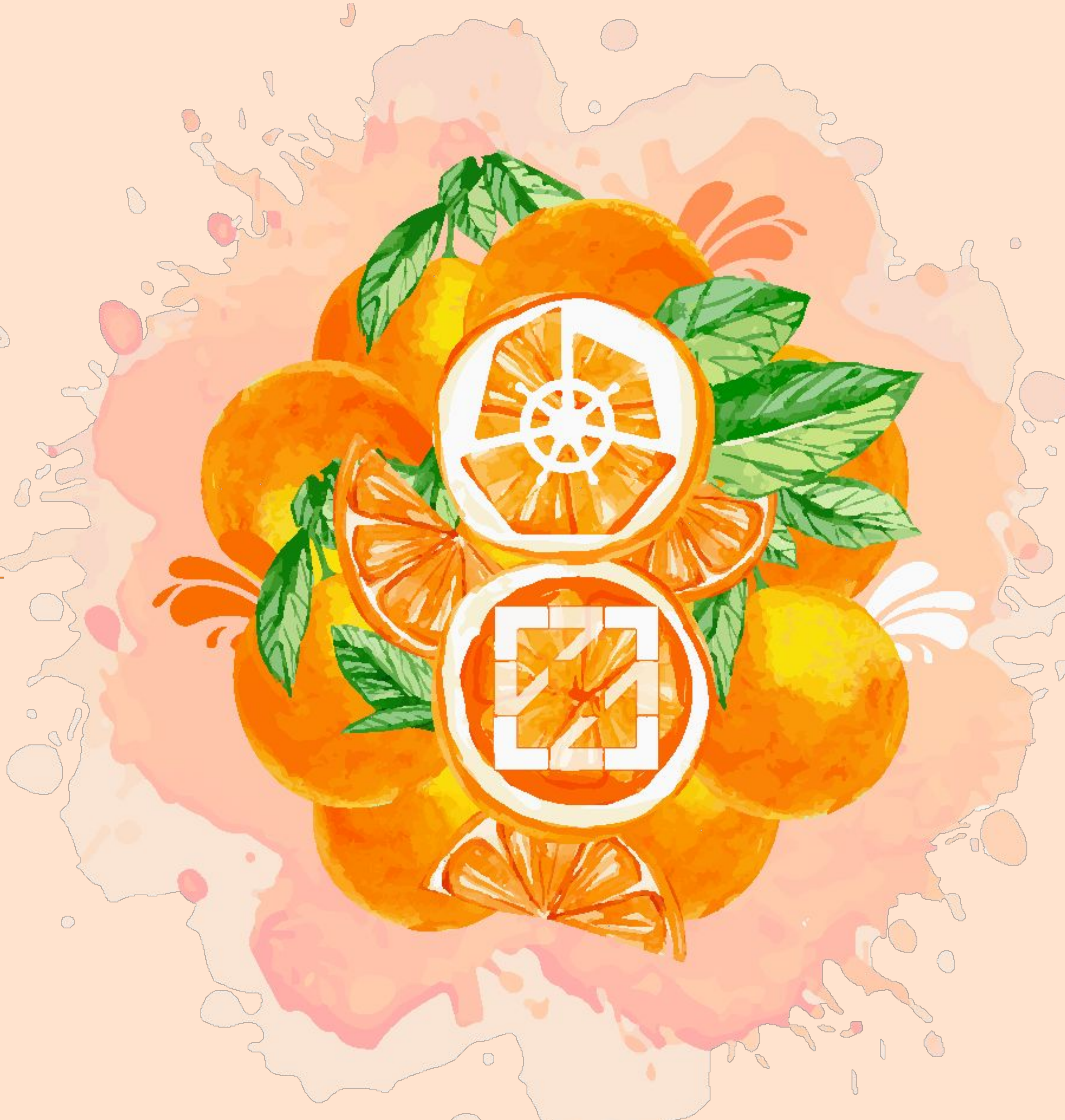
KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA



