and kubeflow

Kueue is adopting a 2-3 months release cadence from now on

How to contribute

- ➔ kueue.sigs.k8s.io
- ➔ Participate in WG Batch git.k8s.io/community/wg-batch
- ➔ Find issues with labels *help-wanted* or *good-first-issue*
- ➔ Open new issues for bugs, features or job integrations