Introduction to the Kubernetes WG Batch

Abdullah Gharaibeh, Google
Aldo Culquicondor, Google
Qingcan Wang, Alibaba

Overview

1. Motivation
2. Earlier initiatives
3. Charter and workstreams
4. Where are we now
5. How to get involved

Motivation

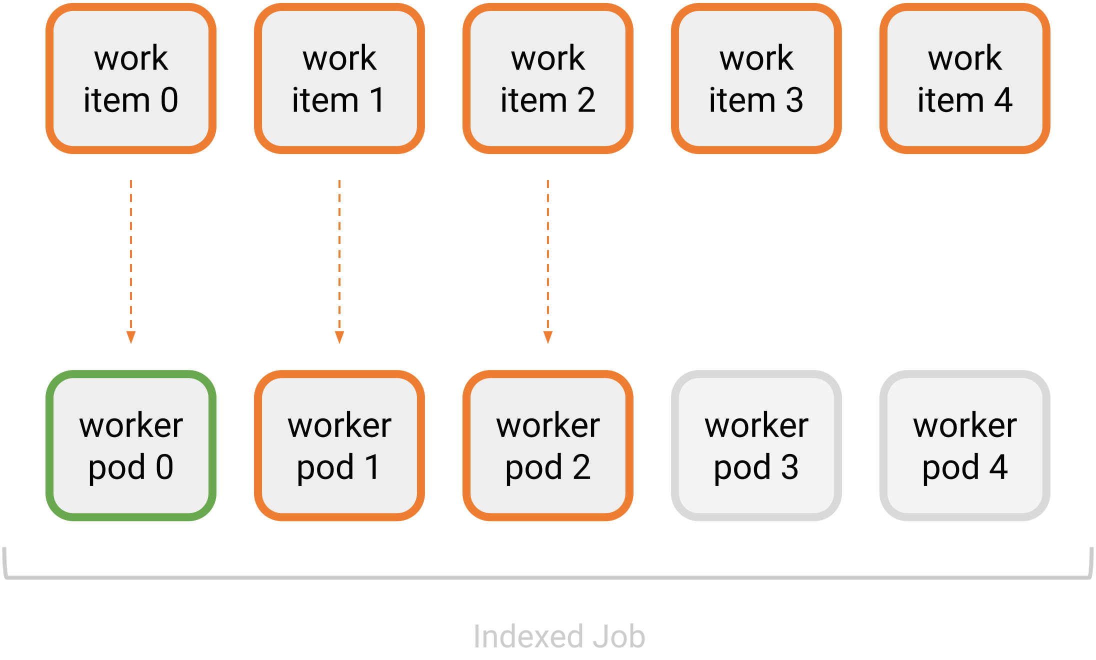
- k8s was originally built for serving applications and there is increasing support for stateful applications.
- Batch workloads can be run on k8s, but there are feature gaps:
  - Advanced completion and failure modes in Job API
  - Support for specialized devices and pinning
  - All-or-nothing pod scheduling
  - Job/workload queueing
- The status-quo was to support those features through CRDs
- There is fragmentation in the ecosystem:
  - Forked pod schedulers
  - Forked Job APIs
  - New CRIs

Earlier initiatives: SIG Apps

- Enhancements to the **batch/v1 Job** API
  - **Indexed jobs** (GA): Associate each pod of a job with an index, allowing static workload partitioning
  - **Suspended jobs** (GA): The ability to create a Job but defer the Pod creations
  - **TTL after Finish** (GA): Automatically cleanup completed jobs
  - **Accurate job tracking** (Beta): Set and manage finalizers on the pods to accurately track number of completions and failures
  - **Number of ready pods** (Beta): Report the number of ready pods in status
- **CronJob** to GA
- **Pod deletion cost**: Inform which pods to remove during ReplicaSet downscale

Earlier initiatives: Indexed Job

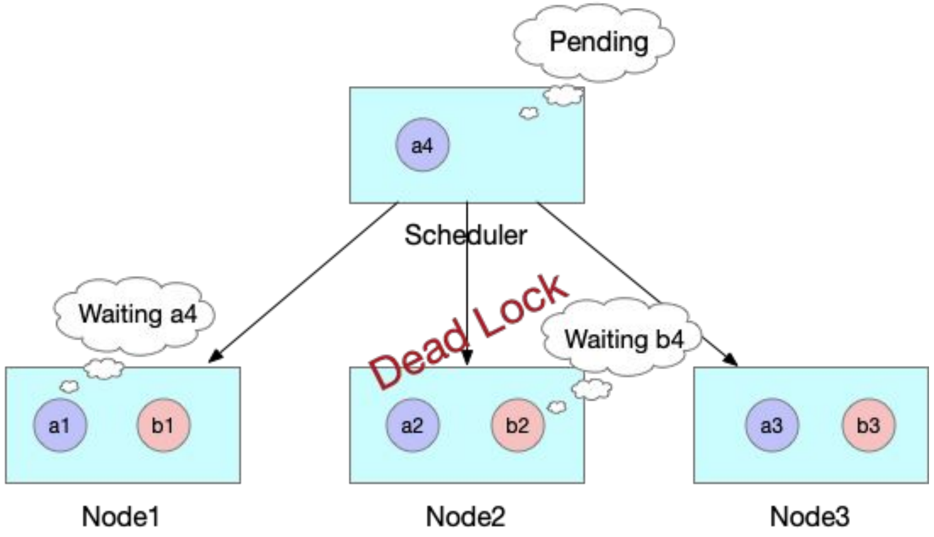
```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: 'indexed-job'
spec:
  completions: 5
  parallelism: 2
  completionMode: Indexed
  template:
    spec:
      restartPolicy: Never
      containers:
      - name: 'input'
        image: 'docker.io/library/bash'
        command:
        - "bash"
        - "-c"
        - |
          items=(banana cherry lemon orange fig)
          echo "Processing ${items[$JOB_COMPLETION_INDEX]}"
          sleep 10
```

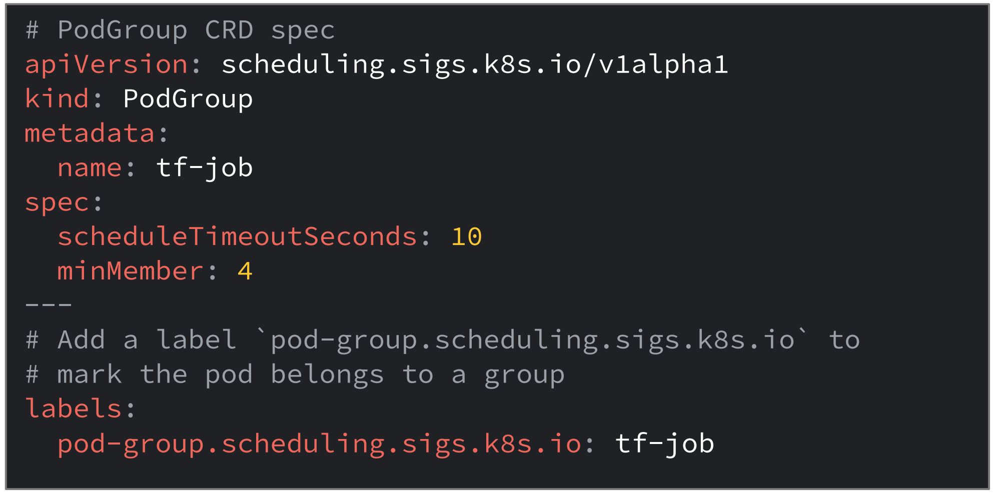


Earlier initiatives: SIG Scheduling

- **All-or-nothing Scheduling** or **Co-Scheduling:** schedule a group of pods in all-or-nothing scenarios.

  https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/pkg/coscheduling



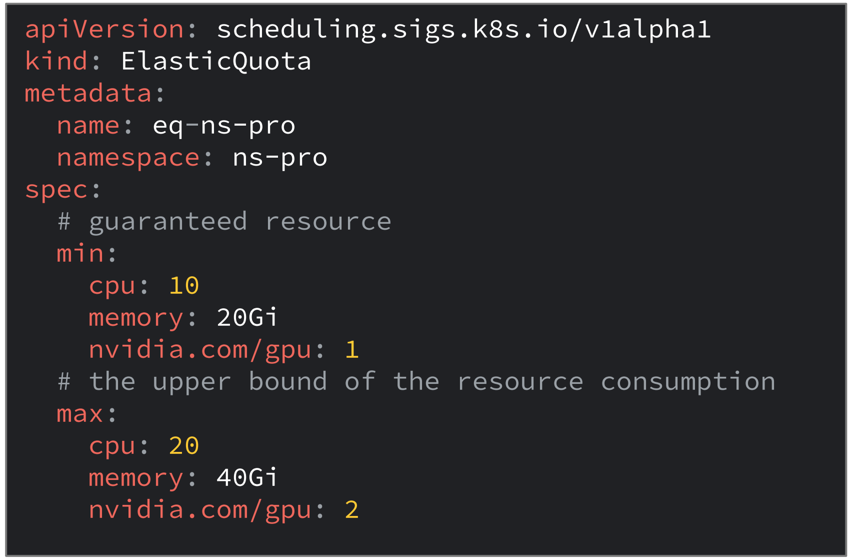


```
# PodGroup CRD spec
apiVersion: scheduling.sigs.k8s.io/v1alpha1
kind: PodGroup
metadata:
  name: tf-job
spec:
  scheduleTimeoutSeconds: 10
  minMember: 4
---
# Add a label `pod-group.scheduling.sigs.k8s.io` to
# mark the pod belongs to a group
labels:
  pod-group.scheduling.sigs.k8s.io: tf-job
```

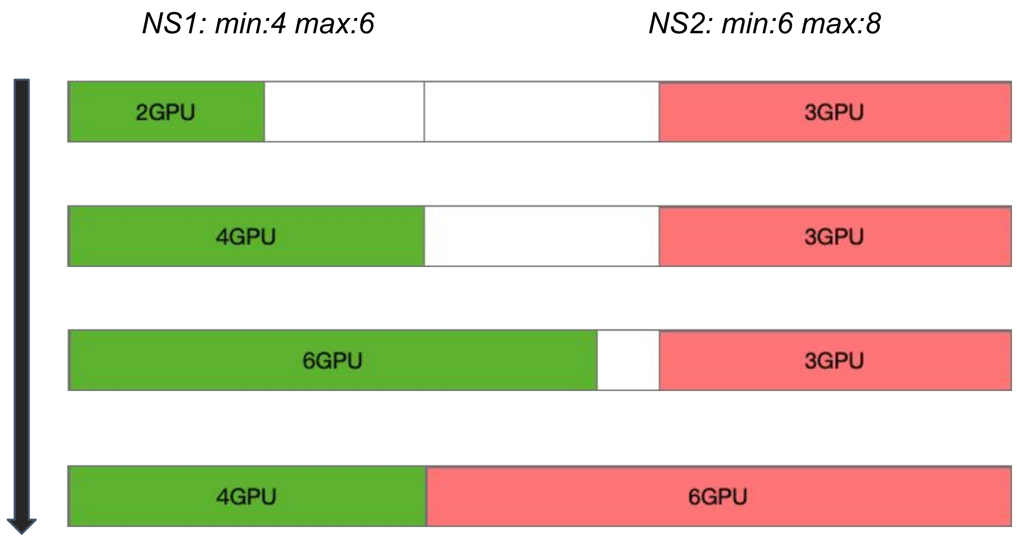
Earlier initiatives: SIG Scheduling

- **CapacityScheduling:** dynamic resource sharing between namespaces.

  https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/pkg/capacityscheduling



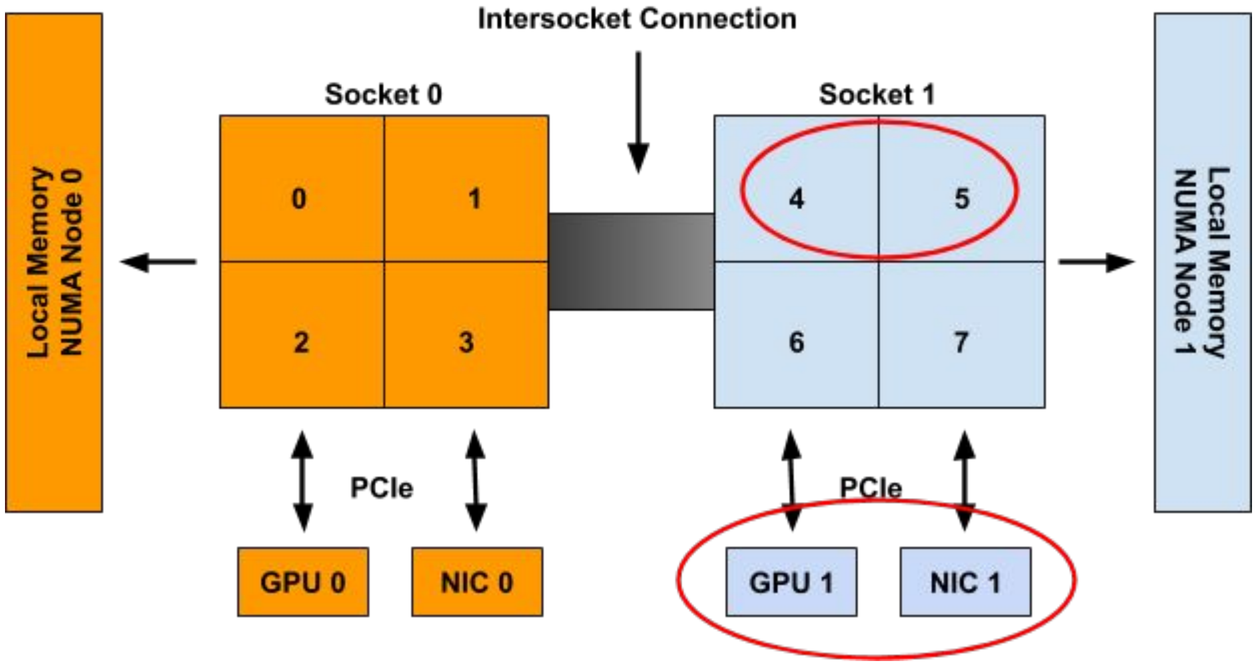
```yaml
apiVersion: scheduling.sigs.k8s.io/v1alpha1
kind: ElasticQuota
metadata:
  name: eq-ns-pro
  namespace: ns-pro
spec:
  # guaranteed resource
  min:
    cpu: 10
    memory: 20Gi
    nvidia.com/gpu: 1
  # the upper bound of the resource consumption
  max:
    cpu: 20
    memory: 40Gi
    nvidia.com/gpu: 2
```



- **Binpack:** place pods on nodes with the least amount of unused resources like CPU, memory and other extend resources (like GPU) . *https://kubernetes.io/docs/concepts/scheduling-eviction/resource-bin-packing/*

Earlier initiatives: SIG Node

- Kubelet's **Topology manager**: policies to allocate a container or a pod to a single or a set of NUMA nodes.



Earlier initiatives: SIG Node

- **NodeResourceTopology** CRD to expose the topology of a node
- **Topology-aware scheduling plugin** sigs.k8s.io/scheduler-plugins/pkg/noderesourcetopology to schedule based on the kubelet topology policy.

WG Batch charter

- **Mission**: Discuss and enhance the support for Batch workloads in core Kubernetes. The goal is to unify the way users deploy batch workloads to improve portability and to simplify supportability for Kubernetes providers.

- **Stakeholders**
  - SIG Apps
  - SIG Autoscaling
  - SIG Node
  - SIG Scheduling

git.k8s.io/community/wg-batch

WG Batch workstreams

- Advance the Job API to support a wider range of workloads (static partitioning, MPI, ML, AI).
- Job management, queueing, provisioning, scheduling and autoscaling.
- Runtime and scheduling support for specialized hardware (accelerators, NUMA, RDMA, etc.)

Where are we now (1/2)

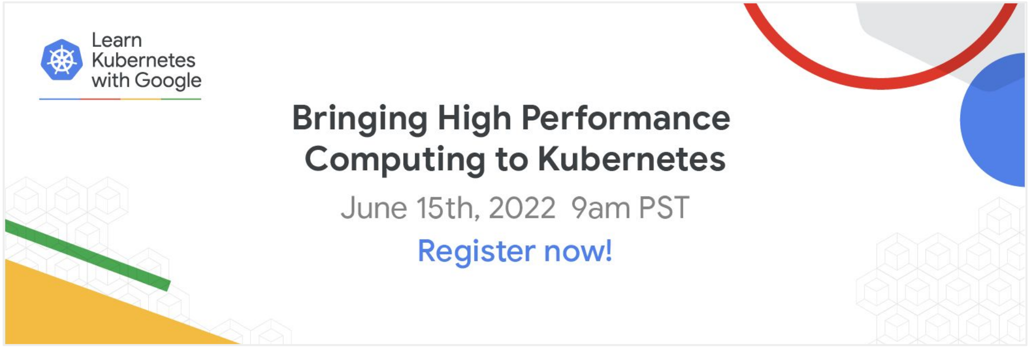
- SIG Scheduling is sponsoring a project for job queueing: sigs.k8s.io/kueue
  - A job-level manager that decides when a job should start or stop
  - Leverages the `.spec.suspend` field in Job
  - Can be extended to any custom workload CRD
  - Main design principle is “no duplication”. Kueue co-exists with kube-controller-manager, kube-scheduler and cluster-autoscaler
  - Usage:
    - Admins define queues with resource flavors, quotas and borrowing cohorts.
    - Users just create jobs
  - v0.1.0 released in April 2022

Where are we now (2/2)

- Current debates:
  - Advanced policies for failures, completions and retries for (Indexed) batch/v1 Jobs
  - An API for grouping pods to do all-or-nothing scheduling
  - An API to reserve node resources for future pods
  - A plugin model for resource managers in kubelet
- Welcoming presentations from batch projects running on kubernetes to understand feature gaps and potential solutions

How to get involved

Do you have a Batch related project, enhancement proposal or prototype? Come present!



Learn Kubernetes with Google

Bringing High Performance Computing to Kubernetes

June 15th, 2022 9am PST

Register now!

goo.gle/LearnK8s-live

- ➔ slack.k8s.io #wg-batch
- ➔ wg-batch@k8s.io
- ➔ git.k8s.io/community/wg-batch

Questions?

Abdullah Gharaibeh, @ahg-g
Aldo Culquicondor, @alculquicondor
Qingcan Wang, @denkensk